

CSS Frameworks

The Ultimate Guide

SUFYAN BIN UZAYR



CRC Press
Taylor & Francis Group

CSS Frameworks

Cascading Style Sheets (CSS) is a programming language that integrates all relevant information related to the display of a web page. CSS defines the style and format of a web page or page, including layout, colors, fonts, & (the space around each element), and more. Together with HTML and JavaScript, CSS forms the basis of the functioning of the Internet.

CSS Frameworks: The Ultimate Guide discusses the fundamental concepts of CSS that will help developers and coders do more with CSS Frameworks. This book is for anyone who wants to create a website and write optimized and well-structured code in CSS. It follows a hands-on approach and offers practical lessons and tutorials related to CSS Frameworks.

Features:

- Provides a detailed focus on CSS programming
- Introduces CSS Frameworks using real-world industry concepts
- Discusses optimization and maintenance of code in CSS Frameworks

This book is a valuable resource for developers already familiar with HTML, and who have an understanding of JavaScript. After finishing this book, anyone will quickly build their website with absolute ease.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

CSS Frameworks

The Ultimate Guide

Sufyan bin Uzayr



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

First edition published 2024

by CRC Press

6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742

and by CRC Press

4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

CRC Press is an imprint of Taylor & Francis Group, LLC

© 2024 Sufyan bin Uzayr

Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, access www.copyright.com or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. For works that are not available on CCC please contact mpkbookspermissions@tandf.co.uk

Trademark notice: Product or corporate names may be trademarks or registered trademarks and are used only for identification and explanation without intent to infringe.

ISBN: 9781032413242 (hbk)

ISBN: 9781032413228 (pbk)

ISBN: 9781003357520 (ebk)

DOI: 10.1201/9781003357520

Typeset in Minion

by codeMantra

For Dad



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Contents

About the Author, xxiii

Acknowledgments, xxv

CHAPTER 1 ■ INTRODUCTION TO CSS FRAMEWORKS	1
IN THIS CHAPTER	1
WHAT IS CSS?	1
HISTORY	2
CSS SYNTAX	2
DIFFERENT TYPES OF CSS YOU CAN USE	4
A DIFFERENT VERSION OF CSS	4
CSS SPECIFICATION	4
CSS MODULES	5
STARTING WITH A SPECIFIC HTML	5
LINKING YOUR HTML AND CSS FILES	5
HOW CSS AND HTML WORK TOGETHER	6
HOW DOES CSS WORK?	6
USE CLASSES IN CSS	7
How to Apply Classes on Paragraphs in HTML	7
STYLE ITEMS BASED ON THEIR LOCATION IN THE DOCUMENT	8
CSS SELECTORS	9
What Is a Selector?	9
UNIVERSAL SELECTOR	9
ELEMENT TYPE SELECTORS	9

Id Selector	9
Class Selectors	10
Descendant Selectors	10
Child Selectors	11
Adjacent Sibling Selectors	11
Grouping Selectors	12
Grouping Selectors	12
CSS COMMENTS	13
HTML AND CSS COMMENTS	13
CSS COLORS	13
Basic CSS Colors	14
CSS Color Names	14
How to Use These Colors	14
In Hexadecimal Notation	14
In RGB Functional Notation	15
List of Color Properties	15
List of Color Values	16
CSS BACKGROUNDS	16
CSS BACKGROUND COLOR	17
Transparency/Opacity	17
To Demonstrate This Using the RGBA	17
CSS Background-Image	17
CSS Background-Repeat	18
CSS Background-Repeat: No-Repeat	18
CSS Background-Attachment	18
CSS BORDERS	19
CSS Border Style	19
CSS Border Width	19
HEX Values	20
CSS Margins	20
CSS Padding	21
Padding and Element Width	21
CSS LENGTH, WIDTH, AND MAX-WIDTH	22

Length and Width of CSS Values	22
CSS BOX MODEL	22
CSS Layout – Location Area	22
<i>The Position Property</i>	22
CSS TABLES	23
CSS Layout – The Display Property	23
BLOCK-LEVEL ELEMENTS	23
Inline Elements	24
CSS LAYOUT – z-INDEX LAYOUT	24
Without z-Index	24
CSS LAYOUT – WIDTH AND MAX-WIDTH (USING WIDTH, MAX-WIDTH, AND MARGIN: AUTO;)	24
CSS LAYOUT – OVERFLOW	25
CSS LAYOUT – FLOAT AND CLEAR	25
The Float Property	25
CSS3 LAYOUTS	25
How to Use CSS Properties	26
CSS FUNCTIONS	26
CSS FLEXBOX	29
Why Flexbox?	29
CSS BOX SIZE	30
CSS Syntax	30
CSS GRID LAYOUT	30
CSS MEDIA TYPES	31
Introduction to Media Types	31
Method 1: Using @media At-rules	31
Method 2: Using @import At-rules	32
Method 3: Using the <link> Element	32
DIFFERENT MEDIA TYPES	32
MEDIA QUESTIONS AND RESPONSIVE WEB DESIGN	33
MEDIA QUERIES	33
CSS ATTRIBUTE SELECTORS	34

CSS FRAGMENTATION	36
Types of Fragmentation	36
Understanding Attribute Selectors	36
Selecting from a List of Values	37
Selecting a Value That Starts with a Character Unit	37
Selecting the Ending Value per Character Unit	38
Selecting a Value That Contains a Character Unit	38
Case Insensitivity	38
Select a Value or a Value Followed by a -	38
BENEFITS YOU MAY KNOW	38
CSS Cons	40
CHAPTER SUMMARY	41
FURTHER READING	41
NOTE	43
CHAPTER 2 ■ BOOTSTRAP	45
IN THIS CHAPTER	45
INTRODUCTION	45
BOOTSTRAP HISTORY	46
VERSIONS OF BOOTSTRAP	47
BOOTSTRAP 5	50
GETTING STARTED WITH BOOTSTRAP	51
HOW TO DOWNLOAD BOOTSTRAP USING PACKAGE MANAGER	51
Downloading Bootstrap Using npm	52
<i>Downloading Bootstrap Using a Cable</i>	52
<i>Download Bootstrap Using Composer</i>	52
How to Download Bootstrap Using CDN	52
What Is CDN?	53
Bootstrap CDN Link and Scripts	53
<i>For CSS</i>	53
<i>For JavaScript</i>	53
<i>For JQuery Library</i>	53
<i>For Popper.js</i>	53

How Do We Use Bootstrap CDN?	54
BOOTSTRAP TEMPLATE	54
Is Popper.js Required for Bootstrap 4?	56
RESPONSIVE META TAG	56
HISTORY	56
Bootstrap 2	56
Bootstrap 3	57
Bootstrap 4	57
Bootstrap 5	57
FEATURES	58
CREATE A HOME PAGE WITH BOOTSTRAP	59
Add HTML5 Document Type	59
Bootstrap 3 Is Advanced	59
BASIC BOOTSTRAP PAGES	59
BOOTSTRAP CONTAINERS	60
BOOTSTRAP GRID SYSTEM	63
GRID CLASSES	63
BASIC STRUCTURE OF A BOOTSTRAP GRID	64
Explanation	65
Code Example	65
Code Example (Extra Small Grid)	67
Code Example (Small Grid)	67
Code Example (Medium Grid)	69
Code Example (Small and Medium Grid Combined)	69
Code Example (Large Grid)	69
Code Example (Extra-Large Grid)	71
HOW THE GRID WORKS	72
Three Equal Columns	73
Two Unequal Columns	74
FLEXBOX	77
Flexbox Properties in CSS	78
Flexbox Properties in Bootstrap	79
Flex-Direction in CSS	80

Flex Direction in Bootstrap	80
<i>Flexbox Features</i>	81
Flex-Wrap in CSS	82
Flex-Wrap in Bootstrap	83
Flex-Flow	83
Justify-Content in CSS	84
Order	86
Justify Content in Bootstrap	89
Align Self	90
Align Items	91
Grow and Shrink	92
Fill	93
RESPONSIVE BREAKPOINTS	93
z-Index	94
COMPONENTS IN BOOTSTRAP	95
Bootstrap Alerts	95
Alert Links	96
BOOTSTRAP BADGES	97
BOOTSTRAP JUMBOTRON	98
Full-Width Jumbotron	98
BOOTSTRAP BUTTONS	99
Button Sizes	100
Block Level Buttons	100
Active/Disabled Buttons	100
Spinner Buttons	100
Colored Spinners	100
Button Tags	101
Outline Buttons	101
DROPDOWNS	102
NAVBAR	103
How It Works	103
Supported Content	104

UTILITIES	105
Border	105
CHAPTER SUMMARY	105
FURTHER READING	105
CHAPTER 3 ■ BULMA	107
IN THIS CHAPTER	107
INTRODUCTION	107
Requirement	107
Install the Bulma CSS Library	108
Using the npm Bulma Package	108
Features and Benefits	108
For CDN	109
BROWSER SUPPORT	109
Starter Bulma Template	110
Column System	111
CUSTOMIZING VARIABLES	111
Bulma-Start	112
WITH NODE-SASS INSTALLATION	113
Create a Package .json File	113
Install the Dev Dependencies	113
CREATE A SASS FILE	113
Compliance with These Community Guidelines	114
HOW TO USE SASS	114
INSTALL SASS	115
LIBRARIES	115
Install Anywhere (Standalone)	115
Now Create a Sass File	115
Create an HTML Template Which Uses Various Bulma Components	116
UTILITIES	119
Mixins	119

CSS Mixins	122
Direction Mixins	125
RESPONSIVE MIXINS	126
from() and until() mixins	126
from ()	127
Description	127
until ()	127
FORM CONTROL MIXINS	129
SIZES	130
CONTROL PLACEHOLDER	130
FUNCTIONS	131
FINDCOLORINVERT () FUNCTION	131
The findLightColor () and findDarkColor ()function	131
EXTENDS	132
CHAPTER SUMMARY	132
FURTHER READING	133
NOTES	133
CHAPTER 4 ■ TAILWIND	135
IN THIS CHAPTER	135
INTRODUCTION	135
WHY TAILWIND CSS?	136
WHO SHOULD USE TAILWIND CSS?	136
GET STARTED WITH TAILWIND CSS	136
Install Tailwind CSS	136
Add the Tailwind Directives to CSS	137
Now Start the Tailwind CLI Build Process	137
Start Using Your Tailwind in HTML	137
CDN	137
Add the CDN Script to HTML	137
TAILWIND TYPOGRAPHY	139
Font Family	139
TAILWIND CSS LAYOUT	141

Tailwind Container	141
KEY CONCEPTS	146
How Do We Use JIT Engine Mode?	147
JIT MODE FEATURES?	148
TAILWIND CSS BOX SIZING	148
TAILWIND CSS DISPLAY	150
Display Tailwind CSS Classes	150
FLOAT CLASSES	156
TAILWIND CSS CLEAR	158
CSS CLEAR CLASSES	158
OVERFLOW CLASSES OF TAILWIND CSS	163
TAILWIND CSS FLEX DIRECTION	171
TAILWIND CSS FLEX	174
FLEX	174
TAILWIND CSS ORDER	176
TAILWIND CSS GRID TEMPLATE COLUMNS	178
TAILWIND CSS ALIGNMENT	181
Justify Content	181
TAILWIND CSS JUSTIFY ITEMS	183
TAILWIND CSS SPACING	186
Padding	186
MARGIN	190
SPACE BETWEEN	194
TAILWIND CSS BORDERS	197
Border Radius	197
BORDER WIDTH	203
BORDER COLOR	206
BORDER OPACITY	208
BORDER STYLE	211
RING OPACITY	213
RING WIDTH	213
Ring – Width Classes	213

TAILWIND EFFECTS	215
TAILWIND TABLES	217
Border Collapse	217
Table Layout	217
BENEFITS OF USING TAILWIND CSS	220
WHY USE TAILWIND CSS?	221
Syntax	221
Small Styles	221
Less Cognitive Load	221
Better Composition	222
Consistency	222
Customization	222
Play CDN	223
Fancy Underline Styles	223
TAILWIND CSS: DISADVANTAGES	223
Style and HTML Are Mixed	223
Learning Takes Time	223
Lack of Essential Style Elements	224
Documentation	224
CHAPTER SUMMARY	224
FURTHER READING	224
NOTE	224
CHAPTER 5 ■ FOUNDATION	225
IN THIS CHAPTER	225
INTRODUCTION	225
WHY FOUNDATION?	226
INSTALLATION	226
Download Foundation	226
Template	227
FILE STRUCTURE OF FOUNDATION FOLDER	228
HTML5 Doctype	228
MOBILE RESPONSIVE FIRST	228
INITIALIZATION OF COMPONENTS	229

WHAT COMES WITH FOUNDATION?	229
COMPATIBILITY	229
WHAT WILL NO LONGER WORK?	230
BASIC PRINCIPLES	230
BASIC CSS KITCHEN SINK	231
Accordion	231
FOUNDATION CSS KITCHEN SINK ACCORDION	
ADVANCED OPTIONS	232
ADVANCED OPTIONS	236
Multi-Expand	236
All Closed	238
ACCORDION AND URLs	238
Deep Linking	238
CALLOUT	240
Coloring	242
SIZING	244
MAKING CLOSABLE	246
CARD	248
CARD DIVIDER	250
Images	250
Sizing	250
DROPDOWN	252
POSITIONING	254
OTHER POSITIONING	257
MEDIA OBJECT	259
SECTION ALIGNMENT	261
OFF-CANVAS	264
OFF-CANVAS POSITION	266
REVEAL (MODEL)	268
SIZING	270
FEATURES	273
FOUNDATION INTEGRATION LEVELS	274

CHAPTER SUMMARY	274
FURTHER READING	274
CHAPTER 6 ■ SKELETON	275
IN THIS CHAPTER	275
INTRODUCTION	275
Installation	276
FILE STRUCTURE	276
SUPPORTED BROWSER	276
SKELETON – THE GRID	277
TYPOGRAPHY	280
Typography Units Used in CSS Frameworks	282
Absolute Lengths	282
RELATIVE LENGTHS	286
BUTTONS	286
FORM	291
LISTS	293
CODE	295
Benefits of Skeleton CSS Framework	297
Disadvantages of Skeleton CSS Framework	297
CHAPTER SUMMARY	297
FURTHER READING	297
NOTES	298
CHAPTER 7 ■ MILLIGRAM	299
IN THIS CHAPTER	299
INTRODUCTION	299
MILLIGRAM INSTALLATION	299
CLI	300
Typography Units Used in CSS Frameworks	300
Absolute Lengths	301
RELATIVE LENGTHS	304
TYPOGRAPHY	304
BLOCKQUOTES	306

BUTTONS	308
LISTS	311
FORMS	313
TABLES	314
GRIDS	316
CODE	318
UTILITIES	319
COLUMN WIDTHS	321
RESPONSIVENESS	323
REASONS TO USE MILLIGRAM	325
WHY NOT TO USE?	325
CHAPTER SUMMARY	326
FURTHER READING	326
NOTES	326
CHAPTER 8 ■ PURE CSS	327
IN THIS CHAPTER	327
INTRODUCTION	327
Why Pure CSS?	327
INSTALLING PURE CSS	328
OVERVIEW	331
RESPONSIVE DESIGN	331
Standard CSS	331
Extensible	331
PURE CSS MENU RESPONSIVE CSS MENU	331
PURE CSS FOUNDATION	339
Pure CSS Hiding Elements	339
PURE CSS IMAGES	341
PURE CSS BUTTONS	343
BUTTONS SIZES	349
PURE CSS TABLES	352
CHAPTER SUMMARY	358
FURTHER READING	358
NOTE	358

CHAPTER 9 ■ MATERIALIZE	359
IN THIS CHAPTER	359
INTRODUCTION	359
WHY USE MATERIALIZE CSS?	359
INSTALLING MATERIALIZE CSS	360
Method 1	360
Method 2	360
Method 3	360
Method 4	360
MATERIALIZE COLOR	362
MATERIALIZED CSS GRIDS	367
COLUMNS FOR SMALL SCREEN DEVICES	367
COLUMNS FOR MEDIUM SIZE SCREEN DEVICES	369
COLUMNS FOR LARGE SIZE SCREEN DEVICES	371
MATERIALIZE CSS UTILITY CLASSES	373
MATERIALIZE CSS MEDIA	375
MATERIALIZE – TABLES	377
MATERIALIZE – BUTTONS	381
MATERIALIZE – CARDS	383
MATERIALIZE – FORM	386
FEATURES	389
CHAPTER SUMMARY	389
FURTHER READING	390
CHAPTER 10 ■ CSS CHEAT SHEET	391
GETTING TO KNOW CSS	391
USING CSS IN HTML PAGES	392
Inline CSS, by Adding a Style Attribute to the HTML Tag	392
Internal CSS, by Inserting a Style Tag in the Head Section	393
External CSS	394
CSS CHEAT SHEET	395
Font	395

Text	397
Backgrounds	402
Borders	408
Box Model	416
Margin	419
Padding	420
Template Layout	420
Table	421
Colors	425
Grid Positioning	429
Columns	432
List and Markers	433
Animations	435
Outline	437
Positioning	439
Transitions	441
3D/2D Transform	442
Pseudo-Class	449
Pseudo-Element	469
CHAPTER SUMMARY	474
CHAPTER 11 ■ APPRAISAL	475
PREREQUISITES OF CSS FRAMEWORKS	476
ALTERNATIVE VERSIONS OF CSS FRAMEWORKS	477
INDEX, 479	



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

About the Author

Sufyan bin Uzayr is a writer, coder, and entrepreneur with over a decade of experience in the industry. He has authored several books in the past, pertaining to a diverse range of topics, ranging from history to computers/IT.

He is the Director of Parakozm, a multinational IT company specializing in EdTech solutions. He also runs Zeba Academy, an online learning and teaching vertical with a focus on STEM fields.

He specializes in a wide variety of technologies, such as JavaScript, Dart, WordPress, Drupal, Linux, and Python. He holds multiple degrees, including ones in management, IT, literature, and political science.

He is a digital nomad, dividing his time between four countries. He has lived and taught in numerous universities and educational institutions around the globe. He takes a keen interest in technology, politics, literature, history, and sports, and in his spare time, he enjoys teaching coding and English to young students.

Learn more at sufyanism.com.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Acknowledgments

THERE ARE MANY PEOPLE who deserve to be on this page, for this book would not have come into existence without their support. That said, some names deserve a special mention, and I am genuinely grateful to:

- My parents, for everything they have done for me.
- The Parakozm team, especially Divya Sachdeva, Jaskiran Kaur, and Simran Rao, for offering great amounts of help and assistance during the book-writing process.
- The CRC team, especially Sean Connelly and Danielle Zarfati, for ensuring that the book's content, layout, formatting, and everything else remain perfect throughout.
- Reviewers of this book, for going through the manuscript and providing their insight and feedback.
- Typesetters, cover designers, printers, and everyone else, for their part in the development of this book.
- All the folks associated with Zeba Academy, either directly or indirectly, for their help and support.
- The programming community in general, and the web development community in particular, for all their hard work and efforts.

— Sufyan bin Uzayr



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Introduction to CSS Frameworks

IN THIS CHAPTER

- Introduction
- Versions of CSS
- Linking HTML with CSS files
- CSS Concepts in brief

In this chapter, we will learn about what is CSS, its fundamental concepts, versions, and syntax, and how we can link our HTML page with CSS files. More concepts like modules, selectors, ids, borders, table, form, and colors will also be covered in this chapter.

So let's get started with CSS.

WHAT IS CSS?

A document is usually a text file created using a markup language – HTML is the common markup language, but you can also merge with other markup languages such as SVG or XML. It presenting a document to a user means converting it into a form used by audience. The browsers such as Firefox, Chrome, or Edge are designed to render text as it is displayed, for example on a computer screen, projector, or printer.

Cascading Style Sheets (CSS) is a programming language that integrates all relevant information related to the display of a webpage. CSS defines the style and format of a webpage or page, including layout, colors, fonts, the space around each element, and more. Together with HTML and Javascript, CSS forms the basis of the functioning of the Internet. All three standards and specifications are managed by the World Wide Web Consortium (W3C).

CSS can be used to create very basic text styling – such as changing the color and size of headings and links. It can be used to create a structure – for example, to convert a single column of text into a structure with a main content area and a bar for related information. It can be used for effects such as animation.

HISTORY

CSS is highly regarded by Norwegian Håkon Wium Lie, who, back in 1994, wanted to make a standard style sheet for the World Wide Web. The first site Lie tried for CSS is the Arena web browser. Since its creation, Lie has continued to coproduce versions of CSS1, CSS2, and RFC 2318 with Tim Berners-Lee and Robert Cailliau. In the first decade of existence (1994–2004), it has become the standard website that has the significant impact on the look and feels of the global web as we know it today. CSS3 was released in 1999.

Web standards are a topic close to Lie's heart. Since the introduction of CSS, he has appealed to major technology players such as Microsoft and other browsers to support the standard web standards and continue to improve.

CSS SYNTAX

Now move on to the original CSS. The thing we will do is make the paragraph text a different color. So, type this into style file .css.

```
p {color: blue; }
```

This looks different from the code in HTML file because it is a different syntax. You will add a white area and cut into that code as follows:

```
p {  
  color: blue;  
}
```

Both of the above examples are exactly the same about browser. But developers often write CSS as the latest example to differentiate styles.

CSS is a rule-based language – defines rules by specifying groups of styles to be applied to specific objects or groups of objects on your webpage. For example, you may decide that a major topic on your page is shown as a red flag. The following code shows a simple CSS rule.

```
p {
  color: red;
  font-size: 5em;
}
```

- For example, the CSS rule opens with a selector.
- Selects the HTML element that will style it. In this case, we style the first level titles (<p>).
- Then we have a set of twisted pieces {}.
- Inside the instruments, there will be one or more announcements, which take the form of the goods and pairs of value. Specifies the location (color in the example above) before the colon, and specifies the value of the property after the colon (red for example).
- This example contains two declarations: one color and the other font size. Each pair specifies the location of the element (s) we select (<p> in this case), and then the value we would like to give the structure.

CSS layouts have different valid values, depending on the specified format. In our example, we have a color package, which can take on different color values. We also have a font size structure. This structure can take units of various sizes as a value.

The CSS style sheet will contain many such rules, written in sequence.

```
p {
  color: red;
  font-size: 5em;
}

h2 {
  color: black;
}
```

DIFFERENT TYPES OF CSS YOU CAN USE¹

We have three types of CSS:

- CSS might appear in another external file.
- It may appear inside the top of a document or webpage.
- It may appear in the queue, next to the CSS text.

External Style Sheets contain CSS instructions, and these are special files, and have a .css file extension. If an external style sheet is installed on any webpage, the CSS file will control its sound and appearance.

A DIFFERENT VERSION OF CSS

CSS3 and CSS4: These versions build in CSS2.1, adding new functionality and maintaining backward compatibility. Some features are still being tested and may change in the future. Use this with caution, as it may cause problems with your site.

CSS2.1 (recommended): This version has fixed many bugs and problems in CSS2 and is now the official, recommended version of CSS.

CSS1 and CSS2 (no longer applicable): These were the first two versions of CSS and are no longer updated or maintained.

CSS SPECIFICATION

All web-level technologies (HTML, CSS, JavaScript, etc.) are defined in large documents called specifications (or “specs”), published by standard organizations (such as W3C, WHATWG, ECMA, and Khronos) and describes accurately. How those technologies should behave. CSS is no other – it is developed by an internal W3C team called the CSS Working Group. This group is made up of representatives of browser vendors and other companies interested in CSS. There are also other people, known as invited experts, who act as independent voices, affiliated with the member organization.

New features are developed or specified by the Working Group – sometimes because a particular browser is interested in a specific skill, sometimes because web designers and developers request a feature, and because

the Working Group itself has identified a requirement. CSS is constantly evolving, with new features available. However, the important thing about CSS is that everyone works hard so that they never change things in a way that can break old websites. A website built in 2000, using the limited CSS that existed then, should still be used in the browser today!

As a newcomer to CSS, you may find the CSS specification too great – it is intended for developers to apply feature support to user agents, not for web developers to learn to understand CSS. Many experienced developers would prefer to look at MDN documents or other courses. However, it is worth knowing that these features exist and to understand the relationship between the CSS you are using, browser support (see below), and details.

CSS MODULES

As there are many things you can style using CSS, the language is divided into modules. You will see a reference to these modules as you explore MDN. Many text pages are organized around a specific module. For example, you can refer to the MDN reference in the Background and Boundaries module to find out what its purpose is and the properties and features it contains.

In this section, you do not need to worry too much about how CSS is created; however, it may be easier to get information if, for example, you know that a certain place may be found among other things, and therefore, it is almost in the same situation.

For example, let's go back to the Background and Borders module – you may think it makes sense that background color and border-color structures are defined in this module. And you will be right.

STARTING WITH A SPECIFIC HTML

HTML and CSS are two methods of tags (code), with their own unique syntax. There is an important difference between the two. You can think of HTML as a page layout, while CSS gives HTML its own style.

HTML = structure
CSS = style

LINKING YOUR HTML AND CSS FILES

Before we can write CSS, we actually have to go back to our HTML. We need to write a new line to link the HTML file and CSS file together. So, open the HTML file and add the provided line “<link href=” style.css “rel =” stylesheet “type=” text / CSS “/>”. Your file code looks like this:

```

<!DOCTYPE html>
<html>
  <head>
    <title>This is page title.</title>
    <link rel="stylesheet" type="text/CSS" href="style.
css" />
  </head>
  <body>
    <h1>This is a heading element</h1>
    <p>Hello world, this is a paragraph.</p>
  </body>
</html>

```

This line of code links a new CSS file to your HTML file. Let's split it: the href attribute specifies the link associated with the CSS file. We'll get to the links later, for now, just make sure the style.css file is in the same folder as index.html file. The rel attribute tells your browser that this is a style sheet. The attribute type tells the browser that this linked file should be translated as CSS syntax.

HOW CSS AND HTML WORK TOGETHER

CSS is only concerned with webpage layout, while page content is defined using marker language such as HTML. The separation of style and content has various advantages, and among them are improved accessibility and more control over web design.

CSS documents are used to define a webpage style and then linked to an HTML document (or document in a different tag language) that contains the content and layout of the page. Setting the style directly to the HTML document is possible, but not recommended. CSS texts can be created in any text editor, such as Text Editor on Mac or Notepad for Windows, as well as many other free or paid options that you can download.

HOW DOES CSS WORK?

CSS contains the rules and values that a web browser can use in the content of a webpage to better display its content. For example, you can use CSS to define that the body part of a page has a blue background, and that the text is displayed in a white Helvetica font with a size of 18px. CSS rules are read by a specific document format (hence the word 'cascading' in Cascading Style Sheets). In general, the next step in a CSS document is the winner – unless the first rule is very clear. For example, you

might have two rules for your conflicting CSS file – such as setting body font in blue and section font in green. In this case, the effective rule will be the category font, because it is much clearer than the body font.

USE CLASSES IN CSS

CSS syntax contains a selector, and the category exactly. You need to style HTML elements – including changing colors, fonts, or text size.

If you want to use the class, use the full stop (.) followed by the class name in the style block. Next, use a bracelet called a declaration block that contains a feature to create an object style, such as text color or text size. Let's take an example, here's what it looks like if you want to turn the text color green:

```
<style>
.text {
color: # 000;
}
</style>
```

Let's divide and see what each part represents:

- `<style>`: The style tag defines the style information of the HTML object. You can place multiple style tags within an HTML document.
- `.classname`: refers to features that will be added to the feature. For example, the name of the class is green text with 36 fonts as the element will be green, with 36 pixels. Also, do not forget to put a dot (.) before the class name as it will not work the other way around.
- `{property}`: contains attributes that will be added to objects, such as green color and font size of 36 pixels.

How to Apply Classes on Paragraphs in HTML

```
<html>
<head>Title</head>
<body>
<!-- After creating the classes, you can apply them
to HTML elements: -->
```

```
<p class="text">This is an example of a sentence with  
regular font size</p></body>  
</html>
```

There are some things you need to look for when creating classes in CSS, such as the attributes you will add to the elements and the number of classes you want to create. If you plan this in advance, it will help you to edit the code easily.

Overall, you should also pay attention to these three factors:

- **Specify your class:** The name of the class should provide information about the attributes added to the feature. Therefore, it will be easier to understand. For example, if you want to add a class that will change the font size to 36, the best bet would be to name it a font-36.
- **Make sure the class does only one thing:** Adding more than one attribute might make the class harder to use again. For example, you cannot use a class that contains two attributes – green text and 36 pixel font size if you only want green text without changing the font size.
- **Organize your classes based on their characteristics:** When you create classes, organize them based on their usage. For example, you can collect classes related to font size. This practice would help you in the long run.

CSS classes will help you style HTML elements quickly. Additionally, you can avoid duplicating the same code for each HTML object using the same style. Therefore, the amount of CSS code to be used decreases – making it more readable and easier to remove.

In another note, you can also use the ID selector to style HTML elements. However, it can only change one HTML object, and the class selector can create more than one object.

STYLE ITEMS BASED ON THEIR LOCATION IN THE DOCUMENT

There are times when you will want something to look different based on where you are in the document. There are a number of selectors who can help you here, but for now, we will look at a few. In our text, there are two aspects – one in the middle and the other in the middle of the list item. To select only the placed inside the element, you can use a selector

called the descendant combinator, which takes the form of space between the other two selectors.

CSS SELECTORS

What Is a Selector?

CSS selector for matching features on a webpage. The style rules connected with that selector will apply to items such as the pattern. Choices are one of the most important CSS features as they allow you to direct certain elements to your webpage in a variety of ways to style them.

A few types of selectors are available in CSS, let's take a closer look.

UNIVERSAL SELECTOR

It is indicated by a star (*), corresponding to each part of the page. The universal selector may be removed if other conditions are present in the feature. This filter is often used to remove automatic genes and pads from elements for quick testing.

Let's try the following example to understand how it basically works:

```
* {
  margin: 0;
  padding: 0;
}
```

The style rules within the selector * will apply to everything in the document.

ELEMENT TYPE SELECTORS

It is the same as every element in a document and the name of the corresponding element type. Let's try some example to see how it really works:

```
p {
  color: blue;
}
```

Id Selector

The id selector is used to describe style rules for one or more items. The id selector is defined by a hash (#) symbol that is immediately followed by the id value.

Example:

```
#text{  
    color: red;  
}
```

This style rule gives the text the object in red, and its id identifier is being set as default.

Class Selectors

It is used to select any HTML component with a class attribute. All features with that class will be formatted according to a defined rule. The class selector is defined by an intermediate symbol (.) That is immediately followed by a class value.

Example:

```
.green {  
    color: blue;  
}
```

Descendant Selectors

You can use these options if you need to select an interest-bearing element of another element, for example, if you want to identify only those anchors contained in the random list, rather than directing all anchor elements. Let's see how it works:

Example:

```
ul.menu li a {  
    text-decoration: none;  
}  
h1 em {  
    color: green;  
}
```

The style rules within the selector `ul.menu li a` apply only to those elements `<a>` contained within the `<ul>` element that has a class `menu`, and that does not affect other links within the document. Similarly, style rules within the `h1 em` selector will only apply to those elements of `<em>` content

contained within the element of `<h1>` and which do not affect other elements of `<em>` `<em>`.

Child Selectors

The child selector is used to select only those specific child items for a particular feature. The children's selector is made up of two or more selectors that are separated by a larger symbol (`>`). You can use this selector, for example, to select the first level of list items within a nested list with more than one level. Let's look at an example how it works.

```
ul > li {
    list-style: circle;
}
ul > li ol {
    list-style: none;
}
```

The style rule inside the selector such as `ul>li` applied to only those `<li>` elements that are direct children of the `<ul>` elements, has no effect on other list elements.

Adjacent Sibling Selectors

Adjacent sibling selectors are used to select its sibling elements (i.e., elements at the same level). This selector has a syntax such as: `E1 + E2`, where `E2` is the target selector.

The `h1+p` selector in the following example will select `<p>` elements only if the `<h1>` and `<p>` elements share the same parent in the document tree and `<h1>` precedes the `<p>` section immediately. This means that only those sections that come immediately after each `<h1>` title will have corresponding style rules. Let's see how this option really works.

Example:

```
h1 + p {
    color: blue;
    font-size: 18px;
}
ul.text + p {
    color: black;
    font-size: 30px;
}
```

Grouping Selectors

The standard sibling selector is similar to the nearest sibling selector (E1+E2), but less powerful. The standard sibling selector is made up of two simple picks separated by a tilde (~) character. It can be written as follows: E1 ~ E2, where E2 is the purpose of the selector.

The h1 ~ p selector in the example below will select all the <p> features preceded by the <h1> section, where all the features share the same parent in the document tree.

```
h1 ~ p {
    color: blue;
    font-size: 18px;
}
ul.task ~ p {
    color: # f0f;
    text-indent: 30px;
}
```

Grouping Selectors

Usually a few selectors on a style sheet share declarations of the same style rules. You can group them with a comma-separated list to narrow the code to your style sheet. It also prevents others from repeating the same style rules over and over again.

Let's take a look:

```
h1 {
    font-size: 36px;
    font-weight: normal;
}
h2 {
    font-size: 28px;
    font-weight: normal;
}
h3 {
    font-size: 22px;
    font-weight: normal;
}
```

CSS COMMENTS

CSS comments are not visible in the browser, but may be helpful in writing your source code. Comments are also used to explain the code, and can be helpful if you edit the source code later. Comments ignored by browsers. CSS comments are embedded within the `<style>` element, and begin with `//` and end with `*/`:

Example:

```
// * This is a one-line comment * /
p {
  color: red;
}
```

You can add comments where you want to code:

Example:

```
p {
  color: red; // * Set text color in red * /
}

// * This
many lines
comment * /

p {
  color: red;
}
```

HTML AND CSS COMMENTS

In the HTML lesson, you learned that you can add comments to your HTML source using `<!-- -->` syntax. In the example, we use a mixture of HTML and CSS comments.

CSS COLORS

Colors are specified using predefined color words, or values of RGB, HEX, HSL, RGBA, and HSLA. In CSS, color can be defined using the predefined

color name. Use the following color picker and color charts to select a color to use within your CSS code. The color selector provides color values for hexadecimal and RGB. The color chart provides the hexadecimal, RGB, and their color names.

Basic CSS Colors

Here is a table of basic color names that have been in CSS since CSS1. Try clicking a value. This will open the color in the online editor of Quick to see what it looks like. There is a link to the color checker where you can get another color preview.

CSS Color Names

Here is the full list of CSS color labels as specified in CSS3. These are based on the X-11 color set. The basic colors above are included in this chart. Try clicking a value. This will open the color in the online editor of Quick to see what it looks like. There is a link to the color checker where you can get another color preview.

How to Use These Colors

Each color can be represented in various different ways. For example, blue can be represented as # 0000ff, # 00f, RGB (0, 0, 255), and other ways. It does not matter which one you choose as long as it is a valid color. You can apply any of the colors to a website or blog by using the appropriate CSS code.

- To set the background color, use the background color.
- To set text color, use color.
- To set a border color, use the border color.

In Hexadecimal Notation

Here is the same color, but in this case, you are using hexadecimal notation instead of the color name. So, the color code now says 0000ff. This should be started with a hashtag, so it will be # 0000ff if it is coded.

In RGB Functional Notation

Here is the same color, but in this case, the active RGB commentary is used. So the color code is 0, 0, 255. This needs to be included within the RGB () function, so it will look like this RGB (0, 0, 255).

List of Color Properties

Here's a list of color related properties you can use with CSS.

- background
- border
- border-bottom-color
- border-color
- border-left-color
- border-right-color
- background-color
- opacity
- outline-color
- outline
- border-top-color
- box-shadow
- caret-color
- color
- column-rule
- column-rule-color
- filter
- text-decoration
- text-decoration-color
- text-shadow

List of Color Values

Here's a list of color values can use with CSS.

- Named Colors
- transparent Keyword
- CS current color Keyword
- 3-Digit Hex Codes
- 4-Digit Hex Codes
- 6-Digit Hex Codes
- CSS 8-Digit Hex Codes
- RGB() Function
- rgba() Function
- HSL() Function
- hsla() Function
- hwb() Function
- System Colors

CSS BACKGROUNDS

The background properties are used to add background effects for any elements. In these chapters, you will get to learn about the CSS background properties:

- background-image
- background-repeat
- background-attachment
- background-color
- background (shorthand property)
- background-position

CSS BACKGROUND COLOR

The background color feature specifies the background color of the element.

With CSS, color is usually specified:

- valid color name: such as 'red'
- HEX value: such as '# ff0000'
- RGB value: such as 'rgb (255,0,0)'

Transparency/Opacity

The invisible element defines the transparency/lightness of the element. It can take any value from 0.0 to 1.0. Low value indicates this.

```
.div {
  background-color: green;
  opacity: 0.3;
}
```

To Demonstrate This Using the RGBA

If you don't want to use opacity on child objects, such as in our example above, use RGBA color values. The following example sets for the background color and not the text.

You learned in the CSS Colors chapter, that you can use RGB as a color value. In addition to the RGB, you can use the RGB color value with the alpha channel (RGBA) – which determines color fading.

The color value of RGBA is specified as rgba (red, green, blue, alpha). The alpha parameter is a number between 0.0 (shows across fully) and 1.0 (fully opaque).

CSS Background-Image

The background-image property defines an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

Example:

Set the background image for a page:

```
body {
  background-image: URL("paper.gif");
}
```

CSS Background-Repeat

By default, the background image repeats the image horizontally and vertically. If the image is repeated only horizontally (background: repeat-x;), the background will look better:

Example:

```
body {
  background-image: URL ("gradient_bg.png");
  background-repeat: multi-x;
}
```

CSS Background-Repeat: No-Repeat

It shows the background image once it is specified by the background-repeat property. Show the background image only once:

```
.div{
  background-image: URL("img_tree.png");
  background-repeat: no-repeat;
}
```

CSS Background-Attachment

The background-attachment specifies that the background image must be scrolled or adjusted.

Example:

It specifies whether the background image needs to be adjusted:

```
body {
  background image: URL ("img_tree.png");
  background repetition: no-repeat;
  background: top right;
  background-attachment: fixed;
}
```

CSS BORDERS

The CSS border properties allow to specify the style, width, and color of an element's border.

CSS Border Style

The border-style property specifies the kind of border to display. The following values are allowed:

- dotted: It defines a dotted border.
- dashed: It defines a dashed border.
- solid: It defines a solid border.
- double: It defines a double border.
- groove: It defines a 3D grooved border. The effect depends on the border-color value.
- ridge: It defines a 3D ridged border. The effect depends on the border-color value.
- inset: It defines a 3D inset border. The effect depends on the border-color value.
- outset: It defines a 3D outset border. The effect depends on the border-color value.
- none: It defines no border.
- hidden: It defines a hidden border.

CSS Border Width

The border-width property defines the width of the four borders. The width can also be set as a specific size (in px, pt, cm, em, etc.) or by using one of the three predefined values: thin, medium, or thick.

Example:

```
p.one {  
  border-style: solid; //You can add your border  
  width here  
  border-width: 5px;  
}
```

The `border-color` property is used to set the color of the four borders. The color can be set by:

- name: It specifies a color name, like 'red'
- HEX: It specifies a HEX value, like '#ff0000'
- RGB: It specifies a RGB value, like 'rgb(255,0,0)'
- HSL: It specifies a HSL value, like 'hsl(0, 100%, 50%)'
- Transparent

HEX Values

The color of the border can be specified using a hexadecimal value known as (HEX):

Example:

```
p.one {
  border-style: dashed;
  border-color: #ff0000; /* red */
}
```

CSS Margins

CSS genetic features are used to create space around elements, without any defined parameters. With CSS, you have full control over margins. There are features for setting the margin for each element of the element such as top, right, bottom, and left. CSS has features that specify a margin for each element of the element:

- margin-top
- margin-right
- margin-bottom
- margin-left

All genetic features can have the following values:

- default: The browser calculates limit.
- length: It specifies margin in px, pt, cm, etc.

- %: It specifies margin in% of the width of the content.
- inheritance: It specifies that the margin should be inherited from the parent.

CSS Padding

CSS finishing features are used to generate space near element content, within any defined parameters. With CSS, you have complete control over the pads. There are pads setup properties on each side of the element (top, right, bottom, and left). CSS has paddling specifications for each element of the element:

- padding-top
- padding-right
- padding-floor
- padding-left

All completion buildings may have the following values:

- length: It specifies padding in px, pt, cm, etc.
- %: It specifies padding in% of the width of the content
- inheritance: It specifies that padding should be inherited from the parent element

Padding and Element Width

The CSS width feature specifies the width of the element content area. Content area is the inner part of the pad, border, and object genes (box model). Therefore, if the element has a specified width, additional padding on that feature will be added to the full feature width. This is often an unwanted consequence.

Example:

Here, the <div> section is given a width of 300px. However, the actual width of the <div> feature will be 350px (300px+25px for left padding+25px for right attachment):

```
div {
  width: 300px;
  padding: 25px;
}
```

CSS LENGTH, WIDTH, AND MAX-WIDTH

CSS templates and widths are used to set the length and width of an object. The maximum CSS feature is used to set the maximum object size.

Length and Width of CSS Values

Buildings of length and width may have the following values:

- default: This is the default. Browser calculates length and width.
- height: It specifies length/width in px, cm, etc.
- %: It specifies the length/width in percentage of the block containing.
- start: It sets the length/width over its default value.
- inheritance: The length/width will be derived from its parent value.

CSS BOX MODEL

The CSS box model is a box that wraps everything HTML. Contains: margins, borders, padding, and original content. The image below shows the box model. In CSS, the term “box model” is used when referring to design and structure.

Description of different parts:

- Content: The contents of the box, in which the text and images appear
- Padding: Clears the area around the content. The padding is obvious
- Border: A border around padding and content
- Margin: It can clear an area outside of the border. The margin is clear.

CSS Layout – Location Area

The spatial structure specifies the type of position used for a feature (vertical, related, fixed, complete, or sticky).

The Position Property

It specifies the type of configuration used for the feature. There are five numbers for different locations:

- static
- relative
- fixed
- absolute
- sticky

The elements are then arranged using the upper, lower, left, and right structures. However, these structures will not work unless the location of the position is pre-set. They also work differently depending on local value.

CSS TABLES

The table in CSS is used to apply various style elements to HTML Table elements to organize data in rows and columns, or possibly in a more complex format. Tables are widely used in communication, research, and data analysis. The table layout in CSS can be used to display table layouts. This feature is used primarily to set the algorithm used to edit <table> cells, rows, and columns.

Border: Used to specify parameters in a table.

Syntax:

```
border: table_width table_color;
```

CSS Layout – The Display Property

The display feature specifies/how the element is displayed. Everything HTML has a default display value depending on the type of element. The default display value for most items is a block or queue.

BLOCK-LEVEL ELEMENTS

The block element always starts in a new line and takes the full range available (extend as far left and right as possible).

Examples of block-level elements are as follows:

- <div>
- <h1>–<h6>

- `<p>`
- `<form>`
- `<title>`
- `<foot>`
- `<section>`

Inline Elements

The inline element does not start in the new line and takes up as much width as needed. This is the `<span>` line item within the paragraph.

Examples of inline elements are as follows:

- `<span>`
- `<a>`
- `<img>`

CSS LAYOUT – z-INDEX LAYOUT

The z-index feature specifies the layout of the element. When elements are suspended, they can override other features. The z-point feature specifies the layout of the element.

Note: z-index works on positioned elements such as absolute, position: relative, position: fixed, or position: sticky, and flex items.

Without z-Index

If two overlap elements pass without the z-index, the feature defined at the end in the HTML code will be displayed at the top.

CSS LAYOUT – WIDTH AND MAX-WIDTH (USING WIDTH, MAX-WIDTH, AND MARGIN: AUTO;)

A block-level element takes up the full width available of the screen. It can set the width of a block-level element and will prevent it from stretching out to the edges of its container. You can set the margins to auto, to center the element within its container. The element will take up the width, and the remaining space will be split equally between the two margins. Where using max-width will improve the browser's handling of small windows. This site is usable on small devices.

CSS LAYOUT – OVERFLOW

An overflowing CSS layout controls what happens to content that is too large to enter. The overflowing feature specifies that content must be pasted or added to scroll bars where the content of the element is too large to fit into a specified area.

An overflowing building has the following values:

- visible: It belongs. Content is provided outside the element box.
- hidden: The overflow is clipped, and the rest of the content will be invisible.
- scroll: In this overflow, scrollbar is added to see the rest of the content.
- auto: The same as scroll, but add scroll bars only if necessary.

CSS LAYOUT – FLOAT AND CLEAR

The float specifies how an element should float. The clear specifies what elements can float beside the cleared element and on which side.

The Float Property

It is used for positioning and formatting content, e.g., let an image float left to the text in a container. The float property can have one of the following values:

- left: It floats element to the left of its container.
- right: It floats element to the right of its container.
- none: It element does not float. This is default.
- inherit: It inherits the float value of its parent.

CSS3 LAYOUTS

W3C is currently running in CSS3. It contains different independent modules, each running a specific CSS field. For example, there are CSS Color Module, CSS Backgrounds and Borders Module, CSS Fonts Module, etc. Each of these modules is in its own way of developing itself, therefore, it can have numbers of different levels. For example, one may be level 3, while the other may be level 4. Regardless, they are all considered 'CSS3'.

Here is a separate list of CSS3 layouts. These are some of the new layouts introduced in CSS with its various modules.

How to Use CSS Properties

CSS layouts are used to apply styles to structured documents, such as those created in HTML or XML. CSS layouts are specified in the CSS level. Each property has a set of some possible values. Some properties can affect the type of element, while others apply only to certain groups of elements. The style sheet contains a list of rules. Each rule or set of rule contains one or more selectors, as well as a declaration block.

CSS features are used within the ad with the corresponding value.

Example:

```
section {
    padding: 20px;
    margin: 20px;
    background-color: cornsilk;
    border: 6px solid gold;
}
h1 {
    color: coral;
}
p {
    color: orange;
}
a:link,
a:visited {
    color: dark orange;
}
a:hover {
    color: orangered;
}
```

CSS FUNCTIONS

CSS functions are used as a number of different CSS structures. For example, you can use RGB () function to assign a color value (similar to the color: RGB (255, 0, 215)) or the attr () function to retrieve the HTML attribute value. Many functions are used in CSS conversion. For example, a rotating function () could be used to rotate an element, a scale () function could be used to resize an object, and a translate() function can be used to move an element.

There are other useful functions, such as a circle () to attach an object to a circle or to create a circle for text to flow around, as well as a counting function () that can be used to give a calculated value to a plot.

Here is a list of CSS functions in CSS3:

- attr()
- blur()
- brightness()
- calc()
- circle()
- contrast()
- counter()
- counters()
- cubic-bezier()
- drop-shadow()
- ellipse()
- filter()
- grayscale()
- HSL()
- hsla()
- hue-rotate()
- hwb()
- image()
- inset()
- invert()
- linear-gradient()
- matrix()
- matrix3d()

- `opacity()`
- `perspective()`
- `polygon()`
- `radial-gradient()`
- `repeating-linear-gradient()`
- `repeating-radial-gradient()`
- `RGB()`
- `rgba()`
- `rotate()`
- `rotate3d()`
- `rotateX()`
- `rotateY()`
- `rotateZ()`
- `saturate()`
- `sepia()`
- `scale()`
- `scale3d()`
- `scaleX()`
- `scaleY()`
- `scaleZ()`
- `skew()`
- `skewX()`
- `skewY()`
- `symbols()`
- `translate()`
- `translate3d()`

- `translateX()`
- `translateY()`
- `translateZ()`
- `URL()`
- `var()`

CSS FLEXBOX

The flexbox CSS layout allows you to easily format HTML. Flexbox makes it easy to direct objects horizontally and horizontally using lines and columns. Items will ‘change’ into different sizes to fill the space. It makes responsive design easier.

CSS flexbox is ideal for standard website design or your application. It is easy to read, supports all modern browsers, and does not take much time to get the basics right. By the end of this guide, you will be ready to start using flexbox for your web projects.

Flexbox is a 1D layout for arranging objects in rows or columns. Items are flexible to fill in the blanks or to shrink into smaller spaces.

The flex container properties are as follows:

- `flex-direction`: It that specifies which direction the main axis runs.
- `flex-wrap`
- `flex-flow`
- `justify-content`: It controls where the flex items sit on the main axis.
- `align-items`: It controls where the flex items sit on the cross axis.
- `align-content`:

Why Flexbox?

For a long time, the only reliable browser-compatible tools available for creating CSS layouts were features such as floats and positioning. These work, but in some ways, they are limited and frustrating.

The following building designs are difficult or impossible to achieve with such tools in any form of simple, flexible approach:

- Vertically centering a block element of content inside its parent
- Making all the children of a container take up the same amount of the width/height
- Making all columns in a multiple-column layout with the same height even if they contain a different amount of content

CSS BOX SIZE

The CSS box size feature allows us to add padding and border to the object's width and length.

When you set the width/length of an element, the feature often appears larger than the one you set. The positioning of the box allows us to include the pad and the border in the value of the width and length of the object.

CSS Syntax

- `box-sizing`: Properties.
- `content-box`: It is default. The width and height properties includes only the content.
- `border-box`: The width and height properties includes content, padding, and border.
- `initial`: It sets this property to its default value.
- `inherit`: It inherits property from its parent element.

CSS GRID LAYOUT

CSS Grid Layout can divide a page into major regions or define the relationship in terms of size, position, and layer, between parts of a built from HTML.

Like tables, grid layout enables the writer to align elements into columns and rows. However, many layouts are possible or easier with the CSS grid than tables. For example, children's grid container items can be set to overlap and layer such as CSS-set elements.

CSS MEDIA TYPES

Introduction to Media Types

One of the most important features of stylesheets is that you can specify different style sheets for different types of media. This is one of the best ways to create webpages ready for printer – it just provides a separate style sheet for the ‘print’ media type.

Some CSS features are for media only. For example, the page sorting feature only applies to pages with pages. There are a few things that can be shared by different media types, but they may require different values for that feature. The font size feature, for example, can be used for both screen and print media, but probably at different values.

Documentation often requires a large font on a computer screen compared to paper for better reading, and sans-serif fonts are considered easy to read on screen, while serif fonts are famous for printing. It is therefore necessary to clarify that a style sheet, or a set of style rules, applies to certain types of media.

Method 1: Using @media At-rules

The @media rule is used to define different style rules for different media types in a single style sheet. It is usually followed by a comma-separated list of media types and a CSS declaration block containing targeted media style rules.

Example:

The style tells the browser to display body content by 14 pixels Arial font on the screen, but when printing, it will be in Times font 12 points.

```
@media screen {  
    body {  
        color: black;  
        font-size: 12px;  
    }  
}  
  
@media print {  
    body {  
        color: # fff;  
        font-size: 10pt;  
    }  
}
```

```
@media screen, print {
    body {
        line-length: 1;
    }
}
```

Method 2: Using @import At-rules

The @import rule is another way to specify specific media style details – it just specifies comma-separated media types after the URL of the imported style sheets.

Example:

```
@import URL ("CSS / screen.css") screen;
@import URL ("CSS / print.css") print;
div {
    background: # f5f5f5;
    line length: 1.2;
}
```

Method 3: Using the <link> Element

The media attribute in the <link> feature is used to specify targeted media on an external style sheet within an HTML document.

```
<link rel = "stylesheet" href = "css / common.css"
media = "all">
<link rel = "stylesheet" href = "css / print.css"
media = "print">
```

DIFFERENT MEDIA TYPES

The following table lists the various media types that may be used to target different types of devices such as printers, handheld devices, and computer screens.

- all: It is used for all media type devices.
- aural: It is used for speech and sound synthesizers.

- braille: It is used for tactile feedback devices.
- embossed: It is used for paged braille printers.
- handheld: It is used for small or handheld devices – usually small screen devices such as mobile phones or PDAs.
- print: It is used for printers.
- projection: It is used for projected presentations, for example projectors.
- screen: It is used primarily for color computer screens.
- tty: It is used for media using a fixed-pitch character grid – such as teletypes, terminals, or portable devices with limited display capabilities.
- tv: It is used for television-type devices – low resolution, color, limited-scrollability screens, sound available.

MEDIA QUESTIONS AND RESPONSIVE WEB DESIGN

Media queries allow you to customize the presentation of webpages to a specific range of devices such as mobile phones, tablets, and desktops without change in markups. A media query contains media type and zero or more expressions such as the type and conditions of certain media features such as device width or screen resolution. Here's a simple example of a typical media query for standard devices.

MEDIA QUERIES

Responsive design was able to emerge due to media query. The definition of Media Queries Level 3 was Candidate Recommendations in 2009, which means it was considered ready for implementation in browsers. Media queries allow us to create a series of tests and apply CSS by choosing to customize the page style to the user's needs.

For example, the following media quiz questions to see if the current webpage is displayed as screen media (so it is not a printed document) and the viewing area is 800 pixels wide. CSS selector `.container` will only be used if these two are true.

```

/* Smartphones (portrait and landscape) ----- */
@media screen and (min-width: 320px) and (max-width:
480px) {
    /* styles */
}
/* Smartphones (portrait) ----- */
@media screen and (max-width: 320px) {
    /* styles */
}
/* Smartphones (landscape) ----- */
@media screen and (min-width: 321px) {
    /* styles */
}
/* Tablets, iPads (portrait and landscape)
----- */
@media screen and (min-width: 768px) and (max-width:
1024px) {
    /* styles */
}
/* Tablets, iPads (portrait) ----- */
@media screen and (min-width: 768px) {
    /* styles */
}
/* Tablets, iPads (landscape) ----- */
@media screen and (min-width: 1024px) {
    /* styles */
}
/* Desktops and laptops ----- */
@media screen and (min-width: 1224px) {
    /* styles */
}
/* Large screens ----- */
@media screen and (min-width: 1824px) {
    /* styles */
}

```

CSS ATTRIBUTE SELECTORS

The attribute selector selects HTML elements with a specific attribute or attribute with a specified value.

Syntax:

[attr]

It represents elements with an attribute name of `attr`.

```
[attr=value]
```

It represents elements with an attribute name of `attr` whose value is exactly `value`.

```
[attr~=value]
```

It represents elements with an attribute name of `attr` whose value is a whitespace-separated list of words, one of which is exactly `value`.

```
[attr|=value]
```

It represents elements with an attribute name of `attr` whose value can be exactly `value` or can begin with `value` immediately followed by a hyphen, - (U+002D). It is often used for language subcode matches.

```
[attr^=value]
```

It represents elements with an attribute name of `attr` whose value is prefixed (preceded) by `value`.

```
[attr$=value]
```

It represents elements with an attribute name of `attr` whose value is suffixed (followed) by `value`.

```
[attr*=value]
```

It represents elements with an attribute name of `attr` whose value contains at least one occurrence of `value` within the string.

```
[attr operator value i]
```

Adding an `i` (or `I`) before the closing bracket causes the value to be compared case-insensitively (for characters within the ASCII range).

```
[attr operator value s] Experimental
```

Adding an `s` (or `S`) before the closing bracket causes the value to be compared case-sensitively (for characters within the ASCII range).

CSS FRAGMENTATION

It is a CSS module that explains how content is displayed when broken (separated) into multiple pages, regions, or columns. Fragmentation occurs when the inline box in the row folds (wrap) in multiple rows. It also happens when a block covers more than one column inside a column structure container, or closes a page break when printing. Each rendering piece of a feature is called a fragment.

Types of Fragmentation

- box-decoration-break
- break-after
- break-before
- break-inside
- orphans
- widows

Understanding Attribute Selectors

CSS attribute options provide an easy and powerful way to apply styles to HTML objects based on the presence of a specific attribute or attribute value. You can create an attribute selector by inserting an attribute – optional with value – in square brackets. You can also put an element type selector in front of it.

The following sections describe the most common attribute choices:

Some HTML elements have attributes. For example, to create a link using an object with a href attribute, the href value of the link the browser will follow when clicked. Form fields use most of the attributes. For example, an input item uses an attribute type to indicate that a field is a text, email address or checkbox.

The attribute selectors give us the ability to identify an object based on the attributes it possesses. When we style text input, we usually do not want the same styles to apply to the checkbox. If we intend to install, that will happen. We can add a class to the elements we want

to style; however, attribute selectors can help us choose the right things we want to style and mean we can avoid adding a class for this purpose.

The two elements below are a form element with a text input type, which will create a text input field, and a check box to create a checkbox. As you can see, they both use the input element.

Selecting and styling text input only uses the attribute selector, adding the attribute and value within the square brackets.

Example:

```
input [type = "text"]
```

Choose an adjective word

In our example above our selector, we can compare the attribute name of the type with the exact value of the text. If we already wanted to select every feature installed that had the attribute, regardless of the value we could use.

```
input [type]
```

Selecting from a List of Values

Some attributes can take a price list separated by a white space character. Let us take classes for example, in the following example we have a class box and a class warning as class attribute values.

```
<p class = "box alert"> </p>
```

To select any component with a box category, even if it is in the class list, we can associate this with the following selector:

```
* [class ~ = box]
```

Selecting a Value That Starts with a Character Unit

You can choose from the values that start, end, and consist of a few small series. To match all links starting with 'HTTPS', we will use the selector:

```
a [href ^ = "HTTPS"]
```

Matching HTTP AND HTTPS links:

```
a [href ^ = "HTTP"]
```

Only matching items on the same page:

```
a [href ^ = "#"]
```

Selecting the Ending Value per Character Unit

You can also select a value if it ends with a specific series. This selector can only select links that end in.com.

```
a [href $ = ". com"]
```

Selecting a Value That Contains a Character Unit

The selector above that selects the links ending in.com will not hold a link to another page of the site, or if a/is inserted at the end of the domain. To catch all of these use a substring match.

```
a [href * = ". com"]
```

Case Insensitivity

By default, all of the matches are case sensitive. If you make any match case insensitive, use the i operator before the closing bracket. This selector would match values starting with HTTP or HTTP or HTTP for that matter!

```
a [href ^ = "HTTP" i]
```

Select a Value or a Value Followed by a -

A use case for this is in selecting language subcodes. This selector would match en-US and en-GB.

```
span [lang | = "en"]
```

BENEFITS YOU MAY KNOW

Let's learn more about CSS benefits such as:

1. Every website has content and has a predefined structure or structure. This structure is used for presenting content. This is the main purpose of CSS, and it separates the document structure from the text presentation. HTML is about structure and nothing else as most of us know this basic meaning of HTML. With CSS, the user can make quick changes if needed, but this is not an easy task with HTML. Previously the background color for my webpage was red (code will be #ff0000). If a developer wants to change any design color, he or she should simply edit one line of code that will appear on any number of webpages. He will (code #eee for light gray) change this and do his job.
2. A little use of coding will increase page efficiency and reduce loading time. Overload time users get frustrated. What really happens in coding is interesting to know and how CSS is a savior. To illustrate a table field with a built-in code, this makes the code reading twice. The first is to understand the basic structure of the table, and the second is to really reflect the content within the table. This duplicate operation may delay website loading. Now, let us discuss the benefits of CSS, although the images are always larger than the largest code used or discussed in the above reference as we applied CSS to a separate file, the built-in CSS code format will be saved after the initial application and is not required to download the page again. This is one of the most important CSS behaviors over HTML.
3. Your presence in the digital market, aka the internet, is a very important way of looking at a business. CSS is considered one of the safest ways to craft web development, and it means that your website has more content than code. One of the benefits of CSS is that CSS is very useful for digital marketing purposes where every promotion of your webpage provides business opportunity. Therefore, more content with a code formula with the help and use of CSS techniques is better for digital marketing and your business in the end.
4. A consistent approach is something that CSS offers its users, and by making small changes to your website, similar changes are seen in other parts of the website. The bigger your website, the more content and layouts are where CSS saves you a lot of time when it comes to conversions. It also checks and verifies that each page is consistent with the CSS benefits.

5. W3C works with the goal of making the entire website equally accessible to disabled users the way normal users use it. First CSS gains access by creating layout and document making it easy for any screen readers to read content accurately. This is useful for deaf users as they rely on the screen reader to run web applications.
6. For a website to work properly, it must have a fast download time. Nowadays, people usually wait a few seconds for a website to load. Therefore, it is important to ensure a fast pace. For companies looking to ensure fast and smooth website information, CSS becomes key to their success.
7. CSS is easy to maintain due to the short maintenance time. This is because single-line code conversion affects the entire webpage. Also, if upgrades are needed, a little more effort is needed to minimize changes to the webpage code.
8. You would not find many good and easy-to-use websites. One thing that is common to all of these websites is the consistency of construction. CSS empowers developers to ensure that style features are applied consistently across every few webpages.
9. People use different smart devices to view a particular website. It can be a smartphone, PC, or laptop. For this purpose, websites are required to be compatible with the device. CSS ensures smooth operation by providing better alignment.
10. Due to its fast speed and easy maintenance, CSS saves a lot of time and effort in the web development process due to fast loading time. Here, a little time ensures the good performance of the designer.

CSS Cons

There are a few drops while using CSS. One has to be aware of these bad situations in order to know and take care of them while designing a website.

1. Confusion due to too many CSS levels: Beginners are at high risk in this regard. They may be confused when choosing to read CSS as there are many CSS levels like CSS2, CSS3, etc.

2. Browser problems: Different browsers work differently. Therefore, you should check if the changes made to the website by the CSS codes are visible to all browsers.
3. Security issues: Security is critical in today's world driven by technology and data. One of the worst things about CSS is that it has limited security.
4. Additional developer work: Design services are required to consider and test all CSS codes in all different browsers for compatibility. As a result of testing developer compatibility in different browsers, their workload is growing.

All in all, we can say that if you like web development, try learning HTML and CSS. For device compatibility, read the Bootstrap framework. While you may see some disadvantages of CSS, many of the benefits are counter-productive and make sure your web development process is smooth and efficient.

CHAPTER SUMMARY

Here we have covered all the basic concepts of CSS that everyone can use to make their pages attractive. Then, in the coming chapter, you will learn CSS framework that makes your work so easy.

FURTHER READING

A Look Back at the History of CSS - <https://css-tricks.com/look-back-history-css/>, accessed on (2022 May 16).

Advantages and Disadvantages of CSS - <https://unstop.com/blog/advantages-and-disadvantages-of-css>, accessed on (2022 May 19).

Advantages of CSS - <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-css/#:~:text=Advantages%20of%20CSS%3A&text=Web%20designers%20needs%20to%20use,web%20site%20and%20maintenance%20time>, accessed on (2022 May 19).

Beginner's Guide to CSS Classes - <https://blog.hubspot.com/website/what-is-css-class>, accessed on (2022 May 17).

Brief history of CSS - <https://www.w3.org/Style/CSS20/history.html>, accessed on (2022 May 16).

Brief Overview - <https://simplecss.eu/css-history-brief-overview.html>, accessed on (2022 May 16).

Comments - <https://developer.mozilla.org/en-US/docs/Web/CSS/Comments#:~:text=A%20CSS%20comment%20is%20used,the%20layout%20of%20a%20document>, accessed on (2022 May 17).

CSS - <https://en.wikipedia.org/wiki/CSS>, accessed on (2022 May 16).

- CSS - <https://www.quackit.com/css/>, accessed on (2022 May 18).
- CSS Attribute Selectors - https://www.w3schools.com/css/css_attribute_selectors.asp, accessed on (2022 May 18).
- CSS bgcolor - https://www.quackit.com/css/css_bgcolor.cfm, accessed on (2022 May 18).
- CSS Box Model - https://www.w3schools.com/css/css_boxmodel.asp, accessed on (2022 May 18).
- CSS Color Properties - <https://www.quackit.com/css/color/properties/>, accessed on (2022 May 18).
- CSS declarations - <https://developer.mozilla.org/en-US/docs/Web/CSS/Syntax>, accessed on (2022 May 17).
- CSS Layout - The display Property - https://www.w3schools.com/css/css_display_visibility.asp, accessed on (2022 May 18).
- CSS selectors - https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Selectors, accessed on (2022 May 17).
- CSS Syntax - https://www.w3schools.com/css/css_syntax.asp, accessed on (2022 May 17).
- CSS Tables - <https://www.geeksforgeeks.org/css-tables/>, accessed on (2022 May 18).
- CSS/Properties/color/keywords - <https://www.w3.org/wiki/CSS/Properties/color/keywords>, accessed on (2022 May 18).
- CSS: Cascading Style Sheets - <https://developer.mozilla.org/en-US/docs/Web/CSS>, accessed on (2022 May 16).
- Document and website structure - https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure, accessed on (2022 May 19).
- Getting Started with CSS Rules - <https://learn.saylor.org/mod/book/view.php?id=36821&chapterid=20180>, accessed on (2022 May 17).
- History of CSS - <https://www.bu.edu/lernet/artemis/years/2020/projects/FinalPresentations/HTML/historyofcss.html>, accessed on (2022 May 16).
- How CSS works - https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works, accessed on (2022 May 17).
- How do you add a class to the paragraph element? - <https://teamtreehouse.com/community/how-do-you-add-a-class-to-the-paragraph-element>, accessed on (2022 May 17).
- How does CSS actually work? - https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works#:~:text=It%20converts%20the%20HTML%20into,videos%2C%20and%20even%20linked%20CSS!, accessed on (2022 May 17).
- How to Link CSS to HTML Files in Wb Development - <https://www.hostinger.in/tutorials/website/how-to-link-a-stylesheet-css-file-to-your-html-file>, accessed on (2022 May 17).
- HTML & CSS - <https://www.w3.org/standards/webdesign/htmlcss>, accessed on (2022 May 16).

Introduction to HTML/CSS - https://www3.ntu.edu.sg/home/ehchua/programming/webprogramming/HTML_CSS_Basics.html, accessed on (2022 May 16).

Media types - <https://www.w3.org/TR/CSS21/media.html%23media-types>, accessed on (2022 May 18).

Responsive Web Design - <https://climbtheladder.com/responsive-web-design-interview-questions/#:~:text=Media%20queries%20are%20a%20key,all%20in%20the%20same%20document>, accessed on (2022 May 19).

Responsive Web Design - https://www.w3schools.com/css/css_rwd_mediaqueries.asp, accessed on (2022 May 19).

Rules and Style Sheets - <https://www.w3.org/Style/LieBos2e/enter/Overview.en.html>, accessed on (2022 May 17).

Types of CSS - <https://www.hostinger.in/tutorials/difference-between-inline-external-and-internal-css#:~:text=Types%20of%20CSS%3A%20Inline%2C%20External%20and%20Internal%20Definitions%20and%20Differences%20Explained&text=Copied!,HTML%20and%20CSS%20files%20separately>, accessed on (2022 May 17).

NOTE

-
- 1 Different versions of CSS – PCBs.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Bootstrap

IN THIS CHAPTER

- Introduction
- Versions of Bootstrap
- Bootstrap installation
- Usage of Bootstrap

INTRODUCTION

Every website on the web uses HTML, CSS, and JavaScript at its core. There are many frameworks available these days that enhance the functionality or visibility of any web page. However, the journey of all web developers starts with HTML, CSS, and JavaScript. The complete HTML form is Hypertext Markup Language and provides the basic structure of any web page. The full form of CSS is Cascading Style Sheets and is used to style web pages. CSS adds beauty to an HTML page that looks blank. JavaScript is also used to add functionality to any web page. Now, this is where Bootstrap starts playing. Building a fully functional website from scratch from these three can be a daunting task. It is very time consuming too. Bootstrap not only solves this problem but also makes the whole process easier. So, what exactly is Bootstrap? Bootstrap is a CSS framework that helps us to create easy-to-use and responsive mobile web pages

without any hassle. A framework can be any software that provides standard functionality as well as other add-ons and features that allow users to customize and adapt according to their needs.

Basic HTML with CSS knowledge is required to learn Bootstrap and eventually become an expert. Bootstrap is a framework that contains HTML and CSS at its core, so understanding that is both important. It has descriptive features that anyone can quickly read and apply. Bootstrap is also completely free and open source, which means that the source code is open and easily accessible online for you to access and edit if you wish to do so. Making any software open source allows for open collaboration, allowing users to contribute to software development.

Bootstrap is focused on early mobile development. Mobile-first development means that when building any web page using Bootstrap, it focuses on how it is done on mobile devices. From there, user interaction is measured in terms of the larger screens. This method is useful as many users around the world access the internet using their mobile phones. Putting their needs first and ensuring that the website works well on any mobile device can increase traffic to any website.

BOOTSTRAP HISTORY

Mark Otto and Jacob Thornton first did Bootstrap. It was released for the first time more than a decade ago, in August 2011. Since then, many changes have taken place, including the release of improved versions. The latest stable version was released in October 2021, and it is called Bootstrap 5. It is now one of the most popular open-source projects and front-end agencies around the world.

Originally, it was called the Twitter Blueprint as both Mark and James used to work on Twitter and wanted to create a framework that would help them maintain consistency throughout the universe. However, using multiple different libraries to create user interactions has caused inconsistencies and variations across the universe. It also seemed difficult to keep the code. The two engineers and designer then came up with Bootstrap. After a few months of initial development and recognition of innate power, many developers began to contribute to it during the Twitter week of hacking. It was then renamed Bootstrap and released as an open-source project for reuse. Mark and Jacob, the original creators, continue to care for it and the team of engineers and the contribution made by its large community of users. Also, it is still used as a style guide on Twitter for the development of internal tools, which was originally intended.

VERSIONS OF BOOTSTRAP

Bootstrap 1: Bootstrap 1 is the original version that was released in August 2011. After its creation, developers used it internally at Twitter, and then, later on, it was released in the public domain as free and open-source software.

Bootstrap 2: The second version of Bootstrap or Bootstrap 2 was first released for public usage in January 2012. Bootstrap made several notable changes in this version. The team also added lots of new components and features to this version. One of the most notable ones among them was the responsive web design. Responsive web design means the styling or the layout of the page will adjust itself on its own across multiple devices like mobile phones, tablets, and desktops depending upon the device being used. The team added Glyph icons and built-in support in this version. Glyph icons are monochromatic icons that help in visually identifying features. There are different icons available for use, both free and paid.

Bootstrap 3: The third version of Bootstrap or Bootstrap 3 was first released for public usage in August 2013. This version was the one that introduced the shift in approach concerning mobile devices. Bootstrap started focusing more on mobile devices and adequately rendering websites on them from here onward. Shifting its approach to mobile devices first ensured their increased user base as most internet users worldwide are concentrated on mobile devices. This version also gave increased attention to ‘flat design’. Flat Design is essentially a design trend that is increasingly popular for its minimalist themes and outlook. The increase in the usage of Flat Design has significantly increased, and it can be seen being used in graphical user interfaces, also known as GUIs. It is also used in posters, guides, art documents, etc.

One other notable feature added was the new plugin system with namespaced events. Bootstrap 3 uses jQuery to add functionality onto any page, and one of the prominent features of jQuery is namespaced events. So, what exactly are namespaced events? We need to first learn about events and event listeners to know about that. Events are any action performed on a web page that triggers the response of another HTML element. Anything could be an event like clicking on a button to submit user information

in a form. There are different events like click, mouse hover, focus, and blur. There are various keyboard events as well that are triggered when a specific button is clicked on the keyboard. An event listener is an object that handles any event. It controls the response to an event in case an event is triggered.

Events with word spaces are effective when multiple event listeners are added to a single activity. In making one event, there are many events that happen. Depending on the requirements of the user and the user, if we wish to activate only one listener to a few, then events with word spaces are helpful. You can open or close the event audience according to user activity and customize it to get better user experience.

Bootstrap 3 also removed additional support for certain browser versions such as Internet Explorer 7 and Firefox 3.6.

Bootstrap 4: Bootstrap 4 was one of the most important versions of Bootstrap, and it completely changed its face. The team made a few major changes to this version. First, Bootstrap 4 was first announced by Mark Otto, one of its first creators, in October 2014. However, the first version of the alpha was released next year in August 2015. However, the company released a beta version in August 2017. Mark, the original creator, had completely stopped working on the previous version, Bootstrap 3 and focused entirely on Bootstrap 4. After four years of publicity, it was finalized and posted to the public domain in January 2018.

The first versions of Bootstrap were styled using LESS CSS. Less CSS is a pre-processed style sheet language that has a powerful processor in its path. The most common definition behind its function is compiled into cascade style sheets, also known as CSS, and runs on the client side, which is the front part where the user can access and appear. It can also be run on the server side, providing a style that can be part of the internal tools and specifications the user will not be able to access. Less CSS is designed by Alexis Sellier.

Bootstrap 4 completely replaced Less CSS and replaced it with Sass. SASS stands for amazing style sheets. It is also a preprocessor-based writing language, and as a minority, it is also compiled or translated into Cascading Style Sheets. One of the additions is that Sass has its own scripting language called Sass Script. Hampton Catlin designed it. Sass contains two different types of syntax, and the original syntax is called indented syntax.

Sass has a hand over Les in many different ways. One of the most important things is that Sass allows you to write pure, simple, and minimal code in CSS compared to minimal. It's fast and contains minimal code making it easy and a quick option for both. Not only that, it is stable and offers beautiful style features because it is an extension of CSS. It also helps designers to successfully create beautiful style templates in a short amount of time.

Bootstrap 4 saw the addition of another feature called 'Restart'. The restart was based on Normalize.css. Normalize.css is a small CSS file that helps to provide all HTML elements consistently everywhere. This sync helps keep one preview across all device types, browsers, and browser versions. This parallel browser compatibility is essential to maintaining the default style intended by the designer and engineer. The standard uses HTML 5, the latest and most widely accepted version of HTML. Restart is inspired by Normalize, which means duplication is one of the main features of Normalize restart. In restarting, the engineers decided to put all the feature-specific changes in a single file. This change gave Bootstrap its own look and feel. Using Reboot helped create a beautiful but straightforward codebase for building on it by users.

Bootstrap also decided to support the CSS Flexible Box, popularly known as CSS FlexBox. CSS FlexBox is a Layout module that helps to lay out content and put you in touch with a popular layout. It allows us to classify our content and classify it according to available space. Build a visual connection and align objects to make them look and feel rich. Certain features are applied to the flexbox such as flex-direction, flex-wrap, flex-base, and flex-grow. We will discuss them in more detail as CSS FlexBox is an integral part of Bootstrap. Another feature that has helped you to customize it is to navigate through all the web pages. Additional navigation support across all pages has made it easier to switch between pages. Time taken back and forth simultaneously is reduced as a result. This version of Bootstrap also brought an important element in terms of classification and size. Bootstrap 4 added responsiveness to space and size. Another comment was presented on classification.

The text size is as follows:

- Xs: Very Small
- Sm: Small
- Md: Medium
- Lg: Great
- Xl: Very Large

BOOTSTRAP 5

Bootstrap 5 is the newer version of Bootstrap currently available. It was officially released in May 2021. The Bootstrap team has made a number of important changes to this version, just like its previous version. One of the most important changes was to completely undo its dependence on jQuery and to completely switch to vanilla JavaScript. Some new menu components have also been added, such as offcanvas. Offcanvas is a hidden side bar that a user can add to the website to enhance navigation skills. Requires a JavaScript plugin for more functionality. The sidebar is usually kept hidden until the event is activated. Then JavaScript is applied to the event audience, and a separate bar appears. Existing code to add the CSS grid was rewritten to achieve broader and more responsive CSS characters.

Documents transferred from Jekyll to Hugo. Both Jekyll and Hugo are consistent site producers. Jekyll is written in Ruby, while Hugo is written in the Go language. The company has chosen Hugo over Jekyll as it promises increased performance and higher stability compared to Jekyll. Similarly, Bootstrap has moved all the testing infrastructure from QUnit to Jasmine. Both of these are open-source JavaScript test frameworks. QUnit was originally developed to test jQuery but was later submitted to test JavaScript code.

A custom set of SVG symbols has also been added. Many custom CSS layouts have been added. The API has been greatly improved, and the forms are being updated. The grid system has been upgraded while customizing the available documents.

What can you do with Bootstrap?

- Create complete responsive websites in no time
- Create easy-to-use websites
- Fast and automatic upgrades depending on various websites
- Create dynamic web pages
- Manage large amounts of content
- Create toolkit and popover tips to show suggestion text
- Create a carousel and photo slide
- Create different types of forms and warning boxes

GETTING STARTED WITH BOOTSTRAP

Before we can start with Bootstrap we need to first configure our code editor. First and foremost, you will need a code editor to get started with Bootstrap. There are tons of free code editors available online and open source. You can choose any of these that suit your needs. Popular code editors are VS Code, Atom, Brackets, etc. You can use any of these code editors.

- Here, we will be using the VS Code. VS Code is one of the most popular and powerful source code creators and was created by Microsoft. It has many amazing tools like syntax highlighting, IntelliSense etc. Intellisense helps to complete intelligent code that helps to write pure code. It also has thousands of extensions that help you customize your code editor according to your needs. VS Code supports all types of web technologies including Bootstrap and is fully functional in web development. If you wish to download VS Code, you can download it from the following link: <https://code.visualstudio.com/download>.
- Atom is also the most powerful source code editor that works well in web technology. It is a free and open-source code editor for macOS, Linux, and Microsoft Windows. There is additional plugin support. Developed by GitHub. Here is the link to download the Atom: <https://atom.io/>. You can also download any other code editor of your choice and configure it according to the instructions provided on the official website.
- After setting your code editor, open the folder and create a file with the .html extension. Bootstrap actually provides us with an HTML and CSS coded code template, and all you have to do is copy those templates into their official website. Open your browser & go to the official Bootstrap website. Here is the link: <https://getbootstrap.com/>.
- There are two ways to get Bootstrap. You can include the Bootstrap 5 link using the CDN, or simply download the Bootstrap 5 website directly to your device using the package manager. You may be wondering which is better. Both options are good for their dignity, but it is best to use a CDN as a starting point.

HOW TO DOWNLOAD BOOTSTRAP USING PACKAGE MANAGER

You can download Bootstrap using different package managers like npm, thread, Composer. When you download Bootstrap, you will gain

access to Sass source and JavaScript files. Downloading Bootstrap using package manager requires you to have additional tools available. First, you will need a Sass compiler to merge all Sass source files into CSS files. You will also need to have an Autoprefixer for CSS vendor prefixing.

Downloading Bootstrap Using npm

If you think Node.js is already installed on your device, install Bootstrap using the npm package. Use the following command to install Bootstrap.

```
$ npm install bootstrap
```

npm install downloads package and its dependencies and can be processed with or without conflict. If you are using the package manager without conflicts npm install to download all the dependencies specified in the package. JSON will also generate a node_modules folder with the installation folder.

To download all Bootstrap plugins in the Bootstrap item use `const bootstrap=search ('bootstrap')` or `import bootstrap from 'bootstrap'`. The bootstrap module will automatically export all plugins. To download individual Bootstrap plugins individually, upload `/js/dist/*.js` files under the package directory.

Downloading Bootstrap Using a Cable

Use the following command to install Bootstrap using the cable package.

```
$ yarn add bootstrap
```

Download Bootstrap Using Composer

Use the following command to import and manage Sass and JavaScript files in Bootstrap using Composer.

```
$ composer require twbs/bootstrap:5.0.2
```

How to Download Bootstrap Using CDN

If you directly install Bootstrap directly on your device, you will also have access to Sass and JavaScript files found inside. You can download the same using package managers like npm, Composer.

If you choose to use a CDN, you can copy the link or template first and paste it into your code editor. CDN stands for Content Delivery Network. Bootstrap uses jsDelivr (<https://www.jsdelivr.com/>) as its content delivery network. JsDelivr is one of the most popular content delivery networks for open-source projects. The content delivery network helps to deliver web files directly from the npm registry and GitHub repository without any additional configuration on the user side.

What Is CDN?

CDN is known for Content Delivery Network & Content Distribution Network. It helps to improve the time and website performance. Bootstrap CDN is a free of cost content delivery network helps us to quickly load Bootstrap CSS, JavaScript, and jQuery libraries on projects to make projects responsive, mobile friendly, and attractive.

Bootstrap CDN Link and Scripts

For CSS

You just need to copy-paste the below link to the <head> section of your code.

```
<link href = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.5/css/bootstrap.min.css" rel =
"stylesheet" />
```

For JavaScript

```
<script src = "http://maxcdn.bootstrapcdn.com/
bootstrap/3.3.6/js/bootstrap.min.js"></script>
```

For JQuery Library

```
<script src = "http://ajax.googleapis.com/ajax/libs/
jquery/1.12.0/jquery.min.js"></script>
```

For Popper.js

```
<script src="HTTPS://cdn.jsdelivr.net/npm/popper.
js@1.14.7/dist/umd/popper.min.js" ></script>
```

How Do We Use Bootstrap CDN?

The below example shows that how we can use Bootstrap CDN on our code.

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title>This is a Bootstrap example of CDN </title >

  <meta initial-scale=1" name = "viewport" content =
"width = device-width">
  <link href = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.3.6/css/bootstrap.min.css" rel =
"stylesheet">
</head>
<body>
<div class = "container">
  <h1 align = "center"> Bootstrap CDN </h1>
  <p>Write your text here..</p>
</div>

  <script src = "http://ajax.googleapis.com/ajax/libs/
jquery/1.11.0/jquery.min.js"></script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.2.6/js/bootstrap.min.js"></script>
</body>
</html>
```

JsDelivr is the official Bootstrap CDN and helps to transfer Bootstrap CSS and JavaScript. One of the benefits of using a CDN is that it leads to faster loading time. As Bootstrap is a popular platform, many websites available online are created using Bootstrap. As a result, many users already have Bootstrap 5 from downloaded jsDelivr. When they visit your site, it will be loaded directly from the cache memory resulting in a faster upload time. Also, many content delivery networks ensure that whenever a user requests a file, the file is delivered from a server close to the user, which also results in faster upload time.

BOOTSTRAP TEMPLATE

Make sure you have your pages set with the latest design and development standards. That means using the HTML5 doctype and inserting a meta

tag for viewing for proper responsive behavior. It's all over and your pages should look like this.

```
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta content="width=device-width, initial-
scale=1, shrink-to-fit=no" name="viewport" >

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivrivr.
net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/
j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">

    <title>Hello, world!</title>
  </head>
  <body>
    <h1>Hello, world!</h1>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap
JS -->
    <script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7a
bK41JSTQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src="HTTPS://cdn.jsdelivrivr.net/npm/popper.
js@1.14.7/dist/umd/popper.min.js" integrity="sha384-
UO2eT0CpHqdsJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF8
6dIHNDz0W1" crossorigin="anonymous"></script>
    <script src="HTTPS://cdn.jsdelivrivr.net/npm/
bootstrap@4.3.1/dist/js/bootstrap.min.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjI
EaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  </body>
</html>
```

Is Popper.js Required for Bootstrap 4?

Tooltips rely on Popper.js for positioning. Popovers rely on Popper.js for positioning. Dropdowns are based on a third-party library called as Popper.js, that can provide dynamic positioning and viewport detection.

So these are the Bootstrap 4 features that require Popper.js:

- dropdowns
- popovers
- tooltips

RESPONSIVE META TAG

Bootstrap is developed for mobile first, a strategy in which we expand the code on mobile devices first and then expand sections as needed using CSS media queries. To ensure proper rendering and zoom touch on all devices, add a meta viewing meta tag to your <head>.

```
<meta name="viewport", content="width=device-width,  
initial-scale=1, shrink-to-fit=no">
```

HISTORY

Bootstrap, nicknamed the Twitter Blueprint, was developed by Mark Otto and Jacob Thornton on Twitter as a framework to promote consistency across all internal tools. Prior to Bootstrap, various libraries were used for visual enhancement, which resulted in significant overload and overhaul.

After a few months of small group development, many developers on Twitter began to contribute to the project as part of Hack Week, a hack-athon style week for the Twitter development team. It was renamed from Blueprint to Bootstrap and released as an open free source project on August 19, 2011. It continued to be kept by Mark Otto, Jacob Thornton, a small group of key developers, and a large community of participants.

Bootstrap 2

On January 31, 2012, Bootstrap2 was released, adding built-in support for Glyphicons, a few new features, and a wide range of existing features. This version supports responsive web design, that means that the design of web pages is dynamically interchangeable, taking into account the features of the device used, whether it is desktop, tablet, mobile.

Bootstrap 3

On August 19, 2013, Bootstrap 3 was released. Redesign the parts to use a flat structure and the original portable method. Bootstrap 3 installs a new plugin system with events with word spaces. Bootstrap 3 has reduced support for Internet Explorer 7 and Firefox 3.6, but there is a polyfill supplement for these browsers.

Bootstrap 4

Mark Otto announced Bootstrap 4 on October 29, 2014. The first alpha version of Bootstrap4 was released on August 19, 2015. The first beta version was released on August 10, 2017. Mark stopped work on Bootstrap 3 on September 6, 2016, to release working time on Bootstrap 4. Bootstrap 4 was completed on January 18, 2018.

Bootstrap 5

Our latest release, Bootstrap 5, focuses on upgrading the v4 codebase with as few major crash changes as possible. We have improved existing features and components, removed the support of older browsers, reduced jQuery to standard JavaScript, and adopted future-use technologies such as custom CSS architectures as part of our tools.

Major changes include

- A new part of the offcanvas menu
- Removes jQuery dependence in favor of vanilla JavaScript
- Rewrite the grid to support responsive gaps and columns placed outside the lines
- Moves documents from Jekyll to Hugo
- Discard Internet Explorer support [
- Moving testing infrastructure from QUnit to Jasmine
- Add a custom set of SVG icons
- Add custom CSS layouts
- Improved API
- Improved grid system
- Customizing advanced documents

- Forms updated
- RTL support

FEATURES

Bootstrap is an HTML, CSS & JS Library focused on simplifying the development of knowledgeable web pages (unlike web applications). The main purpose of adding to a web project is to use Bootstrap options for color, size, font and layout for that project. Thus, the key is whether the leading engineers get those options the way they want. Once added to a project, Bootstrap provides a simple way to style definitions for all HTML features. The result is the same look of prose, tables and form elements in all web browsers. Additionally, developers can use the CSS classes defined in Bootstrap to further customize the look and feel of their content. For example, Bootstrap provided brightly colored and black tables, page titles, high-quality drag quotes, and highlighted text.

Bootstrap also comes with a few JavaScript features in the form of jQuery plugins. They offer additional user interaction features such as chat boxes, tips, and carousels. Each component of Bootstrap contains HTML format, CSS declarations, and in some cases JavaScript-compliant code. It also enhances the functionality of other existing interaction features, including for example the auto-complete function of input fields.

The most prominent features of Bootstrap are its structural components, as they affect the entire web page. The part of the basic structure is called the 'Container', as everything else on the page is placed on it. Engineers can choose between a wide-open container and a wide-open container. While the latter always complements the width of the webpage, the first one uses the same width of the five previously defined, depending on the screen size showing the page:

- Smaller than 576 pixels
- 576–768 pixels
- 768–992 pixels
- 992–1200 pixels
- Larger than 1200 pixels

CREATE A HOME PAGE WITH BOOTSTRAP

Add HTML5 Document Type

Bootstrap uses HTML elements and CSS formats that require HTML5 document type. Always include an HTML5 doctype at the beginning of the page, as well as a language identifier and a set of appropriate characters:

```
<! DOCTYPE html>
<html lang = "Zu">
  <head>
    <meta charset = "utf-8">
  </head>
</html>
```

Bootstrap 3 Is Advanced

Bootstrap 3 is designed to respond to mobile devices. Mobile-first styles are part of the basic framework.

To ensure proper rendering and zoom touch, add the following <meta> tag within the <head> feature:

```
<meta content = "width = device-width, initial-scale
= 1" >
  name = "viewport">
```

Width=device-width determines the width of the page to follow the screen width of the device (which will vary depending on the device). Initial scale=1 sets the first zoom level when the page is loaded in the browser.

BASIC BOOTSTRAP PAGES

There are two classes we can use to get two bootstrap pages that use class container and container liquid.

Example:

```
<! DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf8">
```

```

<meta content = "width = device-width, initial-
scale = 1" name = "viewport">
<link href = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/css/bootstrap.min.css" rel =
"stylesheet" >
<script src = "http://ajax.googleapis.com/ajax/libs/
jquery/3.6.0/jquery.min.js"> </script>
<script src = "http://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>
</head>
<body>

<div class = "container">
  <h1> Container page </h1>
  <p> You can add any data. </p>
</div>

</body>
</html>

```

BOOTSTRAP CONTAINERS

Bootstrap also requires something that encapsulates site content. There are two content classes to choose from

- The container section provides a comprehensive wide-responsive container.

```

<! DOCTYPE html>
<html lang = "en">
<head>
  <title> example -- Bootstrap </title>
  <meta charset = "utf-8">
  <meta initial-scale = 1" , content = "width =
device-width name = "viewport">
<link href = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/css/bootstrap.min.css" rel =
"stylesheet" >
  <script src = "http://ajax.googleapis.com/ajax/libs/
jquery/3.6.0/jquery.min.js"> </script>

```

```

<script src = "http://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

<style>
  body{
    background-color: RGB(187, 186, 186);
  }
  .container{
    background-color: WhiteSmoke;
    height: 400px;
  }
</style>
</head>
<body>

<div class = "container">
  <h1> Container page </h1>
  <p>Lorem ipsum, or lipsum as it is sometimes known,
is rough text used in laying out print, graphic or web
designs.

  The passage is attributed to an unknown typesetter
in the 15th century who is thought to have scrambled
parts of Cicero's De Finibus
    Bonorum et Malorum for the use in a type specimen
book. It usually begins with:
    </p>
</div>
<p>Rest content is out of container.</p>

</body>
</html>

```

Container page

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. The passage is attributed to an unknown typesetter in the 15th century who is thought to have scrambled parts of Cicero's De Finibus Bonorum et Malorum for use in a type specimen book. It usually begins with:

- The container-fluid section provides a comprehensive full container, covering the entire width of the viewing hole.

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta initial-scale = 1" , name = "viewport" content
= "width = device-width>
  <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
  <script src = "http://ajax.googleapis.com/ajax/libs/
jquery/3.6.0/jquery.min.js"> </script>
  <script src = "http://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

  <style>
    body{
      background-color: RGB(187, 186, 186);
    }
    .container{
      background-color: WhiteSmoke;
      height: 400px;
    }
  </style>
</head>
<body>

<div class = "container-fluid">
  <h1> Container page </h1>
  <p>Lorem ipsum, or lipsum as it is sometimes known,
is demo text used in laying out print, graphic or web
designs.

  The passage is attribute is to an unknown
typesetter in the 15th century who is thought to have
scrambled parts of Cicero's De Finibus

  Bonorum et Malorum is for use in a type specimen
book. It usually begins with:
  </p>
```

```

</div>
</body>
</html>

```

Container page

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. The passage is attributed to an unknown typesetter in the 15th century who is thought to have scrambled parts of Cicero's De Finibus Bonorum et Malorum for use in a type specimen book. It usually begins with:

Bootstrap – Container-Fluid

BOOTSTRAP GRID SYSTEM

The Bootstrap grid system is a simple and powerful way to build all structures and sizes. You can then customize it based on its system of 12 columns (12 columns are found in each row). It also has six responsive breakpoints available as default. You can even create your own breakpoint according to your needs. The Bootstrap grid system is totally based on the CSS Flexbox model. The grid system is built using CSS flexbox. The Bootstrap grid program is an easy way to create custom and responsive structures for all types of content. Like all other Bootstrap features, the grid system is advanced and fully responsive.

The Bootstrap grid system uses rows, columns, and containers to build the structure of any web page and align content accordingly. Creating a design using the grid system is one of the fastest and easiest ways to create a responsive web structure. You can use the predefined grid classes provided by Bootstrap to create faster designs.

Grid System is built using Flexbox, and will allow you to create a limit of 12 columns throughout the page or the entire screen width. You can use each of the 12 columns separately or combine columns to create a wider column. You can use six columns each and create two wide columns. Similarly, you can build three wide columns by combining four columns together. You can combine two columns to create six wide columns.

GRID CLASSES

The grid system in Bootstrap 5 has six available classes that we can use to customize our grids as per our convenience. In order to create more dynamic and flexible layouts we can combine these classes together to achieve our desired layout.

<code>.col-</code>	Extra small devices – screen width < 576px
<code>.col-sm-</code>	Small devices – screen width ≥ 576px
<code>.col-md-</code>	Medium devices – screen width ≥ 768px
<code>.col-lg-</code>	Large devices – screen width ≥ 992px
<code>.col-xl-</code>	Extra-large devices – screen width ≥ 1200px
<code>.col-xxl-</code>	Extra extra-large devices – screen width ≥ 1400px

BASIC STRUCTURE OF A BOOTSTRAP GRID

The following is a simple structure of a Bootstrap grid:

```
<div class="row">
  <div class="col-*-*"></div>
  <div class="col-*-*"></div>
</div>
```

Example:

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta initial-scale = 1" name = "viewport" content
= "width = device-width,>
  <link href = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/css/bootstrap.min.css" rel =
"stylesheet">
  <script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

  <style>
    body{
      background-color: RGB(187, 186, 186);
    }
    .container{
      background-color: WhiteSmoke;
      height: 400px;
    }
```

```

.col-md-4{
    border:1px dodgerblue solid;
    height: 100px;
}
</style>
</head>
<body>

<div class = "container">
<h1> Bootstrap Grid</h1>
    <div class="row" >
        <div class="col-md-4">Col 1</div>
        <div class="col-md-4">Col 2</div>
        <div class="col-md-4">Col 3</div>
    </div>

</body>
</html>

```

Explanation

The first line of the code row is a wrapper class which means that the class row would wrap around all the columns. It is necessary to add a wrapper class to initiate the grid system in Bootstrap. Add as many classes after adding a wrapper class, keeping in mind that the sum of columns should not exceed 12.

In the above example, Bootstrap will create the layout by creating three columns of equal width. Each column will contain four grid columns. The columns made will be responsive and align according to the screen size.

Code Example

```

<! DOCTYPE html>
<html lang = "en">
<head>
    <title> Bootstrap example </title>
    <meta charset = "utf-8">
    <meta name = "viewport" content = "width = device-
width, initial-scale = 1">
    <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">

```

```

<script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
<script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

<style>
  body{
    background-color: RGB(187, 186, 186);
  }
  .container{
    background-color: WhiteSmoke;
    height: 400px;
  }
  .col-md-6{
    border:1px dodgerblue solid;
    height: 100px;
  }
</style>
</head>
<body>

<div class = "container">
<h1> Bootstrap Grid</h1>
  <div class="row" >
    <div class="col-md-6">Col 1</div>
    <div class="col-md-6">Col 2</div>
  </div>

</body>
</html>

```

Bootstrap Grid

Col 1	Col 2

Bootstrap – Grid (Medium)

The first value after class col shows the responsiveness. Use default grid classes available in Bootstrap. The second value represents a number that must add up to 12 for each row.

The above example will output two columns, and each column will contain six grid column widths. In smaller screens, the columns will automatically get stacked one on the other.

Code Example (Extra Small Grid)

```
<div class="col-6">Column 1</div>
<div class="col-3">Column 2</div>
```

The above example will create a grid where one column will span 6 column grids, and the other will span 3. It is not always necessary that the sum of columns is 12. It can also be less than 12. In this example, only nine-column grids are being used. The remaining three-column will remain unused and left empty.

Code Example (Small Grid)

```
<! DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width = device-
width, initial-scale = 1">
  <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
  <script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

  <style>
    body{
      background-color: RGB(187, 186, 186);
    }
    .container{
      background-color: WhiteSmoke;
```

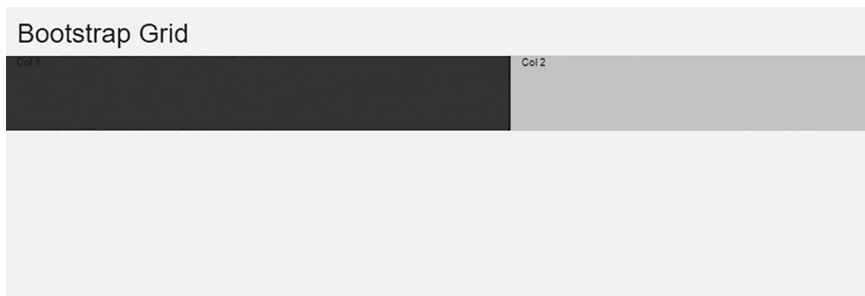
```

        height: 400px;
    }
    .col-sm-7{
        background-color: blue;
        height: 100px;
    }
    .col-sm-5{
        background-color: yellow;
        height: 100px;
    }
}
</style>
</head>
<body>

<div class = "container">
<h1> Bootstrap Grid</h1>
    <div class="row" >
        <div class="col-sm-7">Col 1</div>
        <div class="col-sm-5">Col 2</div>
    </div>

</body>
</html>

```



Bootstrap – Grid (Small)

Small devices have a screen width between 576 and 767 pixels. In the above example, four columns of equal width will be created because we have not specified the number of grids we would like to use for each. In case the number of col elements is not determined, Bootstrap simply divides them equally, spanning the entire width of the screen. As this example is

for small devices, the columns will be horizontal up until 576 pixels, and then they will stack on top of each other in extra small devices.

Code Example (Medium Grid)

```
<div class="col-md-6">Column 1</div>
<div class="col-md-6">Column 2</div>
```

Small devices have a screen width between 768 and 991 pixels. For medium devices, we will use the ‘col-md’ class. In the above example, two columns of equal width will be created, and for screen sizes <768 pixels, the columns will get stacked vertically on top of each other. So, for small devices and extra small devices, the columns will be vertically stacked.

Code Example (Small and Medium Grid Combined)

```
<div class="col-sm-3 col-md-6">Column 1</div>
<div class="col-sm-9 col-md-6">Column 2</div>
```

You can combine multiple grid classes to specify how the columns should be stacked in screens of different sizes. When the website is accessed from an extra small device, the stack would be vertical, but as soon as it switches to a small device, it will shift to two columns of nine and three-column width. And when the screen is scaled even higher to a medium device, it will split in half as we have divided it in an equal ratio.

Code Example (Large Grid)

```
<! DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width = device-
width, initial-scale = 1">
  <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
  <script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>
```

```

<style>
  body{
    background-color: RGB(187, 186, 186);
  }
  .container{
    background-color: WhiteSmoke;
    height: 400px;
  }
  .col-lg-7{
    background-color: blue;
    height: 100px;
  }
  .col-lg-5{
    background-color: yellow;
    height: 100px;
  }
}

</style>
</head>
<body>

<div class = "container">
<h1> Bootstrap Grid</h1>
<!-- Large -->
  <div class="row" >
    <div class="col-lg-7">Col 1</div>
    <div class="col-lg-5">Col 2</div>
  </div>

</body>
</html>

```

Bootstrap Grid



Bootstrap – Grid (Large)

Large devices are those devices that have screen widths from 992 to 1199 pixels. For large devices, we will use the 'col-LG-' class. Devices with a width greater than 1199 pixels will have four columns of equal width, and the rest of the devices smaller than that will have 100% width.

Code Example (Extra-Large Grid)

```
<!DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width = device-
width, initial-scale = 1">
  <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
  <script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

  <style>
    body{
      background-color: RGB(187, 186, 186);
    }
    .container{
      background-color: WhiteSmoke;
      height: 400px;
    }
    .col-xl-7{
      background-color: blue;
      height: 100px;
    }
    .col-xl-5{
      background-color: yellow;
      height: 100px;
    }
  }

</style>
</head>
<body>
```

```

<div class = "container">
<h1> Bootstrap Grid</h1>
<!-- Large -->
  <div class="row" >
    <div class="col-xl-7">Col 1</div>
    <div class="col-xl-5">Col 2</div>
  </div>

</body>
</html>

```



Bootstrap – Grid (x-Large)

Extra-large devices have a screen width of 1200 pixels and above. In the above example, we will see two columns in large devices and above. One will have a grid column width of eight, and the other will have a grid column width of four columns.

Note: The sum should always be 12 or <12.

Code Example (Extra Extra-Large Grid)

```

<div class="col-XXL-6">Column 1</div>
<div class="col-XXL-3">Column 2</div>

```

Extra extra-large devices have a screen width of 1400 pixels and above. The above example will produce a column of six and a column of three, and the rest of the space comprising of three columns would be left empty.

HOW THE GRID WORKS

- Our grid supports six responsive breakpoints. Breakpoints are based on small-scale media queries, which means they affect that area and all over it (e.g., `.col-sm-4` works in sm, MD, LG, XL, and

XXL). This means you can control the container and column size and behavior for each partition.

- Containers are in the middle and browse your content horizontally. Use `.container` with response pixel width, `.container-fluid` width: 100% for all viewing ports and devices, or responsive container (e.g., `.container-MD`) with a combination of liquid width and pixel width.
- The rows are the folds of the columns. Each column has horizontal padding (called gutter) to control the space between them. This finish is then counteracted in the lines with the negative margins to ensure that the content in your columns is visually aligned at the bottom left. Lines also support editing classes to parallel use column balancing classes and trash classes to transform your content space.
- The columns are surprisingly flexible. There are 12 template columns available in each row, which allows you to create different combinations of elements that include any number of columns. Most of the column classes indicate the number of template columns that must be passed (e.g., `col-4` extends four). The width is set as a percentage to maintain the same relative size.
- Gutters are also responsive and customized. Gutter classes are available at all breakpoints, with all the same sizes as our margin and padding space. Replace horizontal gutters with `.gx- *` sections, straight gutters with `.gy- *`, or all gutters with `.g- *` sections. `.G-0` is also available for gutters removal.
- Sass variables, maps, and combinations enable the grid. If you do not want to use the predefined grid classes in Bootstrap, you can use our Sass grid source to create your own with a sentence made with more sentences. We also added some custom CSS layouts to use these Sass variables for greater flexibility.

Three Equal Columns

For small three equal column, we 'use `col-sm-4`'. If you want it, you need to add the first row and then need `.col-sm-4`.

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-4">.col-sm-4</div>
```

```
<div class="col-sm-4">.col-sm-4</div>
</div>
```

Two Unequal Columns

For unequal column you can use col-sm-4 with col-MD-8 also you can write col-sm-2 with col-sm-10.

```
<div class="row">
  <div class="col-sm-4">.col-sm-4</div>
  <div class="col-sm-8">.col-sm-8</div>
</div>
```

Here you can have a complete example of all the col-* used in Bootstrap Grid System.

```
<! DOCTYPE html>
<html lang = "en">
<head>
  <title> Bootstrap example </title>
  <meta charset = "utf-8">
  <meta name = "viewport" content = "width = device-
width, initial-scale = 1">
  <link rel = "stylesheet" href = "https://maxcdn.
bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.
css">
  <script src = "https://ajax.googleapis.com/ajax/
libs/jquery/3.6.0/jquery.min.js"> </script>
  <script src = "https://maxcdn.bootstrapcdn.com/
bootstrap/3.4.1/js/bootstrap.min.js"> </script>

  <style>
    body{
      background-color: RGB(187, 186, 186);
    }
    .container{
      background-color: WhiteSmoke;
      height: 400px;
    }
    .col-xxl-7{
      background-color: yellow-green;
      height: 100px;
    }
  </style>
```

```
.col-xxl-5{
  background-color:palevioletred;
  height: 100px;
}
.col-xl-7{
  background-color: brown;
  height: 100px;
}
.col-xl-5{
  background-color:wheat;
  height: 100px;
}
.col-lg-7{
  background-color: red;
  height: 100px;
}
.col-lg-5{
  background-color: green;
  height: 100px;
}
.col-md-7{
  background-color: orange;
  height: 100px;
}
.col-md-5{
  background-color: purple;
  height: 100px;
}
.col-sm-7{
  background-color: indianred;
  height: 100px;
}
.col-sm-5{
  background-color: pink;
  height: 100px;
}
.col-7{
```

```

        background-color: seagreen;
        height: 100px;
    }
    .col-5{
        background-color: orchid;
        height: 100px;
    }
}

</style>
</head>
<body>

<div class = "container">
<h1> Bootstrap Grid</h1>
<div class="row" >
    <div class="col-xxl-7"><h1>XXL </h1></div>
    <div class="col-xxl-5"><h1>XXL </h1></div>
</div>
<!-- Large -->
    <div class="row" >
        <div class="col-xl-7"><h1>XL </h1></div>
        <div class="col-xl-5"><h1>XL </h1></div>
</div>
<div class="row" >
    <div class="col-lg-7"><h1>LG </h1></div>
    <div class="col-lg-5"><h1>LG </h1></div>
</div>
<div class="row" >
    <div class="col-md-7"><h1>MD </h1></div>
    <div class="col-md-5"><h1>MD </h1></div>
</div>
<div class="row" >
    <div class="col-sm-7"><h1>SM </h1></div>
    <div class="col-sm-5"><h1>SM </h1></div>
</div>
<div class="row" >
    <div class="col-7"><h1> </h1></div>
    <div class="col-5"><h1></h1></div>
</div>

</body>
</html>

```



Bootstrap Complete Grid System

FLEXBOX

A flex in Bootstrap allows us to manage the layout and helps us in aligning the components accordingly. Using Flexbox, we can manage the sizing of the columns in any grid and easily navigate the page. To fully grasp all the display utilities provided through flex in Bootstrap, it is essential to know about CSS Flexbox.

The flexible layout model, also called the Flexbox in CSS, is a way to provide efficient mapping out and distribution of space among all the items in a container. It helps us align and layout the items to get positioned dynamically concerning the available space. If a container becomes a flex, it can easily alter its height or width to fill the open space appropriately. This is done to accommodate all devices with different display and screen widths. The main idea is that any flex container should expand or shrink

and fill the available space to prevent overflow. In CSS, the flexbox layout is generally considered more appropriate for small-scale designs, while grid layouts serve better on a large scale.

Each flex container is seen as a parent element, and each item contained within the parent element or container is referred to as the child/children element.

Flexbox Properties in CSS

Display: this property is used to define a flex container. The flex could be inline or block, depending on its value. This property makes the display container the parent. Therefore, all the elements contained within would become the direct children of this parent element.

Syntax:

- **display: block;**
Block: This value will create a block-level element box which will make line breaks both before and after itself whenever it is placed in the normal flow of the page.
- **display: inline;**
Inline: This value will create one or more inline element boxes that will stick to each other and be in the same line if space is available. It will not generate line breaks before or after itself.
- **display: inline-block;**
Inline-block: This display property value would create a block-level element box that will flow with the surrounding content as if it is a single inline box.
- **display: flex;**
Flex: This value will make the parent container behave like a block element and layout the content inside itself according to the flexbox model. Line break would be generated before and after the container in normal flow.
- **display: inline-flex;**
Inline-flex: This property value makes the element behave like an inline element. However, the content within the component gets placed according to the flexbox model.

- `display: grid;`

It is a 2D system that makes it easier to align and arrange elements on a web page. You can work with rows and columns together, determining how many rows and columns we want our parent element to have.

- `display: inline-grid;`
- `Display: none`

If we set to 0% opacity, it still takes space on the page. But if we set that the thing to `display: none`, it's completely removed from the document flow.

- `display: flow-root;`

Note: For those browsers that support the two-value syntax, the inner value of `display: flex` would be enough to set the outer value to block on its own as that is the expected and the default value. For specific changes, you need to define it as `inline-flex` explicitly.

Flexbox Properties in Bootstrap

The bootstrap properties are similar to CSS but the difference is that we can use Bootstrap as the inline CSS. Responsive variations also exist for `.d-flex` and `.d-inline-flex`.

- `.d-flex`
- `.d-inline-flex`
- `.d-sm-flex`
- `.d-sm-inline-flex`
- `.d-md-flex`
- `.d-MD-inline-flex`
- `.d-LG-flex`
- `.d-LG-inline-flex`
- `.d-xl-flex`
- `.d-xl-inline-flex`

Flex-Direction in CSS

The `flex-direction` property defines the direction in which the flex items would get placed within a flex container. Flexbox essentially works by laying out all the elements in a single order. So, the elements would either get placed horizontally in rows or vertically in columns.

Syntax:

- `flex-direction: row;`

Row: This is the default value of the `flex-direction` property. It goes from left to right. The main axis of the parent element is said to be the same as the direction of the text. The order of the content would be from the main start point of the container to the main endpoint of the container.

- `flex-direction: row-reverse;`

Row-reverse: The core behavior is the same as that of the row, the only difference being the direction in which the content goes is from the container's primary endpoint to the parent container's main start point.

- `flex-direction: column;`

Row: This is the default value of the `flex-direction` property. It goes from right to left. The main axis of the parent element is said to be the same as the direction of the text. The order of the content would be from the main start point of the container to the main endpoint of the container.

- `flex-direction: column-reverse;`

Column-reverse: The core behavior is the same as that of the row, the only difference being the direction in which the content goes is from the container's primary endpoint to the parent container's main start point.

Flex Direction in Bootstrap

Responsive variations also exist for `flex-direction`.

- `.flex-row`
- `.flex-row-reverse`

- `.flex-column`
- `.flex-column-reverse`
- `.flex-sm-row`
- `.flex-sm-row-reverse`
- `.flex-sm-column`
- `.flex-sm-column-reverse`
- `.flex-md-row`
- `.flex-MD-row-reverse`
- `.flex-md-column`
- `.flex-MD-column-reverse`
- `.flex-lg-row`
- `.flex-LG-row-reverse`
- `.flex-lg-column`
- `.flex-LG-column-reverse`
- `.flex-xl-row`
- `.flex-xl-row-reverse`
- `.flex-xl-column`
- `.flex-xl-column-reverse`

Set the variable direction in the flexible container with the control resources. In most cases, you can leave a horizontal section here as the default browser is in line. However, you may encounter situations where you need to explicitly set this value (such as responsive properties).

Use the `.flex-row` to set the horizontal (browser default), or `.flex-row-reverse` to start the horizontal direction from the other side.

Flexbox Features

The following are some of the features of the Flexbox architecture:

- **Direction:** You can organize items on a web page in any way like left to right, right to left, top to bottom, and bottom to top.

- **Wrap:** Using Flexbox, you can rearrange the layout of web page content.
- **Fold:** In the case of static web page content (in one row), you can fold them in multiple rows (both horizontal) and vertically.
- **Align:** Using Flexbox, you can Align web page content with respect to your container.
- **Resize:** By using Flexbox, you can increase or decrease the size of the objects on the page to fit in the available space.

Flex-Wrap in CSS

The default behavior of any child element of a flex container is to fit into one single line. To wrap the child items according to your needs, you can use the flex-wrap property. This property would set the direction in which elements are stacked if the wrapping is permitted. The items have to be flexible to work for this property. If the items are not flexible, this property has no effect whatsoever.

Syntax:

- **flex-wrap: nowrap;**
Nowrap: This is the default value of the flex-wrap property. Setting the flex-wrap value to no-wrap will ensure that all the flex items are in one line. Laying out all the items in one line might cause the parent container to overflow.
- **flex-wrap: wrap;**
Wrap: If you put the value of the flex-wrap property as a wrap, then all the child items within the parent container will wrap onto multiple lines. This wrapping could go on from top to bottom
- **flex-wrap: wrap-reverse;**
Wrap-reverse: This works in a similar way to the wrap property. The only difference is that the direction the child items will wrap would get reversed. All the flex items will wrap onto multiple lines in that reverse order.

Flex-Wrap in Bootstrap

You can change flex items wrapped in a flex container also choose from no wrapping at all with `.flex-nowrap`, wrapping with `.flex-wrap`, or reverse wrapping with `.flex-wrap-reverse`.

Responsive variations also exist for flex-wrap:

- `.flex-nowrap`
- `.flex-wrap`
- `.flex-wrap-reverse`
- `.flex-sm-nowrap`
- `.flex-sm-wrap`
- `.flex-sm-wrap-reverse`
- `.flex-md-nowrap`
- `.flex-md-wrap`
- `.flex-md-wrap-reverse`
- `.flex-lg-nowrap`
- `.flex-lg-wrap`
- `.flex-lg-wrap-reverse`
- `.flex-xl-nowrap`
- `.flex-xl-wrap`
- `.flex-xl-wrap-reverse`

Flex-Flow

It is a shorthand property, and it combines two properties, namely, flex-direction and flex-wrap, which we have discussed above. The flex-flow in its entirety and the properties contained within it depend on whether the parent element is a flexible item or not. If that is not the case, then the flex-flow property in itself has no effect. Both the properties flex-direction and flex-wrap, when combined, form the two main axes of the flex container.

Flex-direction property defines the main axis, while the flex-wrap property defines the cross axis. The default value of the flex-flow property is row no-wrap.

Syntax:

- `/*flex-flow: <'flex-direction'> and <'flex-wrap'> */`
- `flex-flow: row nowrap/wrap/wrap-reverse;`
- `flex-flow: row-reverse nowrap/wrap/wrap-reverse;`
- `flex-flow: column nowrap/wrap/wrap-reverse;`
- `flex-flow: column-reverse nowrap/wrap/wrap-reverse;`

Justify-Content in CSS

This property helps us align the items of the flexible container when the items do not use all the space available within the main axis. The main axis is usually horizontal, but it can also be changed using the flex-direction property. The need to define the alignment across the main axis arises because sometimes, all the items available on a line are not flexible or have reached their maximum size. This property also helps control if there is some kind of overflow of the items. Justify content property helps us align all the flex items at the center of the container. The default value of the property is flex-start.

Syntax:

- `justify-content: center;`
Center: The items are positioned and packed toward each other of the main container or the parent element. The items are centered along the main axis.
- `justify-content: start;`
Start: The items are positioned toward the starting edge of the parent container. The positioning will be along the main axis. The items will stick to each other toward the start with no spacing.
- `justify-content: end;`
End: The items are positioned toward the ending edge of the parent container. The positioning will be along the main axis. The items will stick to each other toward the end with no spacing.

- `justify-content: flex-start;`

Flex-start: It is the default value of the `justify-content` property. Using this property would make sure that the items are positioned toward the start of the container. Using `flex-start` as the value of the property `justify-content` would make sure that all the items are at the edge of the container packed to each other. The start of the container depends upon the main start side of the parent container. There would be no spacing or gap available between all the flex items. However, this property only applies to flex layout items. Any items that are not direct children of any parent container will be treated equivalent to start.

- `justify-content: flex-end;`

Flex-end: All the flex items are positioned toward the edge of the parent container at the end side. The main end side depends upon the alignment of the flex container. There would be no spacing between the flex items. However, this property value only works for flex layout items. For any items that are not direct children of a parent container, this value will be treated equivalent to end.

- `justify-content: left;`

Left: All the items are positioned toward the left edge of the parent container. If the property axis is not parallel with the inline axis of the container, then this value will behave like the start.

- `justify-content: right;`

Right: All the items are positioned toward the right edge of the parent container in the appropriate axis. If the property axis is not parallel with the inline axis of the container, then this value will behave like the start.

For positional alignment:

- `justify-content: normal;`

All the items will be positioned in their default place. It works as if no value of `justify-content` is being set. The value `normal` behaves like the stretch in flex and grid containers.

- `justify-content: space-between;`

All the child items are evenly spaced out and distributed throughout the parent container. The alignment of items happens along the main axis. The spacing between each item in the container is

the same. The gap between all the adjacent pairs of items would also be equal. The very first item of the container would be packed toward the main start edge, whereas the last item of the container would be stuck toward the main end edge of the parent container.

- justify-content: space-around;

All the child items are evenly distributed throughout the container and are separated from each other with an equal amount of space. The items are distributed along the main axis of the parent container. The space between each adjacent pair of child items is the same. There is a space at the start of the container and toward the end of the container. Even though there is spacing among all the container elements, visually, the spacing is not equal. The space before the first and the last item is equal to half of the available space between each pair of adjacent items. If an item has one unit of length against the edge of the container, then it will have two units of space with the next item.

- justify-content: space-evenly;

As the name suggests, the items are evenly distributed throughout the parent container. The alignment occurs along the main axis. The spacing between each available item, including the first item at the main start edge and the last item at the main end edge, is the same.

- justify-content: stretch;

The main axis, if the size of all the available items of a container after combining turns out to be less than the parent container's size, then the auto-sized items would increase in size. This change in size occurs until the child items' combined size fills in the parent container along the main axis.

Note: some of these property values don't have support for specific browser and browser versions. Like space-between doesn't have permission from some versions of edge. The values that work in most browsers and are the safest are flex-start, flex-end, and center.

Order

You can change the visual order of specific flex items with a handful of order utilities. You can provide options for making an item first or last, as well as a reset to use the DOM order. As order takes any integer value (e.g., 5), add custom CSS for any additional values needed.

Responsive variations also exist for order:

- .order-0
- .order-1
- .order-2
- .order-3
- .order-4
- .order-5
- .order-6
- .order-7
- .order-8
- .order-9
- .order-10
- .order-11
- .order-12
- .order-sm-0
- .order-sm-1
- .order-sm-2
- .order-sm-3
- .order-sm-4
- .order-sm-5
- .order-sm-6
- .order-sm-7
- .order-sm-8
- .order-sm-9
- .order-sm-10
- .order-sm-11
- .order-sm-12

- .order-md-0
- .order-md-1
- .order-md-2
- .order-md-3
- .order-md-4
- .order-md-5
- .order-md-6
- .order-md-7
- .order-md-8
- .order-md-9
- .order-md-10
- .order-md-11
- .order-md-12
- .order-LG-0
- .order-LG-1
- .order-LG-2
- .order-LG-3
- .order-LG-4
- .order-LG-5
- .order-LG-6
- .order-LG-7
- .order-LG-8
- .order-LG-9
- .order-LG-10
- .order-LG-11
- .order-LG-12

- `.order-xl-0`
- `.order-xl-1`
- `.order-xl-2`
- `.order-xl-3`
- `.order-xl-4`
- `.order-xl-5`
- `.order-xl-6`
- `.order-xl-7`
- `.order-xl-8`
- `.order-xl-9`
- `.order-xl-10`
- `.order-xl-11`
- `.order-xl-12`

Justify Content in Bootstrap

Responsive variations also exist for justify-content:

- `.justify-content-start`
- `.justify-content-end`
- `.justify-content-center`
- `.justify-content-between`
- `.justify-content-around`
- `.justify-content-sm-start`
- `.justify-content-sm-end`
- `.justify-content-sm-center`
- `.justify-content-sm-between`
- `.justify-content-sm-around`
- `.justify-content-md-start`

- `.justify-content-md-end`
- `.justify-content-md-center`
- `.justify-content-md-between`
- `.justify-content-md-around`
- `.justify-content-lg-start`
- `.justify-content-lg-end`
- `.justify-content-lg-center`
- `.justify-content-lg-between`
- `.justify-content-lg-around`
- `.justify-content-xl-start`
- `.justify-content-xl-end`
- `.justify-content-xl-center`
- `.justify-content-xl-between`
- `.justify-content-xl-around`

Align Self

You can use `align-self` flexbox items to change their alignment on the cross axis. Choose from the same options as `align-items` such as `start`, `end`, `center`, `baseline`, or `stretch`. Responsive variations also exist for `align-self`.

- `.align-self-start`
- `.align-self-end`
- `.align-self-center`
- `.align-self-baseline`
- `.align-self-stretch`
- `.align-self-sm-start`
- `.align-self-sm-end`
- `.align-self-sm-center`
- `.align-self-sm-baseline`

- `.align-self-sm-stretch`
- `.align-self-md-start`
- `.align-self-md-end`
- `.align-self-md-center`
- `.align-self-md-baseline`
- `.align-self-md-stretch`
- `.align-self-lg-start`
- `.align-self-lg-end`
- `.align-self-lg-center`
- `.align-self-LG-baseline`
- `.align-self-lg-stretch`
- `.align-self-xl-start`
- `.align-self-xl-end`
- `.align-self-xl-center`
- `.align-self-xl-baseline`
- `.align-self-xl-stretch`

Align Items

You can use `align-items` on flexbox containers to change the alignment of flex items on the cross various axis (the y-axis to start, x-axis if flex-direction: column). Choose from start, end, center, baseline, or stretch (browser default). Responsive variations also exist for `align-items`.

- `.align-items-start`
- `.align-items-end`
- `.align-items-center`
- `.align-items-baseline`
- `.align-items-stretch`
- `.align-items-sm-start`

- `.align-items-sm-end`
- `.align-items-sm-center`
- `.align-items-sm-baseline`
- `.align-items-sm-stretch`
- `.align-items-md-start`
- `.align-items-md-end`
- `.align-items-md-center`
- `.align-items-md-baseline`
- `.align-items-md-stretch`
- `.align-items-lg-start`
- `.align-items-lg-end`
- `.align-items-lg-center`
- `.align-items-LG-baseline`
- `.align-items-lg-stretch`
- `.align-items-xl-start`
- `.align-items-xl-end`
- `.align-items-xl-center`
- `.align-items-xl-baseline`
- `.align-items-xl-stretch`

Grow and Shrink

You can use `.flex-grow-*` utilities to toggle a flex item's ability to grow to fill rest space. The `.flex-grow-1` elements use all available space it can, while allowing the remaining two flex items their necessary space. Responsive variations also exist for `flex-grow` and `flex-shrink`.

- `.flex-{grow|shrink}-0`
- `.flex-{grow|shrink}-1`
- `.flex-sm-{grow|shrink}-0`

- `.flex-sm-{grow|shrink}-1`
- `.flex-MD-{grow|shrink}-0`
- `.flex-MD-{grow|shrink}-1`
- `.flex-LG-{grow|shrink}-0`
- `.flex-LG-{grow|shrink}-1`
- `.flex-xl-{grow|shrink}-0`
- `.flex-xl-{grow|shrink}-1`

Fill

Use the `.flex-fill` class on a collection of sibling elements to force them into widths equal to their content while taking up all available horizontal space. Responsive variations also exist for `flex-fill`.

- `.flex-file`
- `.flex-sm-fill`
- `.flex-md-fill`

e

- `.flex-LG-fill`
- `.flex-xl-fill`

RESPONSIVE BREAKPOINTS

As Bootstrap has been developed to be mobile first, we use a few media queries to create realistic breakouts for our architecture and visual connectors. These high-density points are based on the small width of the viewing hole and allow us to maximize features as the viewing hole changes.

Bootstrap primarily uses the scope of the following media queries – or breakpoint – in our Sass files that are the source of our architecture, grid system, and components.

```
// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }
```

```
// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

z-Index

A few Bootstrap sections use z-index, a CSS feature that helps control the structure by providing a third-party content editor. We use the default z-index scale in Bootstrap designed to properly navigate, navigation tools and popovers, models, and more.

These high numbers start at the wrong number, high and clear enough to avoid collision. We need a common set of this for all our horizontal parts – tools, popovers, navbes, downgrades, models – so that we can be reasonably consistent in behavior. There’s no reason we couldn’t use 100+ or 500+.

We do not recommend customization of these individual numbers; if you change one, you probably need to replace them all.

- \$zindex-dropdown
- \$zindex-sticky
- \$zindex-fixed
- \$zindex-modal-backdrop
- \$zindex-modal
- \$zindex-popover
- \$zindex-tooltip

To manage cross-border parameters within components (e.g., buttons and input groups), we use the minimum single-digit values of z, index 1, 2, and 3 for default, up, and active conditions. For high/fixed/active navigation, we bring something forward with a high z value indicator to indicate its boundary over sibling elements.

COMPONENTS IN BOOTSTRAP

Bootstrap Alerts

These are used to provide a way to create predefined alert messages. It adds a style to your messages to make it more appealing to the users. There are four classes that are used within `<div>` element for alerts.

- `.alert-success`
- `.alert-info`
- `.alert-warning`
- `.alert-danger`

In Bootstrap4 adds new 4 alerts in Bootstrap Alert defined in Bootstrap 3 tutorial.

These are

- Primary: This alert box indicates an important action.
- Secondary: This alert box indicates a less important action.
- Dark: Dark gray alert box.
- Light: Light gray alert box.

It is created with the `.alert` class, followed by one of the contextual classes. List of all contextual classes:

- `.alert-success`
- `.alert-info`
- `.alert-warning`
- `.alert-danger`
- `.alert-primary`
- `.alert-secondary`
- `.alert-light`
- `.alert-dark`

Alert Links

You have to add the `alert-link` class to any links inside the alert box to create “matching colored links”: (`alert-link`)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.
bootstrapcdn.com/bootstrap/4.0.0-beta.2/css/bootstrap.
min.css">

</head>
<body>

<div class="container">
  <h2>Bootstrap Alert Links</h2>
  <div class="alert alert-success">
    <strong>Success!</strong>
  <div class="alert alert-info">
    <strong>Info!
  <div class="alert alert-warning">
    <strong>Warning!</strong>
  <div class="alert alert-danger">
    <strong>Danger!</strong>
  <div class="alert alert-primary">
    <strong>Primary!</strong>
  <div class="alert alert-secondary">
    <strong>Secondary!</strong>
  <div class="alert alert-dark">
    <strong>Dark!</strong>
  <div class="alert alert-light">
    <strong>Light!</strong>
  </div>
</div>

</body>
</html>
```

Output

Bootstrap Alert Links



Bootstrap Alert-Link

BOOTSTRAP BADGES

It is used to add extra information to any content, and you can use the `.badge` class together with a contextual class (like `.badge-secondary`) within `<span>` elements to create rectangular badges.

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="HTTPS://cdn.jsdelivrivr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</head>
```

```

<body>

<div class="container">
  <h2>Badges</h2>
  <p>Example heading <span class="badge badge-
secondary bg-primary">1</span></p>
</div>

</body>
</html>

```

BOOTSTRAP JUMBOTRON

A jumbotron indicates a gray box for getting extra attention to some special content or information.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>CSS Framework - Bootstrap</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="HTTPS://cdn.jsdelivriv.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</head>
<body>
  <div class="jumbotron">
    <p>Bootstrap is the most famous HTML, CSS, JS
framework for developing rhightly responsive, mobile-
first projects on the web.</p>
  </div>
</body>
</html>

```

Full-Width Jumbotron

If you wish to use a full-width jumbotron without rounded (circular) borders, that can add the `.jumbotron-fluid` class and a `.container` or `.container-fluid` inside of it:

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<title>CSS Framework - Bootstrap</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<link rel="stylesheet" href="HTTPS://cdn.jsdelivrnvr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</head>
<body>

<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <p>Bootstrap is the most famous HTML, CSS, and JS
framework for developing highly responsive, mobile-
first projects on the web.</p>
  </div>
</div>
</body>
</html>

```

BOOTSTRAP BUTTONS

It includes several predefined button styles, each button serving its own semantic purpose.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="https://cdn.jsdelivrnvr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
<body>

<div class="container">
  <h2>Button Styles</h2>
  <button class="btn">Basic</button>
  <button class="btn btn-primary">Primary</button>
  <button class="btn btn-secondary">Secondary</button>
  <button class="btn btn-success">Success</button>
  <button class="btn btn-info">Info</button>
  <button class="btn btn-warning">Warning</button>
  <button class="btn btn-danger">Danger</button>
  <button class="btn btn-dark">Dark</button>

```

```

    <button class="btn btn-light">Light</button>
    <button class="btn btn-link">Link</button>
</div>
</body>
</html>

```

Button Sizes

You can use the `.btn-LG` class for large buttons or `.btn-sm` class for small buttons:

```

<button type="button" class="btn btn-primary btn-LG">Large</button>
<button type="button" class="btn btn-primary">Default</button>
<button type="button" class="btn btn-primary btn-sm">Small</button>

```

Block Level Buttons

You can add class `“.btn-block”` to create a block level button that spans the entire width of the parent element.

```

<button type="button" class="btn btn-primary btn-block">Full-Width Button</button>

```

Active/Disabled Buttons

The class `.active` makes an appear pressed, and the `disabled` attribute makes a button unclickable.

```

<button class="btn btn-primary active">Active Primary</button>

```

Spinner Buttons

If you want to create a spinner/loader, use the `.spinner-border` class:

```

<button class="btn btn-primary">
  <span class="spinner-border spinner-border-sm">Wait
</span>
</button>

```

Colored Spinners

You can use any text color utilities to add a color to the spinner:

Example:

```
<div class="spinner-border text-muted"></div>
```

Button Tags

The `.btn` classes are developed to be used with the `<button>` element. However, you can also use any of these classes on `<a>` or `<input>` elements. When using button classes on `<a>` elements that are used to trigger in-page functionality (such as collapsing content), rather than linking to new pages or sections within the current page, these links should be given a `role="button"` to appropriately convey their purpose to assistive technologies such as screen readers.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="https://cdn.jsdelivrivr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</body>

<div class="container">
<a class="btn btn-primary" href="#"
role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</
button>
<input class="btn btn-primary" type="button"
value="Input">
<input class="btn btn-primary" type="submit"
value="Submit">
<input class="btn btn-primary" type="reset"
value="Reset">
</div>
</body>
```

Outline Buttons

To replace the default modifier classes with the `.btn-outline-*` ones to remove all background images and colors on any button.

Example:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <link rel="stylesheet" href="HTTPS://cdn.jsdelivrivr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
</body>

<div class="container">
<button class="btn btn-outline-primary">Primary</
button>
<button class="btn btn-outline-secondary">Secondary</
button>
<button class="btn btn-outline-success">Success</
button>
<button class="btn btn-outline-danger">Danger</button>
<button class="btn btn-outline-warning">Warning</
button>
<button class="btn btn-outline-info">Info</button>
<button class="btn btn-outline-light">Light</button>
<button class="btn btn-outline-dark">Dark</button></
div>
</body>

```

DROPDOWNS

Dropdowns are toggleable, text overlays for displaying lists of links and more. They made interactive with the included dropdown plugin. They are toggled by clicking, not by hovering; this is an intentional design decision.

Dropdowns are built on a third-party library, Popper.js, which provides dynamic positioning & viewport detection. Always include popper.min.js before bootstrap.bundle.min.js/bootstrap.bundle.js which contains Popper.js.

Example:

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<title>Bootstrap Example</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="HTTPS://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.slim.min.js"></script>
<script src="HTTPS://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"></script>
<script src="HTTPS://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet" href="HTTPS://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
<body>
<div class="container">
<!-- Example single danger button -->
<div class="btn-group">
<button type="button" class="btn btn-danger dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
Action
</button>
<div class="dropdown-menu">
<a class="dropdown-item" href="#">Action</a>
<a class="dropdown-item" href="#">Another action</a>
<a class="dropdown-item" href="#">Something else here</a>
<div class="dropdown-divider"></div>
<a class="dropdown-item" href="#">Separated link</a>
</div>
</div>
</body>

```

NAVBAR

How It Works

Navbars require a wrapping of `.navbar` with `.navbar-expand{<sm|-md|-lg|-xl>}` for responsive collapsing and scheme classes. Navbars and their contents are fluid by default. Use optional containers to limit horizontal width. It can use spacing and flex utility classes for the controlling of spacing and alignment within navbars. These are responsive by default, but you can modify them easily to change that. Responsive behavior depends on Collapse JavaScript plugin. Navbars are hidden by default when printing.

Force them to be printed by adding `.d-print` to the `.navbar`. Make sure you should use a `<nav>` element or, if using a more generic element such as a `<div>`, add a `role="navigation"` to every navbar to explicitly identify it as a landmark region for users of assistive technologies.

Supported Content

Navbar comes with built-in support of sub-components. Choose from the following as needed:

- `.navbar-brand` is used for company, product, or project name.
- `.navbar-nav` for a full-height and lightweight navigation.
- `.navbar-toggler` for used with our collapse plugin and other navigation toggling behaviors.
- `.norm-inline` used for any form controls and actions.
- `.navbar-text` used for adding vertically centered strings of text.
- `.collapse .navbar-collapse` used for grouping and hiding navbar contents by a parent breakpoint.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1">
  <script src="https://cdn.jsdelivr.net/npm/
jquery@3.6.0/dist/jquery.slim.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.
js@1.16.1/dist/umd/popper.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/
bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"></
script>
  <link rel="stylesheet" href="https://cdn.jsdelivr.
net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">

<body>

<nav class="navbar navbar-expand-lg navbar-light
bg-light">
```

```

<a class="navbar-brand" href="#">Navbar</a>
<button type="button" data-toggle="collapse"
class="navbar-toggler" data-
target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false"
aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse"
id="navbarNavAltMarkup">
  <div class="navbar-nav">
    <a class="nav-item nav-link active"
href="#">Home <span class="sr-only">(current)</span></
a>
    <a class="nav-item nav-link" href="#">Features</
a>
    <a class="nav-item nav-link" href="#">Pricing</
a>
    <a class="nav-item nav-link disabled" href="#"
tabindex="-1" aria-disabled="true">Disabled</a>
  </div>
</div>
</nav>
</body>

```

UTILITIES

Border

You can use border utilities to add or remove an element's borders. Here is the complete list from which you can choose all borders or one at a time.

CHAPTER SUMMARY

The entire chapter is about Bootstrap we have learned that Bootstrap can be used with CSS to enhance its properties we can make pages more attractive. Bootstrap can make it possible to do so. It can be added just by installing or adding CDN in your HTML file and you can start working on it.

FURTHER READING

A brief overview of the history of Bootstrap - <https://getbootstrap.com/docs/4.0/about/history/>, accessed on (2022 May 20).

Alerts - <https://getbootstrap.com/docs/4.1/components/alerts/>, accessed on (2022 May 21).

- Bootstrap (front-end framework) - [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)), accessed on (2022 May 20).
- Bootstrap 5- <https://blog.getbootstrap.com/2021/05/05/bootstrap-5/>, accessed on (2022 May 21).
- Bootstrap and its Features - <https://dzone.com/articles/bootstrap-and-its-features>, accessed on (2022 May 21).
- Bootstrap CDN - <https://www.belugacdn.com/bootstrap-cdn/>, accessed on (2022 May 21).
- Bootstrap Get Started - https://www.w3schools.com/bootstrap/bootstrap_get_started.asp, accessed on (2022 May 21).
- Bootstrap Grids - https://www.w3schools.com/bootstrap/bootstrap_grid_basic.asp, accessed on (2022 May 21).
- Bootstrap Templates - https://www.w3schools.com/bootstrap/bootstrap_templates.asp, accessed on (2022 May 21).
- Bootstrap Versions - https://www.w3schools.com/bootstrap/bootstrap_ver.asp, accessed on (2022 May 20).
- Brief history of Bootstrap - <https://studysection.com/blog/a-brief-history-of-bootstrap/>, accessed on (2022 May 20).
- Build fast, responsive sites with Bootstrap - <https://getbootstrap.com/>, accessed on (2022 May 20).
- Custom components - <https://getbootstrap.com/docs/4.0/examples/>, accessed on (2022 May 21).
- Features - [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)), accessed on (2022 May 20).
- Front-end framework for developing responsive - <https://cdnjs.com/libraries/bootstrap>, accessed on (2022 May 21).
- Getting started with Bootstrap - <https://getbootstrap.com/docs/3.3/getting-started/>, accessed on (2022 May 21).
- Grid system - <https://getbootstrap.com/docs/4.1/layout/grid/>, accessed on (2022 May 21).
- History of Bootstrap - <https://dev.to/yongdev/brief-history-of-bootstrap-3gi6>, accessed on (2022 May 20).
- Versions - <https://getbootstrap.com/docs/versions/>, accessed on (2022 May 20).
- What is a CDN - <https://www.akamai.com/our-thinking/cdn/what-is-a-cdn#:~:text=A%20CDN%20is%20a%20network,and%20stored%20elsewhere%20as%20needed>, accessed on (2022 May 21).
- What is Bootstrap - <https://www.techtarget.com/whatis/definition/bootstrap>, accessed on (2022 May 20).

Bulma

IN THIS CHAPTER

- Introduction
- Versions of Bulma
- Bulma installation
- Usage of Bulma

In the previous chapter, we talked about most famous CSS framework, i.e., Bootstrap but now, we will discuss another CSS framework used by many companies and developers, Bulma.

INTRODUCTION

Bulma is a modern open-source framework for developing web applications using CSS-only, and JavaScript is not required. Developed based on flexbox. Bulma is a CSS library. This means it offers CSS classes to help you style your HTML code. To use Bulma, you can use the pre-assembled .css file or add the .sass files so you can customize it according to your needs.

Requirement

You need to know the basic HTML and CSS languages

Install the Bulma CSS Library

Like any other CSS library, Bulma offers many ways to integrate into your app.

- CDN CSS link
- Npm Package
- Local installation

Using the npm Bulma Package

Contains Bulma package provided at npm library. You can use npm or you can use yarn or bower packager to install in your app as follows.

Go to the root directory of your application, and use the command below as per package manager.

```
$ npm install bulma  
or
```

```
$ bower install bulma  
or  
$ yarn add bulma
```

It installs the its module into your application and makes the following changes.

Features and Benefits

- Easy to use and easy to read
- Open source also supports major browsers
- Mobile-first design
- Flexbox integrated grid system
- Responsive design
- Modular: use only the required features
- NPM integration
- Supports modular structure

- SASS and SCSS support
- Customizing is very easy
- JavaScript is not required, and everything is coded in CSS

For CDN

You can use `"http://cdnjs.cloudflare.com/ajax/libs/bulma/0.9.3/css/bulma.min.css"`

Bulma is a CSS framework. Thus, the only output is a single CSS file: `bulma.css`

You can use that file, ‘out of the box’, or download Sass source files to customize the dynamics. No JavaScript installed. People usually want to use their JS implementation. Bulma can be considered a ‘natural agnostic’: it is just a style layer beyond imagination.

BROWSER SUPPORT

Bulma uses the autoprefixer to integrate (many) Flexbox features into earlier versions of the browser. According to Can I use, Bulma is compliant with the latest versions of:

- Chrome
- Edge
- Firefox
- Opera
- Safari

For Bulma to work properly, you need to get your web page responsive.

1. Use the HTML5 version doctype

```
<!DOCTYPE html>
```

2. You can also add the responsive viewport meta tag like this,

```
<meta content="width=device-width name="viewport",
initial-scale=1">.
```

Starter Bulma Template

If you want to get started away, you can use any HTML starter template.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta content="width=device
e-width, initial-scale=1" name="viewport">
<title>Hello Bulma!</title>
<link href="https://cdnjs.cloudflare.com/ajax/libs/
bulma/0.7.5/css/bulma.min.css" rel="stylesheet" >
<script src="https://use.fontawesome.com/releases/
v5.3.1/js/all.js"></script>
</head>
<body>
  <section class="section">
    <div class="container">
      <h1 class="title">
        Hello World
      </h1>
      <p class="subtitle">
        Our first page with <strong>Bulma</strong>!
      </p>
    </div>
  </section>
</body>
</html>
```

Let us start with the package.json file for managing dependencies.

```
npm init
```

You can download the npm package from the npm repository.

```
$ npm install Bulma
```

We need to import the Bulma's module into your CSS file.

```
@import "../node_modules/bulma/css/bulma.css";
```

We can also download the minified `bulma.min.css` file from [here](#) and include it in our webpage using the `<link>` tag.

```
<link rel="stylesheet" href="bulma.css" />
```

Column System

One of the most amazing features of Bulma is the responsiveness column grid system. For building a columns layout to every row in Bulma is defined by a columns container itself. Every column can be defined as a column and can be nested within the columns container.

Example:

```
<div class="columns has-text-centered">
  <div class="column has-background-success">1st
column</div>
  <div class="column has-background-danger">2nd
column</div>
  <div class="column has-background-info">3rd column</
div>
</div>
```

This snippet would create a responsiveness column layout with three columns of equal width. You can also define the column size by using the size classes.

```
<div class="columns has-text-centered">
  <div class="column has-background-success
is-half">1st column</div>
  <div class="column has-background-danger is-one-
quarter">2nd column</div>
  <div class="column has-background-info">3rd column</
div>
</div>
```

CUSTOMIZING VARIABLES¹

Bulma is highly customizable to 419 Sass variables across 28 files. To customize Bulma, there are four types or variables:

- Initial variables: These are global variables with real values.
 colors: \$blue: HSL(229, 53%, 53%)
 font sizes: \$size-1: 3rem
 dimensions: \$gap: 32px
 other values: \$easing: ease-out or \$radius-large: 6px. Here is the complete list of initial variable in the table.
- Derived variables: These are taken from other variables.
- \$primary: \$turquoise

\$link: \$blue

\$success: \$green

\$warning: \$yellow

\$danger: \$red

\$info: \$cyan

\$link: \$blue: you can use the links of the primary color

\$background: \$white-ter: a general background color

\$dark: \$grey-darker

\$text: \$grey-dark

- General variables: These are HTML elements that do not have CSS classes.
- Element variables: These are the Bulma-specific variables.

Bulma-Start

It is a tiny npm (node packages manager) package that includes the npm dependencies you need to build your own website with Bulma.

WITH NODE-SASS INSTALLATION²

Create a Package .json File

In your terminal, create a new folder called `firstbulma`, redirect to it, then type the following command:

```
$ npm init
```

It will launch a setup to create `package.json`. When it prompted for an entry point, enter `sass/mystyles.scss`.

Install the Dev Dependencies

You need two packages to customize Bulma: Bulma and `node-sass` itself.

```
$ npm install node-sass --save-dev
$ npm install bulma --save-dev
```

Your json file name “`package.json`” should look like this at this point.

```
{
  "name": "mybulma",
  "version": "1.0.0",
  "main": "sass/mystyles.scss",
  "license": "MIT",
  "devDependencies": {
    "bulma": "^0.7.2",
    "node-sass": "^4.9.2"
  }
}
```

CREATE A SASS FILE

Before we create the sass, let's talk about something about it. Sass (representing Syntactically Style Sheets) is a CSS extension that allows you to use things like variables, nest rules, in-line import and more. It also helps to keep things well organized and allows to create style sheets quickly. Sass is compatible with all its versions of CSS. The only requirement to use it is that you have to install Ruby. Users are also required to follow the Sass Community Guidelines.

Sass is more than technical; Sass is run by a community of people who have the potential for development and daily use. As a community, we want to very differences that have made our partnership stronger, and work together to provide the best possible environment for learning, nurturing, and sharing ideas. It is important that we keep Sass a fun, welcoming, challenging, and appropriate place to play.

Compliance with These Community Guidelines

We urge all private and public gatherings of the Sass Community to adhere to these rules of conduct and to take appropriate action to create a safe and accessible environment for all in attendance. This should include the publication of a Code of Conduct that covers specific anti-bullying procedures. Please contact us if you feel the need to deviate further from these guidelines by sending an email to:

We understand and agree that the “Sass Community” is a group of people who identify as Sass enthusiasts. We do not have the power to remove anyone from society. Finally, we can only use our voice to publicly declare that we do not agree with or support people or organizations that do not comply with these guidelines.

HOW TO USE SASS

In the next section, we will set out some basic tips for using Sass, using examples from the official Sass website. See Sass Documentation for additional references and examples.

Syntax:

Sass includes two syntax options:

- SCSS (Sassy CSS): Uses .scss file extension and fully compliant with CSS syntax
- Indent (simply called ‘Sass’): Uses the .sass file extension and underscore; is not fully compatible with CSS syntax, but is faster to type
- Note that files can also be converted from one syntax to another using the sass-convert command.

INSTALL SASS

There are many great apps that will allow you to use Sass in a few minutes for Mac, Windows, and Linux. You can download many apps for free but a few of them are paid apps (and totally worth it).

- CodeKit (Paid) Mac
- Ghostlab (Paid) Mac Windows
- Hammer (Paid) Mac
- LiveReload (Paid, Open Source) for Mac Windows
- Prepros (Paid) Mac Windows Linux
- Scout-App (Free, Open Source) Windows Linux Mac

LIBRARIES

The Sass team maintains two Node.js packages for Sass, both of which support the standard JavaScript API. The sass package is a clean, slow JavaScript package that can be installed on all Node.js supported forums. The embedded sass package wraps the JS API next to Dart VM, so it's fast but only supports Windows, Mac OS, and Linux. There are also community-based threats in the following languages:

- Java, including the Gradle plugin.

Install Anywhere (Standalone)

You can install Sass on Windows, Mac, or Linux by downloading your operating system package from GitHub and adding it to your PATH. That's all – there is no external dependency and nothing else you need to include **npm**.

If you are using Node.js, you can re-install Sass using npm for running

```
npm install -g sass
```

Now Create a Sass File

Create a sass folder to which you can add a file called mystyles.scss:

```
@charset "utf-8";
@import "../node_modules/bulma/bulma.sass";
```

Be sure to write the correct path for the Bulma.sass file.

Create an HTML Template Which Uses Various Bulma Components

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta HTTP-equiv="X-UA-Compatible"
content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>My custom Bulma website</title>
    <link rel="stylesheet" href="css/mystyles.css">
  </head>
  <body>
    <h1 class="title">
      CSS Framework - Bulma
    </h1>

    <div class="field">
      <div class="control">
        <input class="input" type="text"
placeholder="Input">
      </div>
    </div>

    <div class="field">
      <p class="control">
        <span class="select">
          <select>
            <option>Select dropdown</option>
          </select>
        </span>
      </p>
    </div>

    <div class="buttons">
      <a class="button is-primary">Primary</a>
      <a class="button is-link">Link</a>
    </div>
  </body>
</html>
```

Save this file as `mypage.html`. Open the page in your browser. You will see The unstyled page.

To build a CSS from a Sass file, we should use node scripts. In `package.json`, add the following:

```
"scripts": {
  "css-build": "node --sass --omit-source-map --url
sass/mystyles.scss css/mystyles.css",
  "css-watch": "npm run css-build -- --watch",
  "start": "npm run css-watch"
}
```

- CSS-build takes `sass/mystyles.scss` as input, and removes CSS/`mystyles.css`, while extracting the source map
- CSS-watch creates CSS and watches changes
- the start is simply a CSS clock shortcut.

To check it, enter your terminal and execute the following command:

```
$ npm run css-build
```

Reload the page & it should be styled.

Bulma's default styles. To see for changes, just launch the following command:

```
$ npm start
```

Now replace the content of the `sass/styles.scss` file with the following:

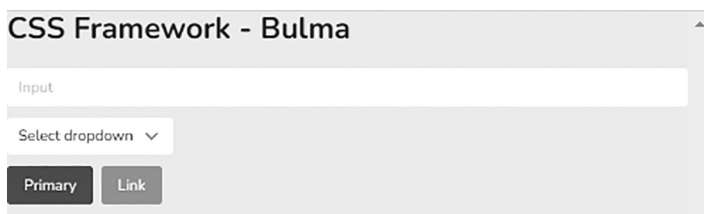
```
@charset "utf-8";
// Import a Google Font
@import URL('https://fonts.googleapis.com/
css?family=Nunito:400,700');
// Set your brand colors
$purple: #8A4D76;
$pink: #FA7C91;
$brown: #757763;
$beige-light: #D0D1CD;
$beige-lighter: #EFF0EB;
// Update Bulma's global variables
```

```

$family-sans-serif: "Nunito", sans-serif;
$grey-dark: $brown;
$grey-light: $beige-light;
$primary: $purple;
$link: $pink;
$widescreen-enabled: false;
$fullhd-enabled: false;
// Update some of Bulma's component variables
$body-background-color: $beige-lighter;
$control-border-width: 2px;
$input-border-color: transparent;
$input-shadow: none;
// Import only what you need from Bulma
@import "../node_modules/Bulma/sass/utilities/_all.sass";
@import "../node_modules/bulma/sass/base/_all.sass";
@import "../node_modules/Bulma/sass/elements/button.sass";
@import "../node_modules/bulma/sass/elements/container.sass";
@import "../node_modules/Bulma/sass/elements/title.sass";
@import "../node_modules/bulma/sass/form/_all.sass";
@import "../node_modules/Bulma/sass/components/navbar.sass";
@import "../node_modules/Bulma/sass/layout/hero.sass";
@import "../node_modules/Bulma/sass/layout/section.sass";

```

Since you are watching for changes, first save the file to see the result:



Bulma Sass Output

UTILITIES

You will know about Bulma mixes. Bulma Blends help users create a reusable CSS code throughout the website.

Bulma uses Sass Mix to create CSS output and is widely used within the context of the Bulma framework. Bulma has different types of mixins as given below;

- Element mixins: These mixes are used to create visual HTML elements.
- CSS mixins: These mixes are used to add CSS rules to the HTML element.
- Directional mixins: These combinations are used to add left or right alignment to the HTML element.
- Forms mix: These combinations are used with tor buttons and form controls created in Bulma.
- Responsive mixins: These compounds are used to describe different styles of different screen sizes.

Through the mix, no particular class is given to Bulma other than we create our own classes and style them using SASS mix.

Syntax:

```
.bulma-mixin {
  @include mixin();
}
```

Mixins³

These are custom combinations of custom elements and other CSS helpers. Throughout the codebase, Bulma uses Sass Mix to enhance and facilitate CSS output. Although these compounds are widely used in the context of Bulma, they are available for reuse in your projects.

These combinations form a visual HTML feature.

- The arrow
The arrow() mixin function creates a down-facing arrow. The \$color is a parameter that defines the color of the arrow.

```
.bulma-arrow-mixin {
  @include arrow(any_color_name);
}
```

Example:

```
<span class="Bulma-arrow-mixin"></span>
```

- Hamburger

The `hamburger()` mixin function creates a 16px×16px set of three horizontal bars grouped within square. These dimensions of the square are defined by the `$dimensions` parameter.

Syntax:

```
.bulma-hamburger-mixin {
  @include hamburger(dimensions);
}
```

Example:

```
<button class="Bulma-hamburger-mixin">
  <span></span>
  <span></span>
  <span></span>
</button>
```

- Delete

The `delete()` mixin function creates a 20 px×20px circle that contain a cross. It comes with three modifiers: `is-small`, `is-medium` and `is-large`.

Syntax:

```
.bulma-delete-mixin {
  @include delete;
}
```

Example:

```
<button class="bulma-delete-mixin is-small"></button>
<button class="bulma-delete-mixin"></button>
<button class="bulma-delete-mixin is-medium"></button>
<button class="bulma-delete-mixin is-large"></button>
```

- Loader
- The loader() mixin function creates a lem spinning circle.

Syntax:

```
.bulma-loader-mixin {
  @include loader;
}
```

Example:

```
<span class="Bulma-loader-mixin"></span>
```

- Font Awesome container
 - The fa() mixin that can creates a square inline-block element, for contain the Amazing Font icon, or any other type of icon font.
 - The first \$size parameter defines the font size of the icon.
 - The second \$dimensions parameter defines the size of a square container.

Example:

```
.bulma-fa-mixin {
  @ include fa(1rem, 2rem);
  background-color: lavender; // For demo purposes
}
```

Example:

```
<span class="Bulma-fa-mixin">
  <i class="fas fa-thumbs-up"> </i>
</span>
```

CSS Mixins

These mixins add CSS rules to the element.

- Block

The `block()` mixin function adds spacing below an element, but only if it is not the last child. The `$spacing` parameter is used as the value of the `margin-bottom`.

```
.bulma-block-mixin {
  @include block($spacing);
}
```

Example:

```
<p class="bulma-block-mixin">This element has a
margin-bottom.</p>
<p class="bulma-block-mixin">This element too.</p>
<p class="bulma-block-mixin">Not this one because it's
the last child.</p>
```

- Overlay

The `overlay()` mixin makes the element cover its most closest positioned ancestor.

The `$offset` parameter defines how far the inside element is positioned from each edge (top, bottom, left and right).

Syntax:

```
Bulma-overlay-mixin {
  @include overlay(1rem);
  background-color: orange;
  border-radius: 0.1em;
  color: white;
  opacity: 0.4;
  padding: 1.2em;
}
```

Example:

```
<div class="Bulma-overlay-mixin-parent">
  <div class="Bulma-overlay-mixin">Overlay element</div>
</div>
```

- Center

The `center()` mixin function allows to position an element with fixed dimensions at the center of its closest positioned ancestor.

The value of the `$width` argument should be the width of the element the mixin is applied on. It unless the element has four dimensions, the second parameter `$height` is required and its value should be the height of the element the mixin is applied on.

Syntax:

```
.bulma-center-mixin {
  @include center;
}
```

Example:

```
<div class="Bulma-center-mixin-parent">
  
</div>
```

- Placeholder

The `placeholder()` mixin function allows to change the style of an input's placeholder.

The `$offset` defines how far inside the element is positioned from each edge (top, bottom, left and right).

Syntax:

```
.bulma-placeholder-mixin {
  @include placeholder {
```

```

    color: light blue;
  }
}

```

Example:

```

<input class="input bulma-placeholder-mixin"
type="text" placeholder=" Styled placeholder" >

```

- Clearfix

The `clearfix()` mixin adds a `::after` pseudo-element with a `clear:` both rule.

Syntax:

```

.bulma-clearfix-mixin {
  @include clearfix;
}

```

Example:

```

<div class="Bulma-clearfix-mixin">
  
  <p>This is a image description.</p>
</div>
<p>The text is following the clearfix element and is
correctly placed after.</p>

```

- Reset

The `reset()` mixin function applies a soft style reset. This is especially useful for `<button>` elements.

Syntax:

```

.bulma-reset-mixin {
  @include reset;
}

```

Example:

```

<button>Default button</button>

```

```
<button class="Bulma-reset-mixin">Reset button</button>
```

- Unselectable

The `unselectable()` mixin function makes an element not selectable. This is to prevent the text from being selected when double-clicked.

```
.bulma-unselectable-mixin {
  @include unselectable;
}
```

Syntax:

```
<p>This text is selectable.</p>
<p class="Bulma-unselectable-mixin">The text is not
selectable.</p>
```

- Overflow Touch

The `overflow-touch()` mixin function add the `-webkit-overflow-scrolling: touch;` rule to the element.

Direction Mixins

These mixins are related to directions and rules to the element.

- Left-to-right and Right-to-left Mixins

Bulma has a global `$ rtl` flag, which allows the frame to export the CSS version from right to left. By default, the value of this flag is set to `false`. This means that the frame releases the Left-to-Right version.

Mixin `@mixin ltr` and `@mixin rtl` are here to release CSS rules based on the `$ rtl` value:

if `$ rtl: true`, `@include ltr` output nothing

if `$ rtl: false`, `@include rtl` output nothing

This is useful for features that are specific to the element.

- LTR Position

The `ltr-position ()` mixin is a quick way to switch between left and right CSS layouts when working with positioned objects.

The first `$ space` parameter defines the value of the offset, whether right or left.

The second `$ right` parameter determines whether the product comes from the right (default) or the left.

- Here is what the output looks like with a \$ space parameter set in 1rem:

Flag →	\$rtl: false;	\$rtl: true;
@include ltr-position(1rem, false)	left: 1rem	right: 1rem
@include ltr-position(1rem, true)	right: 1rem	left: 1rem

- LTR Property

The ltr-position () mixin works as an ltr-position () mixin but lets you choose which CSS asset to set. The mixin will add a right- or left-hand to the structure. This is especially useful for jeans and padding.

The first \$ property parameter is what feature you want to “search” left and right.

The \$ second parameter defines the value of the offset, either right or left.

The third right parameter defines whether the product comes from the right (default) or the left. Here’s what the results look like with a \$ space parameter set to 1rem.

Flag →	\$rtl: false;	\$rtl: true;
@include ltr-property(“margin”, 1rem, true)	margin-right: 1rem	margin-left: 1rem
@include ltr-property(“margin”, 1rem, false)	margin-left: 1rem	margin-right: 1rem

RESPONSIVE MIXINS

Responsive Mixins that allows to define different styles for each screen size.

from() and until() mixins

The responsiveness in CSS is based on media queries. Bulma provides two useful responsive mixins:

- `@mixin from($breakpoint)` is used to target devices with a screen wider than or equal to the breakpoint
- `@mixin until($breakpoint)` is used to target devices with a screen narrower than the breakpoint

`from ()`

The `mixin function from ()` has one parameter that sets the width of the screen to use styles that contain:

Sass Source Code:

```
.element {
  background: red;
  @include from (1280px) {
    background: blue;
  }
}
```

CSS Code:

```
.element {
  background: red;
}

@media screen and (min-width: 1280px) {
  .my feature {
    background: blue;
  }
}
```

Description

On screens 1279px wide or less, the background of the element will be red.

At 1280px-wide or higher screens, the background of the element will be blue.

`until ()`

The `until()` mixin has a single parameter that can set the screen width (minus “1px”) until that the styles it contains will be applied. This means that if you set a value of 1280px, styles will apply to the screen width of 1279px but not to the screen width of 1280px. The reason for this 1px

removal is to allow you to use both from () and up to () the same breakpoint value. This leaves no gap between the two rule sets.

Sass Source:

```
$ breakpoint: 1280px;

. element {
  @include until ($ breakpoint) {
    background: green;
  }

  @include from ($ breakpoint) {
    background: orange;
  }
}
```

CSS output:

```
@media screen and (maximum width: 1279px) {
  .element {
    background: green;
  }
}

@media screen and (min-width: 1280px) {
  . element {
    background: orange;
  }
}
```

On screens 1279px wide or less, the background of the element will be green.

On 1280px-wide or higher screens, the background of the element will be orange.

By having four breakpoints and supporting five screen sizes, Bulma can support many different settings.

While you can use mixins

```
@include from ()
and
@include to ()
```

Bulma provides quick shortcuts with 11 named mixins

These responsive combinations are named for screen size and breakout points used in Bulma, so you can use them to create responsive designs.

- Mobile Up to 768px: Tablet
- Between 769px and 1023px: Desktop
- Between 1024px and 1215px: Widescreen
- Between 1216px and 1407px: FullHD
- 1408px and above

FORM CONTROL MIXINS

These Mixins are for Bulma's buttons and form controls in Bulma, the form controls are an essential part of the framework. They comprise the following elements:

- `.button`
- `.input`
- `.select`
- `.file-cta.file-name`
- `.pagination-previous` `.pagination-next` `.pagination-link` `.pagination-ellipsis`

The `control()` mixin ensures consistency by providing a set of styles shared between all these form controls. You can use it to create additional controls:

Example:

```
<button class="Bulma-control-mixin">
  My control
</button>
```

SIZES

It can controls have a default font size of `$size-normal` and also come in three additional sizes, which can be accessed via three additional mixins:

```
@include control-small; with a font size $size-small
@include control-medium; with a font size $size-medium
@include control-large; with a font size $size-large
```

Syntax:

```
.bulma-control-mixin {
  &.is-small {
    @include control-small;
  }

  &.is-medium {
    @include control-medium;
  }

  &.is-large {
    @include control-large;
  }
}
```

CONTROL PLACEHOLDER

The `control()` mixin exists as Sass placeholder `%control`.

Syntax:

```
.bulma-control-extend {
  @extend %control;
  background: blue;
  color: white;
}
```

Example:

```
<button class="Bulma-control-extend">
  My control
</button>
```

FUNCTIONS

The system works to calculate colors and other values. Bulma has custom Sass functions to help find alternately Related colors:

- `findColorInvert ($ color)`: returns about 70% black or 100% dark white depending on light color
- `findLightColor ($ color)`: restores current color with at least 96% brightness
- `findDarkColor ($ color)`: restores current color with at least 29% brightness

Bulma also has a number of useful calculation functions:

- `powerNumber ($ number, $ exp)`: calculates the number of numbers displayed in another. Returns the number
- `colorLuminance ($ color)`: describes the color black or light. Return the decimal number between 0 and 1 when ≤ 0.5 is dark and > 0.5 light.

FINDCOLORINVERT () FUNCTION

The `findColorInvert ($ color)` function takes color as an input, and renders the black rgba transparent (`# 000, 0.7`) or white `#fff`:

- if `colorLuminance ($ color) > 0.55`, output rgba (`# 000, 0.7`)
- otherwise, it releases `#fff`

Its purpose is to ensure a readable shade of text where input color is used as a background.

The `findLightColor ()` and `findDarkColor ()` function

The `findLightColor ($ color)` and `findDarkColor ($ color)` functions take color as input, and produce light and dark color versions, respectively.

color	findLightColor	findDarkColor
hsl(171deg, 100%, 41%)	hsl(171deg, 100%, 96%)	hsl(171deg, 100%, 29%)
HSL(229deg 53% 53%)	HSL(229deg 52% 96%)	HSL(229deg 53% 47%)
HSL(153deg 53% 53%)	HSL(153deg 52% 96%)	HSL(153deg 53% 31%)
HSL(44deg 100% 77%)	HSL(45deg 100% 96%)	HSL(44deg 100% 29%)

EXTENDS

Sass expands to keep your CSS code DRY. In Bulma, many items share a collection of styles. While the mix allows for sharing, it repeats the CSS rules each time they are applied.

To avoid duplication, Bulma uses the `@extend` rule code to share code. The law tells the Cassis that one voter must inherit another's style. Learn more about the law of extension. Instead of creating CSS classes that can be used to be a source of style sets, Bulma uses Sass pronouns:

- `%control`
- `%unselectable`
- `%arrow`
- `%block`
- `%delete`
- `%loader`
- `%overlay`
- `%reset`

Each of these placeholders are simply the `@extend` version of their corresponding mixins.

CHAPTER SUMMARY

Today, we have learned that Bulma can be used with CSS to enhance features. It can be added just by installing or adding CDN in your HTML file and you can start working on it. There are various minor concepts you need to learn if necessary, you can follow it official documentation.

FURTHER READING

A Simple and Intuitive CSS Framework - <https://blog.openreplay.com/introduction-to-bulma-a-simple-and-intuitive-css-framework/>, accessed on (2022 May 23).

Bulma - <https://dev.to/t/bulma>, accessed on (2022 May 23).

Bulma - Introduction - https://www.tutorialspoint.com/bulma/bulma_introduction.htm, accessed on (2022 May 23).

Bulma Introduction - <https://www.geeksforgeeks.org/bulma-introduction/>, accessed on (2022 May 23).

Bulma: the modern CSS framework - <https://bulma.io/>, accessed on (2022 May 23).

Features of Bulma CSS framework - <https://mobiosolutions.com/the-introduction-and-features-of-bulma-css-framework/>, accessed on (2022 May 23).

Getting started - <https://app.wftutorials.com/Introduction-to-Bulma-a-CSS-framework>, accessed on (2022 May 23).

Learn Bulma CSS in 5 minutes - <https://www.freecodecamp.org/news/learn-bulma-in-5-minutes-ec5188c53e83/>, accessed on (2022 May 23).

The modern CSS framework - <https://versions.bulma.io/0.9.3/>, accessed on (2022 May 23).

NOTES

- 1 Variables in Bulma – Bulma.
- 2 With node-sass – Bulma.
- 3 Bulma with webpack – Bulma.



Taylor & Francis

Taylor & Francis Group

<http://taylorandfrancis.com>

Tailwind

IN THIS CHAPTER

- Introduction
- Tailwind installation
- Usage of Tailwind

In the previous chapter, we talked about Bulma CSS framework, and now, we will discuss another CSS framework, Tailwind.

INTRODUCTION

Tailwind CSS is the first CSS Utility framework to create a custom UI. It is a customized, low-quality CSS framework that gives you all the building blocks you need. Also, it's a cool way to write a line style and gain good interaction without writing a single line of your CSS.

Tailwind CSS is a stylish, first-aid, low-level CSS framework that gives you a complete set of cross-country browsing classes and usable resources. These classes can be used on your web page to add your unique look and feel to your website. All known that 'Tailwind is useful first' because instead of using traditional framework classes, it provides the minimal resources needed to style any HTML object, such as finish, template, width, and length. Tailwind is so simple that can you understand the principles of naming and patterns, and you can almost guess many functions without the need for text.

WHY TAILWIND CSS?

As we know, there are many CSS frameworks but people always choose a quick and easy to read and use framework for the project. Tailwind comes with many features and styles created by users to choose from and is used to reduce the tendency to write CSS code and create a beautiful custom UI. It will help you to overcome difficult tasks. Tailwind CSS creates small resources with a defined set of options that allow easy integration of existing classes directly into HTML code.

WHO SHOULD USE TAILWIND CSS?

Tailwind is not for beginner web developers (HTML, CSS, or JS). Anyone interested in learning frontend programming should start with the basics of CSS. Before using any CSS frameworks, make sure you are comfortable with CSS layouts.

Now, if you are a person with a respectable CSS expertise and are looking for a pre-packaged style solution that will allow you to enter a little code and create a beautiful responsive structure. Then, Tailwind CSS will simplify your work.

Tailwind CSS is also a great option if you are a team working on major projects. The Tailwind Class Picker provides a universal and safe way to add style to web objects. In addition, universal class names allow the group to build quickly, consistently, and duplicate small code.

GET STARTED WITH TAILWIND CSS¹

Tailwind CSS works by scanning all of HTML files, JavaScript components, and other templates for classes names, generating the corresponding styles and then writing them to a static CSS file. It is fast, flexible, and reliable with zero-runtime.

Install Tailwind CSS

The simplest and fastest way to get up and working with Tailwind CSS framework from scratch is with the Tailwind CLI tool.

To install `tailwindcss` using `npm`, you need to create your `tailwind.config.js` file.

```
$ npm install tailwindcss
$ npx tailwindcss init
```

You can install `tailwind` by using the `yarn` command:

```
$ yarn add tailwindcss
```

then configure template paths and add the paths to all of your template files in your `tailwind.config.js` file.

Example:

```
module.exports = {
  content: [\"./src/**/*.{html,js}\"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Add the Tailwind Directives to CSS

Add the `@tailwind` directives for each of Tailwind's layers to your main CSS file.

Example:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

Now Start the Tailwind CLI Build Process

Run the CLI tool to scan template files for classes and build your CSS.

Start Using Your Tailwind in HTML

Add compiled CSS file to the `<head>` and start using Tailwind's utility classes to style HTML file content.

CDN

Use the CDN to try Tailwind right in the browser without any downloading step. The CDN is designed for development purposes only and is not the best choice for production itself.

Add the CDN Script to HTML

Add the CDN script tag to the `<head>` of your HTML file, and start using Tailwind's utility classes to style content.

Example:

```
<script src="https://cdn.tailwindcss.com"></script>
```

Here, we have simple example of Tailwind CSS. In this, we create form using

```
<!DOCTYPE html>
<html lang="en">

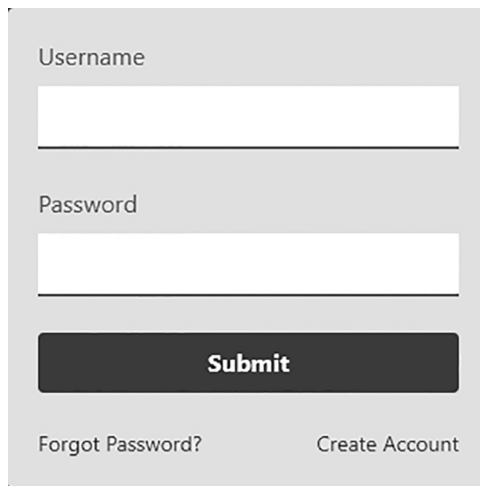
<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
</head>
<body>
  <body class="flex h-screen bg-indigo-700">
    <div class="w-full max-w-xs m-auto bg-indigo-100
rounded p-5">
      <form>
        <div>
          <label class="block mb-2 text-
indigo-500" for="username"> Username </label>
          <input class="w-full p-2 mb-6 text-
indigo-700 border-b-2 border-indigo-500 outline-none
focus:bg-gray-300" type="text" name="username">
        </div>
        <div>
          <label class="block mb-2 text-
indigo-500" for="password"> Password </label>
          <input class="w-full p-2 mb-6 text-
indigo-700 border-b-2 border-indigo-500 outline-none
focus:bg-gray-300" type="password" name="password">
        </div>
        <div>
          <input class="w-full bg-indigo-700
hover:bg-pink-700 text-white font-bold py-2 px-4 mb-6
rounded" type="submit">
        </div>
      </form>
      <footer>
        <a class="text-indigo-700 hover:text-
pink-700 text-sm float-left" href="#"> Forgot
Password? </a>
```

```

        <a class="text-indigo-700 hover:text-
pink-700 text-sm float-right" href="#"> Create Account
    </a>
    </footer>
</div>
</body>
</body>

</html>

```



Tailwind – Form

TAILWIND TYPOGRAPHY

Font Family

It accepts a lot of font names in tailwind in which all the properties are covered in class form. It is the alternative to the CSS font-family property. It is used to specify the font of an element. It can have multiple fonts as a backup.

Font family classes are as follows:

- font-sans
- font-serif
- font-mono

Example:

```

<!DOCTYPE html>
<head>
  <link rel="stylesheet" href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
  >
</head>

<body class="text-center mx-3 space-y-2">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind CSS Font Family Class
  </h1>
  <b></b>

  <div class="mx-24 bg-green-200">
    <p class="font-sans p-4">
      Font-Family - font-sans
    </p>

    <div class="mx-24 bg-green-200">
      <p class="font-serif p-4">
        Font-Family - font-serif
      </p>
    </div>
    <div class="mx-24 bg-green-200">
      <p class="font-mono p-4">
        Font-Family - font-mono
      </p>
    </div>
  </div>
</body>

</html>

```

Tailwind CSS Font Family Class

Font-Family - font-sans

Font-Family - font-serif

Font-Family - font-mono

Font Family


```

        h1 {
            text-align: center;
        }
    </style>
</head>

<body>
    <h1 style="color:green">
        Padding Classes in Tailwind CSS

    </h1><br />

    <!-- Padding -->
    <div class="container">
        <div class=" p-20">
            This is p class
        </div>
    </div>

    <br>
    <div class="container">
        <div class=" pb-20">
            This is pb-auto class
        </div>
    </div>
    <br>

    <div class="container">
        <div class=" pr-20">
            This is pr-20 class
        </div>
    </div>
    <br>

    <div class="container">
        <div class=" pt-20">
            This is pt-20 class
        </div>
    </div>
    <br>

    <div class="container">
        <div class=" pl-20">
            This is pl-20 class
        </div>
    </div>

```

```

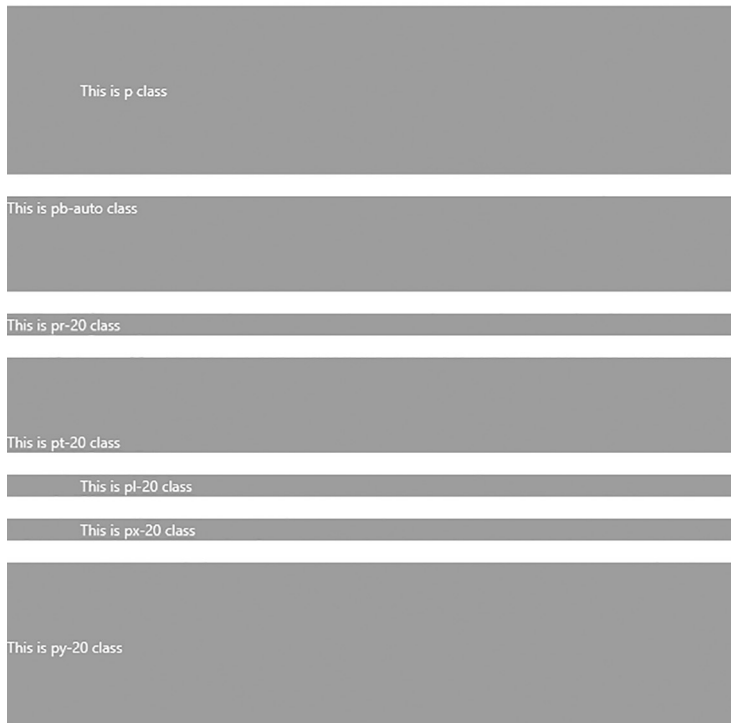
</div>
<br>
<div class="container">
  <div class=" px-20">
    This is px-20 class
  </div>
</div>
<br>

<div class="container">
  <div class=" py-20">
    This is py-20 class
  </div>
</div>
<br>
</div>
</body>
</html>

```

Output:

Padding Classes in Tailwind CSS

**Padding Classes**

Here is the complete example of the Margin Classes in Tailwind:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
  <style>
    .container {
      background-color: rgb(2, 201, 118);
      color: white;
      max-width: 800px;
      margin: 0 auto;
    }

    h1 {
      text-align: center;
    }
  </style>
</head>

<body>
  <h1 style="color:green">
    Margin Classes in Tailwind CSS

  </h1><br />

  <!-- Padding -->
  <div class="container">
    <div class=" m-20">
      This is m class
    </div>
  </div>

  <br>
  <div class="container">
    <div class=" mb-20">
      This is mb-auto class
    </div>
  </div>
  <br>

```

```
<div class="container">
  <div class=" mr-20">
    This is mr-20 class
  </div>
</div>
<br>

<div class="container">
  <div class=" mt-20">
    This is mt-20 class
  </div>
</div>
<br>

<div class="container">
  <div class=" ml-20">
    This is ml-20 class
  </div>
</div>
<br>
<div class="container">
  <div class=" mx-20">
    This is mx-20 class
  </div>
</div>
<br>

<div class="container">
  <div class=" my-20">
    This is my-20 class
  </div>
</div>
<br>
</div>
</body>

</html>
```

Margin Classes in Tailwind CSS

This is m class

This is mb-auto class

This is mr-20 class

This is mt-20 class

This is ml-20 class

This is mx-20 class

This is my-20 class

Margin Classes

Breakpoints in tailwind CSS are as follows:

Breakpoint	min-width
sm (small)	640px
md (medium)	768px
lg (large)	1024px
xl (X-large)	1280px
2xl (XX-double)	1536px

KEY CONCEPTS

- Provides API for your design system. Tailwind service sections help the user to work within the system limits instead of dumping your style sheets at special values.

- They make it easy to match the color choices, space, typing, shadows, and everything else that makes a well-designed design system.
- Tailwind is low-level, never encourages you to build/design a dual site. Even with a palette of the same color and scale, it is easy to build the same component.
- It automatically removes all unused css when you build a production which means that your last bit of css is very small which can usually be <10kb.
- Tailwind lets you create responsive designs right on your html. You can increase the screen size before any use stage and you will use it in a specific area.
- You can style different things on mood swings in stick hover, focus, active, disabled, internal focus, visual focus, or tailwind's group-hover, and you can write something in these mood swings.
- The Tailwind CSS IntelliSense extension integrates well with the conflicting code that gives you smart suggestions for auto-completion, integration, class descriptions, and more.
- Tailwind provides first-class css grid, integrated integration with css-enabled gradients, support for state-of-the-art options such as focus-visible.
- Use `tailwind.config.js` to customize your design system and let tailwind transform into your custom css framework.

Just-In-Time (JIT) is a very powerful, much-needed engine for tailwind css v2.1 +. Includes Tailwind css that produces your styles when you need them most as you write your templates instead of producing everything in advance during the initial construction.

How Do We Use JIT Engine Mode?

If you want to enable timely mode, set the mode option to log in to the `tailwind.config.js` file. You will need to configure the cleanup option in `tailwind.config.js` with all your template options.

JIT MODE FEATURES?

- **Lightening fast build times:** This library can integrate with major projects in about 800 ms (with a fast-growing construction of about 3 ms), no matter what tool you use. This usually takes 3–8 s to combine first using CLI and 30–45 s in the webpack.
- **All variations are enabled out of the box** as the library produces styles in high demand, you can use any variation you want, whenever you want. You can pack them as `sm: hover: active: disabled: opacity-75`.
- **Your CSS is the same in development and production** you do not need to remove unused production styles as you see css exactly everywhere.
- **Improved browser performance in development** as development constraints are as small as production builds, the browser does not need to analyze and manage many pre-produced css megabytes.

TAILWIND CSS BOX SIZING

This section accepts more than one value in tailwind CSS for all class-covered layouts. It is one of the places to rate CSS boxes. This section is used to define how a user should calculate the total width and length of an object, i.e., padding and borders, which should be installed or not.

Box size:

- `box-border`
- `box contents`

Box border: In this mode, the width and height of the structures include content, pads, and parameters that is, if we set the feature width to 200 pixels, those 200 pixels will fit any border or complement we added, and the content box will blur to 200 pixels plus that extra width. This usually makes the size of the elements much easier.

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
  <style>
    .container {
      background-color: rgb(2, 201, 118);
      color: white;
      max-width: 800px;
      margin: 0 auto;
    }

    h1 {
      text-align: center;
    }
  </style>
</head>
<body>

  <h1 class="text-green-600 text-2xl font-bold ">
    Box Border Tailwind CSS
  </h1>
  <div class="box-border bg-red-50 border w-64 p-4
mx-auto">
    Box Border
  </div>

  <br>
  <h1 class="text-green-600 text-2xl font-bold">
    Box Content Tailwind CSS
  </h1>
  <div class="box-content h-28 w-40 p-4 border-4
bg-yellow-500 mx-auto">
    Box Content </div>
</body>

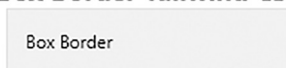
</html>
</body>

</html>

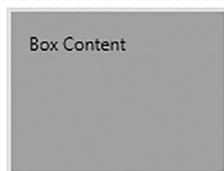
```

Output will be,

Box Border Tailwind CSS



Box Content Tailwind CSS



Box Content and Border Content

TAILWIND CSS DISPLAY

This section accepts more than one value in CSS for tailwind. All buildings are covered as in the classroom. It is one of the CSS displays. This class is used to describe how sections (div, hyperlink, heading, etc.) will be placed on a web page. As the name suggests, this feature is used to describe the display of different parts of a web page.

Display Tailwind CSS Classes

- **block:** Used to display an object as a block element.
- **inline-block:** Used to display a feature such as an intermediate block container.
- **inline:** Used to display an object as an inline element.
- **flex:** Used to display a feature like a block-level flex container.
- **inline-flex:** Used to display a feature as a flexible line level flexible container.

Let's have a single example of each of the above class right below:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
```

```

    <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
    <style>

    </style>
</head>
<body>

    <h1 class="text-green-600 text-5xl text-center
font-bold">
    <b> Tailwind CSS Display Classes
    </h1>
<div class="bg-gray-200 p-4 mx-16">
    <b> Tailwind CSS block Class <b>
    <span class="block h-12 w-2/5 bg-pink-500 ">1
</span> <br>
    <span class="block h-12 w-3/5 bg-red-500 "> 2
</span><br>
    <span class="block h-12 w-4/5 bg-yellow-500 "> 3
</span>
</div>

    <div class="bg-green-200 p-4 mx-48 space-y-4">
    <b> Tailwind CSS Inline Block Class <b>
    <span class="inline-block w-48 h-12
bg-green-500"> 1 </span>
    <span class="inline-block w-48 h-12
bg-green-500"> 2 </span>
    <span class="inline-block w-48 h-12
bg-green-500"> 3 </span>
    </div>

    <div class="bg-green-200 p-4 mx-48 space-y-4">
    <b> Tailwind CSS Inline Class <b>
    <span class="inline w-48 h-12 bg-green-500">
Box number 1 </span>
    <span class="inline w-48 h-12 bg-green-500">
Box number 2 </span>
    <span class="inline w-48 h-12 bg-green-500">
Box number 3 </span>
    </div>

```

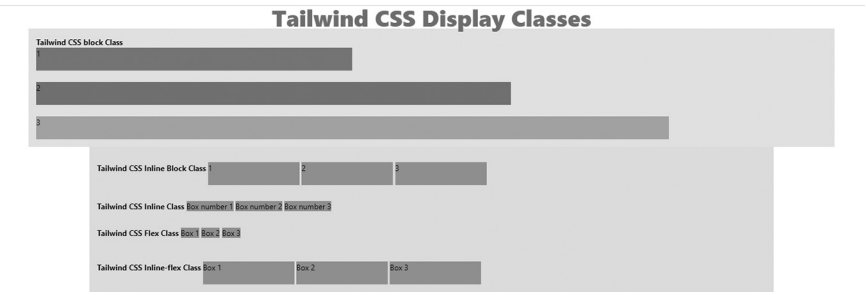
```
<div class="bg-green-200 p-4 mx-48 space-y-4">
  <b> Tailwind CSS Flex Class <b>
    <span class="flex-1 w-48 h-12 bg-green-500">
Box 1 </span>
    <span class="flex-1 w-48 h-12 bg-green-500">
Box 2 </span>
    <span class="flex-1 w-48 h-12 bg-green-500">
Box 3 </span>
</div>

<div class="bg-green-200 p-4 mx-48 space-y-4">
  Tailwind CSS Inline-flex Class
    <span class="inline-flex w-48 h-12 bg-green-
500">Box 1</span>
    <span class="inline-flex w-48 h-12 bg-green-
500">Box 2</span>
    <span class="inline-flex w-48 h-12 bg-green-
500">Box 3</span>
  </div>
</body>

</html>
</body>

</html>
```

Output:



Display Classes (Block, Inline, Flex)

- table: Used to set behavior as <table> for all elements.
- Table-caption: Used to set behavior as <caption> for all features.
- table-cell: Used to set behavior as <td> for all features.

- `table-column`: Used to set behavior such as `<col>` in all aspects.
- `table-column-group`: Used to set behavior as the `<column>` of all elements.
- `table-footer-group`: Used to set behavior as `<footer>` for all features.
- `table-header-group`: Used to set behavior as `<header>` for all elements.
- `table-row-group`: Used to set behavior such as `<row>` in all aspects.
- `Table-row`: Used to set behavior as `<tr>` for all features.

Here we have simple example of table right below,

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
  <style>

    .table, .table-row , .table-cell{
      border: 1px solid green;
    }
  </style>
</head>
<body>
  <h1 class="text-green-600 text-5xl text-center
font-bold "> Table </h1><br>
  <div class="table w-screen h-64">
    <div class="table-header-group">
      <div class="table-row">
        <div class="table-cell text-left"> 1 </div>
        <div class="table-cell text-left"> 2 </div>
        <div class="table-cell text-left"> 3 </div>
      </div>
    </div>
    <div class="table-row-group">
      <div class="table-row">
        <div class="table-cell text-center"> 11 </div>
        <div class="table-cell text-center"> 22 </div>
        <div class="table-cell text-center"> 33 </div>
      </div>
```

```
<div class="table-row">
  <div class="table-cell text-center"> 111
</div>
  <div class="table-cell text-center"> 2 22
</div>
  <div class="table-cell text-center"> 3 33
</div>
</div>
<div class="table-row">
  <div class="table-cell text-center"> 11 11
</div>
  <div class="table-cell text-center"> 22 22
</div>
  <div class="table-cell text-center"> 33 33
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

Table

11	22	33
111	222	333
1111	2222	3333

Display Classes (Table)

- flow-root: Used to set default value.
- grid: Used to display a feature such as a block-level grid container.

Here, we have simple example of grid right below:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
```

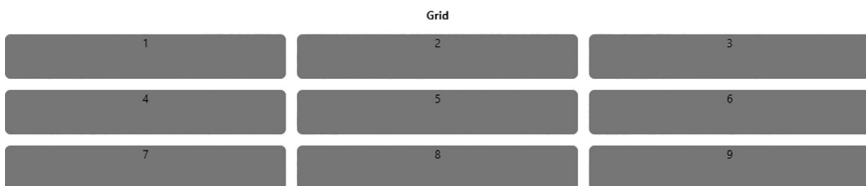
```

<style>
  p{
    background-color: rgb(217, 85, 217);
  }
</style>
</head>
<body>
<div class="container mx-auto">
  <h1 class="text-green-600 text-5xl text-center font-
bold" >
    <b> Tailwind CSS Display Classes <b> </h1> <br>
    <h2 class="text-black-600 font-bold text-center">
Grid</h2>

  <div class="grid gap-4 grid-cols-3 grid-rows-3 text-
center h-64 mx-32 p-4 ">
    <p class="rounded-lg"> 1 </p>
    <p class="rounded-lg"> 2 </p>
    <p class="rounded-lg"> 3 </p>
    <p class="rounded-lg"> 4 </p>
    <p class="rounded-lg"> 5 </p>
    <p class="rounded-lg"> 6 </p>
    <p class="rounded-lg"> 7 </p>
    <p class="rounded-lg"> 8 </p>
    <p class="rounded-lg"> 9 </p>
  </div>
</div>
</body>
</html>

```

Tailwind CSS Display Classes



Display Classes (Grid)

- inline grid: Used to display a feature such as an inline level grid container.
- Contents: Used for disappearing container.
- hidden: Used to remove feature.

FLOAT CLASSES

This section accepts more than one value in CSS for tailwind. It is one of the floating CSS sites. The float section describes the flow of content to control the folding of content in an object.

- float-right: It used to make the element float on the right side of the container.

Syntax:

```
<element class="float-right"> Write your content here
</element>
```

- float-left: It used to make the element float on the left side of the container.

Syntax:

```
<element class="float-left"> Write your content here
</element>
```

- float-none: It used to make the element float at the default place. We are using the center tag so the image is placed at the center.

Syntax:

```
<element class="float-none"> Write your content here
</element>
```

Here, we have simple example of float right below:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link href= "https://unpkg.com/tailwindcss@^2/
dist/tailwind.min.css" rel="stylesheet">
  <style>

.float-left{
  background-color: azure;

}
.float-right{
  background-color: beige;
}
</style>
</head>
<body>
  <div class="container-fluid">
    <h1 class="text-green-600 text-5xl text-center
font-bold"> Float Classes: </h1> <br>
    <p class="float-left w-2/5 text-black-600 text-
5xl "> Float Left </p>
    <p class="float-right w-2/5 text-black-600 text-
5xl "> Float Right </p><br>
    <p class="float-none w-2/5text-black-600 text-
center"> Float none</p>
  </div>
</body>

</html>

```

Output:

Float Classes:		
Float Left	Float none	Float Right

Float (Right, None, Left)

TAILWIND CSS CLEAR

This class accepts more than one or two value in tailwind CSS. All the properties that are covered as in class form. It is the parallel to the CSS clear property and used to define which side of floating elements are not allowed to float. It sets or returns back the position of the element in relation to floating objects. If your element can fit horizontally in the next space to another element that is floated. It is used for controlling the wrapping of the content around an element.

CSS CLEAR CLASSES

- clear-left: This section is used to determine if elements are not allowed to float on the left side in relation to something else.
- clear-right: This section is used to determine whether elements are not allowed to float right in relation to another object.

Here, we have simple example of clear right below:

```
<!DOCTYPE html>
<head>
  <link rel="stylesheet" href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
>
</head>
<style>
  .float-left{
    height: 200px;
    width:200px;
    background-color: aquamarine;
  }
  .float-right{
    height: 200px;
    width:200px;
    background-color: blanchedalmond;
  }
</style>
<body class="text-center">
  <b class=" text-green-600 text-5xl font-bold">
Tailwind CSS clear Class <b>
```

```

<div class="m-12 mx-56">
  <b class=" text-blue-600 text-xl font-bold ">
Clear Class - Left </b>
  <div class="float-left p-2 w-20 h-20 " ></div>
  <p class="text-justify clear-left">
    Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
    Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
    Curabitur non imperdiet nunc, vel fringilla
nunc.
    Donec elementum ultricies velit maximus
suscipit. Duis non diam
    vel mi venenatis mattis a interdum erat. Cras
malesuada ex purus, e
    get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
    roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
    ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
    leo vehicula accumsan. Etiam eleifend velit in
justo tristique
    facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
    blandit nunc venenatis id. Pellentesque
commodo augue elit, at
    sagittis diam convallis in. </p>
</div>

```

```

<div class="m-12 mx-56">
  <b class=" text-blue-600 text-xl font-bold ">
Clear Class - Right </b>
  <div class="float-left p-2 w-20 h-20 " ></div>
  <p class="text-justify clear-right">
    Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
    Sed lorem mauris, aliquam ac tempor sed, malesuada
et ligula.
    Curabitur non imperdiet nunc, vel fringilla nunc.
    Donec elementum ultricies velit maximus suscipit.
Duis non diam

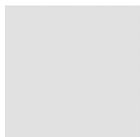
```

```

    vel mi venenatis mattis a interdum erat. Cras
malesuada ex purus, e
    get tristique nibh sollicitudin et. Duis viverra
rutrum tincidunt. P
    roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
    ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
    leo vehicula accumsan. Etiam eleifend velit in
justo tristique
    facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
    blandit nunc venenatis id. Pellentesque commodo
augue elit, at
    sagittis diam convallis in. </p>
</div>
</body>
</html>

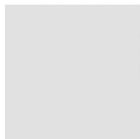
```

Tailwind CSS clear Class



Clear Class - Left

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at sagittis diam convallis in.



Clear Class - Right

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at sagittis diam convallis in.

Clear Classes (Left, Right)

- **clear-both:** This section is used to determine whether floating elements are not allowed to float on either side.
- **clear-none:** This category is used to specify a default value. It allows the elements to float on both sides.

```

<!DOCTYPE html>
<head>
  <link href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
  rel="stylesheet">
</head>
<style>
  .float-left{
    height: 200px;
    width:200px;
    background-color: aquamarine;
  }
  .float-right{
    height: 200px;
    width:200px;
    background-color: blanchedalmond;
  }
</style>
<body class="text-center">
  <b class=" text-green-600 text-5xl font-
bold">Tailwind CSS clear Class </b>
  <div class="m-12 mx-56">
    <b class=" text-blue-600 text-xl font-bold ">
Clear Class - None </b>
    <div class="float-left p-2 w-20 h-20 " ></div>
    <p class="text-justify clear-none">
      Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
      Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
      Curabitur non imperdiet nunc, vel fringilla
nunc.
      Donec elementum ultricies velit maximus
suscipit. Duis non diam
      vel mi venenatis mattis a interdum erat. Cras
malesuada ex purus, e
      get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
      roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle

```

```

        ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend velit in
justo tristique
        facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
        sagittis diam convallis in nibh sollicitudin
et. Duis viverra rutrum tincidunt. P
        roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
        ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend velit in
justo tristique
        facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
        sagittis diam convallis in. </p>
</div>
<br>

<div class="m-12 mx-56">
    <b class=" text-blue-600 text-xl font-bold ">
Clear Class - Both
    <div class="float-left p-2 w-20 h-20 " ></div>
    <p class="text-justify clear-both">
        Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
        Sed lorem mauris, aliquam ac tempor sed, malesuada
et ligula.
        Curabitur non imperdiet nunc, vel fringilla nunc.
        Donec elementum ultricies velit maximus suscipit.
Duis non diam
        vel mi venenatis mattis a interdum erat. Cras
malesuada ex purus, e
        get tristique nibh sollicitudin et. Duis viverra
rutrum tincidunt. P
        roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
        ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed

```

```

        leo vehicula accumsan. Etiam eleifend velit in
justo tristique
        facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
        blandit nunc venenatis id. Pellentesque commodo
augue elit, at
        sagittis diam convallis in. </p>
</div>
</body>
</html>

```

Tailwind CSS clear Class

Clear Class - None

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at sagittis diam convallis in.

Clear Class - Both

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at sagittis diam convallis in.

Clear Classes (None, Both)

OVERFLOW CLASSES OF TAILWIND CSS

This class accepts more than 1 value in Tailwind CSS. It is one of the areas of CSS Overflow. This overflow of control over how the content of the element is treated to the largest of the container. Specify that content must be recorded or added to scroll bars. There is a separate area in CSS for CSS Overflow-x and CSS Overflow-y.

- **overflow-default:** Automatically add a scroll bar whenever needed. This section adds scroll bars to an item where its content fills the boundary of that element.
- **overflow:** It cut and all content is invisible. This section is used to attach any content within an element that exceeds the boundaries of that element.

Here, we have simple example of overflow classes right below:

```
<!DOCTYPE html>

<head>
  <link rel="stylesheet href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
  >
</head>

<body class="text-center">
  <b class="text-green-600 text-5xl font-
bold">Tailwind CSS Overflow Class </b>

  <div class="overflow-auto bg-gray-200 p-15
m-16 h-24 text-justify">
    Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
    Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
    Curabitur non imperdiet nunc, vel fringilla
nunc.
    Donec elementum ultricies velit maximus
suscipit. Duis non diam
    vel mi venenatis mattis a interdum erat.
Cras malesuada ex purus, e
    get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
    roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
    ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
    leo vehicula accumsan. Etiam eleifend velit in
justo tristique
    facilisis
    id. Pellentesque commodo augue elit, at
    sagittis diam convallis in.
```

roin tempor sed nulla a aliquet. Nullam
 mauris nunc, molestie pelle
 ntesque lacus id, facilisis fringilla elit.
 Aenean non massa sed
 leo vehicula accumsan. Etiam eleifend
 velit in justo tristique
 facilisis. Nam et varius risus. Donec
 mattis dignissim diam, eu
 blandit nunc venenatis id. Pellentesque
 commodo augue elit, at
 </div>

<div class="overflow-hidden bg-gray-200 p-15
 m-16 h-24 text-justify">

Sed est tortor, aliquet sit amet rutrum at,
 molestie at ex.

Sed lorem mauris, aliquam ac tempor sed,
 malesuada et ligula.

Curabitur non imperdiet nunc, vel fringilla
 nunc.

Donec elementum ultricies velit maximus
 suscipit. Duis non diam

vel mi venenatis mattis a interdum erat.
 Cras malesuada ex purus, e

get tristique nibh sollicitudin et. Duis
 viverra rutrum tincidunt. P

roin tempor sed nulla a aliquet. Nullam
 mauris nunc, molestie pelle

ntesque lacus id, facilisis fringill/a
 elit. Aenean non massa sed

leo vehicula accumsan. Etiam eleifend
 velit in justo tristique

facilisis. Nam et varius risus. Donec
 mattis dignissim diam, eu

blandit nunc venenatis id. Pellentesque
 commodo augue elit, at

</div>

</body>

</html>

Tailwind CSS Overflow Class

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, eget tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. Proin tempor sed nulla a aliquet. Nullam

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, eget tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. Proin tempor sed nulla a aliquet. Nullam

Overflow Classes

- overflow-visible
- overflow-scroll
- overflow-x-auto
- overflow-y-auto
- Overflow-x: It specifies whether to add a scroll bar, clip the content, or display overflow content of a block-level element when it overflows at the left and right edges.
- Overflow-x-auto: It provides a scrolling mechanism for overflowing boxes.

Here we have simple example of overflow classes right below:

```
<!DOCTYPE html>

<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
      rel="stylesheet">
</head>

<body class="text-center">
  <b class="text-green-600 text-5xl font-
  bold">Tailwind CSS Overflow Class </b>
```

```

<div class="overflow-x-visible bg-gray-200
p-15 m-16 h-24 text-justify">
  <p>Overflow X visible</p>
  Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
  Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
  Curabitur non imperdiet nunc, vel fringilla
nunc.
  Donec elementum ultricies velit maximus
suscipit. Duis non diam
  vel mi venenatis mattis a interdum erat.
Cras malesuada ex purus, e
  get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
  roin tempor sed nulla a aliquet. Nullam
mauris nunc, molestie pelle
  ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
  leo vehicula accumsan. Etiam eleifend
velit in justo tristique
  facilisis. Nam et varius risus. Donec
mattis dignissim diam, eu
  blandit nunc venenatis id. Pellentesque
commodo augue elit, at
</div>
<br>
<div class="overflow-x-hidden bg-gray-200 p-15
m-16 h-24 text-justify">
  <p>Overflow X hidden</p>
  Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
  Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
  Curabitur non imperdiet nunc, vel fringilla
nunc.
  Donec elementum ultricies velit maximus
suscipit. Duis non diam
  vel mi venenatis mattis a interdum erat.
Cras malesuada ex purus, e
  get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
  roin tempor sed nulla a aliquet. Nullam
mauris nunc, molestie pelle

```

```

        ntesque lacus id, facilisis fringill/a
elit. Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend
velit in justo tristique
        facilisis. Nam et varius risus. Donec
mattis dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
nibh sollicitudin et. Duis viverra rutrum tincidunt. P
        roin tempor sed nulla a aliquet. Nullam mauris
nunc, molestie pelle
        ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend velit in
justo tristique
        facilisis. Nam et varius risus. Donec mattis
dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
        sagittis diam convallis in.
    </div>
</body>
</html>

```

Tailwind CSS Overflow Class

Overflow X visible

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at

Overflow X hidden

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e

Overflow Classes (x)

- overflow-x-hidden
- overflow-y-hidden
- overflow-x-visible
- overflow-y-visible
- overflow-x-scroll
- overflow-y-scroll

Here, we have simple example of overflow classes right below:

```
<!DOCTYPE html>

<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
      rel="stylesheet">
</head>

<body class="text-center">
  <b class="text-green-600 text-5xl font-
  bold">Tailwind CSS Overflow Class </b>
  <div class="overflow-y-visible bg-gray-200
  p-15 m-16 h-24 text-justify">
    <p>Overflow Y visible</p>
    Sed est tortor, aliquet sit amet rutrum at,
    molestie at ex.
    Sed lorem mauris, aliquam ac tempor sed,
    malesuada et ligula.
    Curabitur non imperdiet nunc, vel fringilla
    nunc.
    Donec elementum ultricies velit maximus
    suscipit. Duis non diam
    vel mi venenatis mattis a interdum erat.
    Cras malesuada ex purus, e
    get tristique nibh sollicitudin et. Duis
    viverra rutrum tincidunt. P
    roin tempor sed nulla a aliquet. Nullam
    mauris nunc, molestie pelle
```

```

        ntesque lacus id, facilisis fringilla elit.
Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend
velit in justo tristique
        facilisis. Nam et varius risus. Donec
mattis dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
        </div>
<br>
        <div class="overflow-y-hidden bg-gray-200 p-15
m-16 h-24 text-justify">
        <p>Overflow Y hidden</p>
        Sed est tortor, aliquet sit amet rutrum at,
molestie at ex.
        Sed lorem mauris, aliquam ac tempor sed,
malesuada et ligula.
        Curabitur non imperdiet nunc, vel fringilla
nunc.
        Donec elementum ultricies velit maximus
suscipit. Duis non diam
        vel mi venenatis mattis a interdum erat.
Cras malesuada ex purus, e
        get tristique nibh sollicitudin et. Duis
viverra rutrum tincidunt. P
        roin tempor sed nulla a aliquet. Nullam
mauris nunc, molestie pelle
        ntesque lacus id, facilisis fringill/a
elit. Aenean non massa sed
        leo vehicula accumsan. Etiam eleifend
velit in justo tristique
        facilisis. Nam et varius risus. Donec
mattis dignissim diam, eu
        blandit nunc venenatis id. Pellentesque
commodo augue elit, at
        </div>
</body>

</html>

```

Tailwind CSS Overflow Class

Overflow Y visible

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get tristique nibh sollicitudin et. Duis viverra rutrum tincidunt. P roin tempor sed nulla a aliquet. Nullam mauris nunc, molestie pelle ntesque lacus id, facilisis fringilla elit. Aenean non massa sed leo vehicula accumsan. Etiam eleifend velit in justo tristique facilisis. Nam et varius risus. Donec mattis dignissim diam, eu blandit nunc venenatis id. Pellentesque commodo augue elit, at

Overflow Y hidden

Sed est tortor, aliquet sit amet rutrum at, molestie at ex. Sed lorem mauris, aliquam ac tempor sed, malesuada et ligula. Curabitur non imperdiet nunc, vel fringilla nunc. Donec elementum ultricies velit maximus suscipit. Duis non diam vel mi venenatis mattis a interdum erat. Cras malesuada ex purus, e get

Overflow Classes (y)

TAILWIND CSS FLEX DIRECTION

The CSS flexbox is a feature to develop the frontend, there are four directions available in tailwind CSS, and all the properties are covered. It is the alternative of flex-direction of CSS Property for quick development of frontend. To use the flex-direction, you have to add the flex class in your element before the flex-direction class.

Flex Direction:

- flex-row
- flex-row-reverse
- flex-col
- flex-col-reverse

```
<!DOCTYPE html>
```

```
<head>
```

```
  <title>Tailwind flex-row Class</title>
```

```

<link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"

```

```

    rel="stylesheet">

```

```

</head>

```

```

<body class="text-center">

```

```

    <h1 class="text-green-600 text-5xl font-bold">

```

```

    </h1>

```

```

    <h1 class="text-green-600 text-3xl font-bold">

```

```

Tailwind Flex - Row </h1>

```

```

    <div id="main" class="flex flex-row
justify-evenly">

```

```

        <div class="bg-green-900 w-24 h-12"> 1 </div>

```

```

        <div class="bg-green-800 w-24 h-12"> 2 </div>

```

```

        <div class="bg-green-700 w-24 h-12"> 3 </div>

```

```

        <div class="bg-green-600 w-24 h-12" > 4 </div>

```

```

        <div class="bg-green-500 w-24 h-12"> 5 </div>

```

```

        <div class="bg-green-400 w-24 h-12"> 6 </div>

```

```

    </div>

```

```

    <br>

```

```

    <h1 class="text-green-600 text-3xl font-bold">

```

```

Tailwind Flex - Row Reverse </h1>

```

```

    <div id="main" class="flex flex-row-reverse
justify-evenly">

```

```

        <div class="bg-green-900 w-24 h-12"> 1 </div>

```

```

        <div class="bg-green-800 w-24 h-12"> 2 </div>

```

```

        <div class="bg-green-700 w-24 h-12"> 3 </div>

```

```

        <div class="bg-green-600 w-24 h-12"> 4 </div>

```

```

        <div class="bg-green-500 w-24 h-12"> 5 </div>

```

```

        <div class="bg-green-400 w-24 h-12"> 6 </div>

```

```

    </div>

```

```

<br>

```

```

<h1 class="text-green-600 text-3xl font-bold">

```

```

Tailwind Flex - Column </h1>

```

```

    <div id="main" class="flex flex-col justify-evenly">

```

```

        <div class="bg-green-900 w-24 h-12"> 1 </div>

```

```

    <div class="bg-green-800 w-24 h-12"> 2 </div>
    <div class="bg-green-700 w-24 h-12"> 3 </div>
    <div class="bg-green-600 w-24 h-12"> 4 </div>
    <div class="bg-green-500 w-24 h-12"> 5 </div>
    <div class="bg-green-400 w-24 h-12"> 6 </div>
  </div>
  <br>

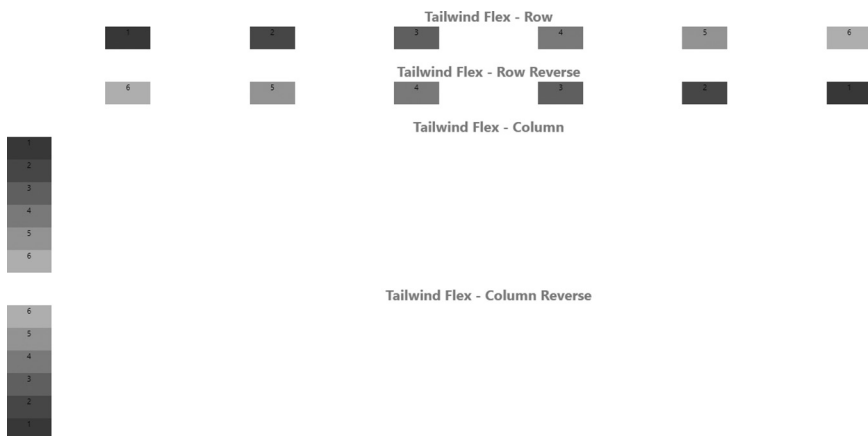
  <h1 class="text-green-600 text-3xl font-bold">
Tailwind Flex - Column Reverse </h1>

<div id="main" class="flex flex-col-reverse
justify-evenly">
  <div class="bg-green-900 w-24 h-12"> 1 </div>
  <div class="bg-green-800 w-24 h-12"> 2 </div>
  <div class="bg-green-700 w-24 h-12"> 3 </div>
  <div class="bg-green-600 w-24 h-12"> 4 </div>
  <div class="bg-green-500 w-24 h-12"> 5 </div>
  <div class="bg-green-400 w-24 h-12"> 6 </div>
</div>

</body>

</html>

```



Flex (Flex-Direction)

TAILWIND CSS FLEX

The CSS flexbox is an important feature of frontend enhancement, rather than the flex found in tailwind CSS, and all layouts are covered in a classy way. It is one of the CSS flex Property for fast frontend development. It is used to set the length of the variable. The flex class is very responsive, easy to use, and mobile-friendly. It is easy to positioning child elements and a main container. Margins do not wrap in content margins. The layout of any element can be easily changed without editing the HTML section.

FLEX

flex-1: Specific measure, how much material will grow compared to other variables. It is used to allow a flexible object to grow and shrink as needed, ignoring its original size.

Syntax:

```
<div class="flex-1"> Write your content here </div>
```

flex-initial: This section describes how the object will grow compared to other variables. It is used to allow a flexible object to shrink but not to grow, considering its original size

Syntax:

```
<div class="flex-initial"> Write your content here </div>
```

flex-auto: This section specifies how much the object will grow compared to the content of the flexible material. It is used to allow a flexible object to grow and shrink as needed, ignoring its original size.

Syntax:

```
<div class="flex-auto"> Write your content here </div>
```

flex-none: This category limits the growth or reduction of a feature if there is more space or not. This is a protection against being flexible or degraded, it only takes up space depending on the size of the content.

Syntax:

```
<div class="flex-none"> Write your content here </div>
```

Here is the complete example of flex:

```
<!DOCTYPE html>
<html>

<head>
  <title> Tailwind flex-1 Class </title>

  <link href=
"https://unpkg.com/tailwindcss@1.0/dist/tailwind.min.css"
    rel="stylesheet">
</head>

<body>
  <h1 class="text-center text-green-600 text-5xl
font-bold">
    Tailwind CSS flex-1 Class
  </h1>
  <div id="main" class="bg-green-200 border-4
border-green-600 w-2/3 ml-32">
    <p class="ml-2">
      This is the effect of Flex-1:
    </p>
    <div class="flex m-2 text-white">
      <div class="bg-green-800 flex-1">
        Flex-1
      </div>
    </div>

    <p class="ml-2">
      This is the effect of Flex-initial:
    </p>
    <div class="flex m-2 text-white">
      <div class="bg-green-900 flex-initial">
        Flex-initial
      </div>
    </div>
    <p class="ml-2">
      This is the effect of flex-auto:
    </p>

    <div class="flex m-2 text-white">
      <div class="bg-green-800 flex-auto">
```

```

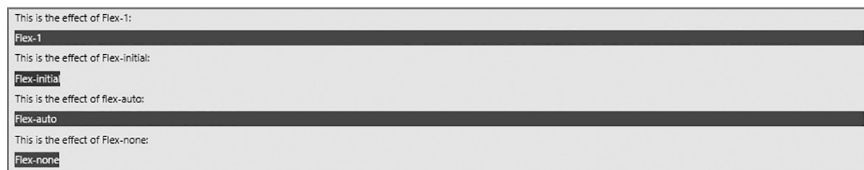
        Flex-auto
      </div>
    </div>

    <p class="ml-2">
      This is the effect of Flex-none:
    </p>
    <div class="flex m-2 text-white">
      <div class="bg-green-800 flex-none">
        Flex-none
      </div>
    </div>
  </div>
</body>

</html>

```

Tailwind CSS flex-1 Class



Flex (Flex-1)

TAILWIND CSS ORDER

It is one of the best features of tailwind by using class, and we can order the flex and grid items according to our needs. There are various order class. It is used to render flex and grid items in a different order than they appear in the DOM:

- order-1
- order-2
- order-3
- order-4
- order-5
- order-6
- order-7

- order-8
- order-9
- order-10
- order-11
- order-12
- order-first
- order-last
- order-none

Parameter: It accepts two types of parameters, but one at a time, can only follow the order number or the position you will mention.

- number: Integer number to follow the order index.
- string: In word order index position, only accept last and first.

```
<!DOCTYPE html>
<html>

<head>
  <title> Tailwind order Class </title>

  <link href=
"https://unpkg.com/tailwindcss@1.0/dist/tailwind.min.css"
    rel="stylesheet">
</head>

<body class="text-center">

  <b class="text-green-600 text-5xl font-
bold">Tailwind CSS order Class </b>

  <div id="main" class="flex flex-row
justify-evenly">
    <div class="bg-green-900 order-last w-24
h-12"> 1 </div>
    <div class="bg-green-800 w-24 h-12"> 2 </div>
    <div class="bg-green-700 w-24 h-12"> 3 </div>
```

```

        <div class="bg-green-600 w-24 h-12"> 4 </div>
        <div class="bg-green-500 order-first w-24
h-12"> 5 </div>
        <div class="bg-green-400 w-24 h-12"> 6 </div>
    </div>
</body>

</html>

```



TAILWIND CSS GRID TEMPLATE COLUMNS

This section accepts more than one value in tailwind CSS for all class-covered layouts used to set number of columns and grid column sizes, and here, we will do the same but for faster frontend upgrades. The number of columns is set to the value of the values assigned to this category.

Grid Template Columns:

- grid-cols-1
- grid-cols-2
- grid-cols-3
- grid-cols-4
- grid-cols-5
- grid-cols-6
- grid-cols-7
- grid-cols-8
- grid-cols-9
- grid-cols-10
- grid-cols-11
- grid-cols-12
- grid-cols-none

Here is the complete example of grid col right below:

```
<!DOCTYPE html>
<html>

<head>
  <title>Tailwind grid-cols Class</title>

  <link      rel="stylesheet" href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.css"
  >
</head>

<body class="text-center">

  <b class="text-green-600 text-5xl font-bold">
Tailwind CSS grid-cols Class

  <div id="main" class="grid grid-cols-2 m-12 gap-1
justify-evenly">
    <div class="bg-green-700 w-26 h-12"> 1 </div>
    <div class="bg-green-500 w-26 h-12"> 2 </div>
    <div class="bg-green-300 w-26 h-12"> 3 </div>
    <div class="bg-green-700 w-26 h-12"> 4 </div>
    <div class="bg-green-500 w-26 h-12"> 5 </div>
    <div class="bg-green-300 w-26 h-12"> 6 </div>
  </div>
  <div id="main" class="grid grid-cols-3 m-6 gap-1
justify-evenly">
    <div class="bg-green-700 w-26 h-12"> 1 </div>
    <div class="bg-green-500 w-26 h-12"> 2 </div>
    <div class="bg-green-300 w-26 h-12"> 3 </div>
    <div class="bg-green-700 w-26 h-12"> 4 </div>
    <div class="bg-green-500 w-26 h-12"> 5 </div>
    <div class="bg-green-300 w-26 h-12"> 6 </div>
  </div>

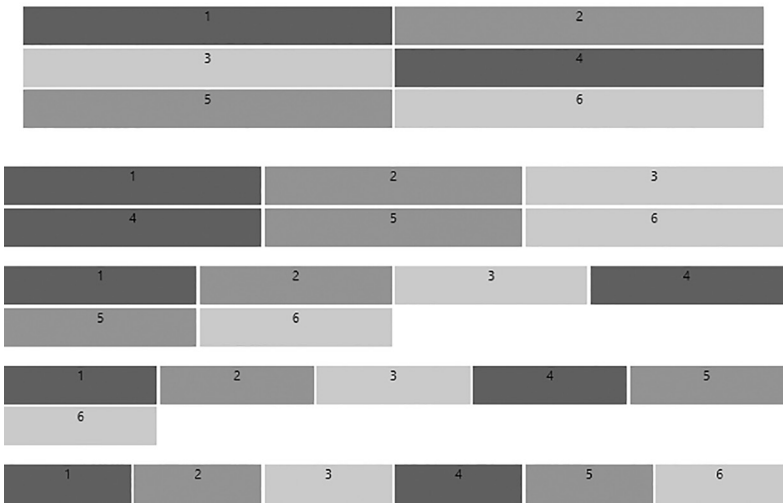
  <div id="main" class="grid grid-cols-4 m-6 gap-1
justify-evenly">
    <div class="bg-green-700 w-26 h-12"> 1 </div>
    <div class="bg-green-500 w-26 h-12"> 2 </div>
    <div class="bg-green-300 w-26 h-12"> 3 </div>
    <div class="bg-green-700 w-26 h-12"> 4 </div>
```

```
<div class="bg-green-500 w-26 h-12"> 5 </div>
<div class="bg-green-300 w-26 h-12"> 6 </div>
</div>

<div id="main" class="grid grid-cols-5 m-6 gap-1
justify-evenly">
  <div class="bg-green-700 w-26 h-12"> 1 </div>
  <div class="bg-green-500 w-26 h-12"> 2 </div>
  <div class="bg-green-300 w-26 h-12"> 3 </div>
  <div class="bg-green-700 w-26 h-12"> 4 </div>
  <div class="bg-green-500 w-26 h-12"> 5 </div>
  <div class="bg-green-300 w-26 h-12"> 6 </div>
</div>
<div id="main" class="grid grid-cols-6 m-6 gap-1
justify-evenly">
  <div class="bg-green-700 w-26 h-12"> 1 </div>
  <div class="bg-green-500 w-26 h-12"> 2 </div>
  <div class="bg-green-300 w-26 h-12"> 3 </div>
  <div class="bg-green-700 w-26 h-12"> 4 </div>
  <div class="bg-green-500 w-26 h-12"> 5 </div>
  <div class="bg-green-300 w-26 h-12"> 6 </div>
</div>
</body>

</html>
```

Tailwind CSS grid-cols Class



Grid Cols

TAILWIND CSS ALIGNMENT

Justify Content

This class accepts two values in tailwind CSS. It is the same to the CSS justify-content property. It is used to describe the alignment of the flexible box container. It contains the space between and around content items along the main axis of a flex container. It is used for controlling how flex and grid items are positioned along a container's main axis.

Justify Content classes:

- justify-start: It is used to align-flex items from the start of the container
- justify-end
- justify-center
- justify-between
- justify-around
- justify-evenly

```
<!DOCTYPE html>
<head>
  <link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
      rel="stylesheet">
</head>

<body class="text-center m-12">

  <b class="text-green-600 text-5xl font-bold">
Tailwind CSS Justify Content Class
  <h1 class="text-black-100 text-2xl text-left font-
bold"> Justify Start </h1>

  <div id="main" class="flex justify-start
flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>
    <div class="bg-green-500 w-24 h-12"> 3 </div>
```

```

    <div class="bg-green-400 w-24 h-12"> 4 </div>
  </div>

```

```

  <h1 class="text-black-100 text-2xl text-right
font-bold"> Justify End </h1>
  <div id="main" class="flex justify-end flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>
    <div class="bg-green-500 w-24 h-12"> 3 </div>
    <div class="bg-green-400 w-24 h-12"> 4 </div>
  </div>

```

```

  <h1 class="text-black-100 text-2xl text-center font-
bold" > Justify Center </h1>
  <div id="main" class="flex justify-center flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>
    <div class="bg-green-500 w-24 h-12"> 3 </div>
    <div class="bg-green-400 w-24 h-12"> 4 </div>
  </div>

```

```

  <br>
  <h1 class="text-black-100 text-2xl text-center font-
bold"> Justify Around </h1>
  <div id="main" class="flex justify-around flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>
    <div class="bg-green-500 w-24 h-12"> 3 </div>
    <div class="bg-green-400 w-24 h-12"> 4 </div>
  </div>

```

```

  <br>
  <h1 class="text-black-100 text-2xl text-center font-
bold"> Justify Between </h1>
  <div id="main" class="flex justify-between flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>
    <div class="bg-green-500 w-24 h-12"> 3 </div>
    <div class="bg-green-400 w-24 h-12"> 4 </div>
  </div>

```

```

  <br>
  <h1 class="text-black-100 text-2xl text-center font-
bold"> Justify Evenly </h1>
  <div id="main" class="flex justify-evenly flex-row">
    <div class="bg-green-700 w-24 h-12"> 1 </div>
    <div class="bg-green-600 w-24 h-12"> 2 </div>

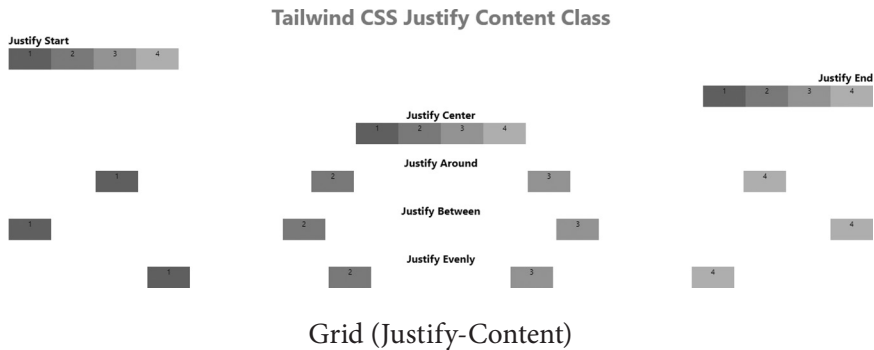
```

```

<div class="bg-green-500 w-24 h-12"> 3 </div>
<div class="bg-green-400 w-24 h-12"> 4 </div>
</div>

</html>

```



TAILWIND CSS JUSTIFY ITEMS

It accepts two values in tailwind CSS. It is the same to the CSS justify-items property. The class is used for controlling how grid items are aligned along with their inline axis.

Justify Items:

- justify-items-auto: Used to justify grid items automatically on their inline axis.
- justify-items-start: Used to justify grid items against the start of their inline axis.
- justify-items-end: Used to justify grid items against the end of their inline axis.
- justify-items-center: Used to justify grid items along their inline axis.
- justify-items-stretch: Used to stretch items along their inline axis.

```

<!DOCTYPE html>
<head>
  <link href=
    "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
    css"
    rel="stylesheet">
</head>

```

```

<body class="text-center m-24">
  <h1 class="text-green-600 text-5xl font-bold">
    Tailwind CSS Justify Items Class    </h1>

    <h1 class="text-black-100 text-2xl text-center
font-bold">Justify Item Auto</h1>
    <div id="main" class="grid justify-items-auto
grid-cols-3">
      <div class="bg-green-400 rounded-lg m-4 h-12">
1 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
2 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
3 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
4 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
5 </div>
      <div class="bg-green-00 rounded-lg m-4 h-12">
6 </div>
    </div>

    <h1 class="text-black-100 text-2xl text-center
font-bold">Justify Item Start</h1>
    <div id="main" class="grid justify-items-start
grid-cols-3">
      <div class="bg-green-600 rounded-lg m-4 h-12">
1 </div>
      <div class="bg-green-600 rounded-lg m-4 h-12">
2 </div>
      <div class="bg-green-600 rounded-lg m-4 h-12">
3 </div>
      <div class="bg-green-600 rounded-lg m-4 h-12">
4 </div>
      <div class="bg-green-600 rounded-lg m-4 h-12">
5 </div>
      <div class="bg-green-600 rounded-lg m-4 h-12">
6 </div>
    </div>

    <h1 class="text-black-100 text-2xl text-center
font-bold">Justify Item End</h1>

```

```

    <div id="main" class="grid justify-items-end
grid-cols-3">
      <div class="bg-green-400 rounded-lg m-4 h-12">
1 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
2 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
3 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
4 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
5 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
6 </div>
    </div>

```

```

    <h1 class="text-black-100 text-2xl text-center
font-bold">Justify Item Center</h1>
    <div id="main" class="grid justify-items-center
grid-cols-3">
      <div class="bg-green-400 rounded-lg m-4 h-12">
1 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
2 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
3 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
4 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
5 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
6 </div>
    </div>

```

```

    <h1 class="text-black-100 text-2xl text-center
font-bold">Justify Item Strtech</h1>
    <div id="main" class="grid justify-items-stretch
grid-cols-3">
      <div class="bg-green-400 rounded-lg m-4 h-12">
1 </div>
      <div class="bg-green-400 rounded-lg m-4 h-12">
2 </div>

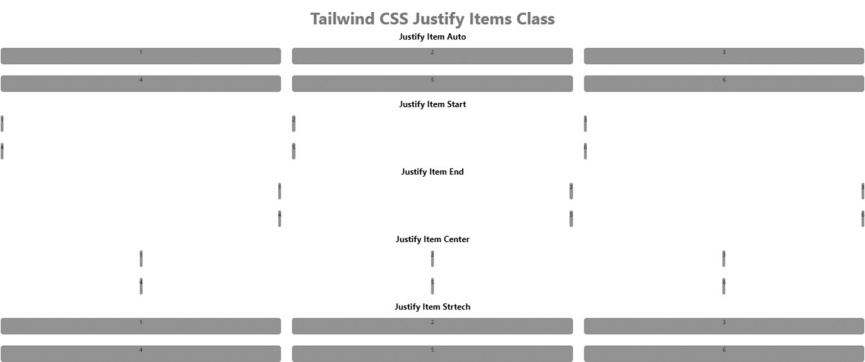
```

```

    <div class="bg-green-400 rounded-lg m-4 h-12">
3 </div>
    <div class="bg-green-400 rounded-lg m-4 h-12">
4 </div>
    <div class="bg-green-400 rounded-lg m-4 h-12">
5 </div>
    <div class="bg-green-400 rounded-lg m-4 h-12">
6 </div>
    </div>

</body>

</html>
```



TAILWIND CSS SPACING

Padding

It accepts lots of values in tailwind in which all the properties are covered in class form. It is the same Padding Property in CSS. It is used to create space around the element, inside any defined border. We can set different paddings for individual sides (top, right, bottom, left). It is important to add border properties to implement padding properties. There are lots of CSS properties used for padding like CSS padding-top, CSS padding-bottom, CSS padding-right, and CSS padding-left.

Padding Classes:

- `p-0`: Used to define the padding on all the sides.
- `py-0`: Used to define padding on the y-axis, i.e., `padding-top` and `padding-bottom`.
- `px-0`: Used to define padding on the x-axis, i.e., `padding-left` and `padding-right`.
- `pt-0`: Used to add padding on top.
- `pr-0`: Used to add padding on right.
- `pb-0`: Used to add padding on the bottom.
- `pl-0`: Used to add padding on left.

You can change the number “0” with the valid “rem” values.

```
<!DOCTYPE html>
<head>
  <link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
      rel="stylesheet">
</head>

<body class="text-center">
  <h1 class="text-green-600 text-5xl font-bold">
    Tailwind CSS Padding Class    </h1>
  <br>
  <p class="text-black-400 text-4xl font-bold"> P
Class of Padding </p>
  <div class="flex justify-center">
    <div class="p-4 bg-green-300 w-24 h-24">
      <div class="border-2 border-green-800
        bg-green-600 w-16 h-16">

    </div>
  </div>
</div>
```

```

<br>
<p class="text-black-400 text-4xl font-bold"> Py
Class of Padding </p>
  <div class="flex justify-center">
    <div class="py-4 bg-green-300 w-24 h-24">
      <div class="border-2 border-green-800
        bg-green-600 w-16 h-16">

    </div>
  </div>
</div>

```

```

<br>
<p class="text-black-400 text-4xl font-bold"> Px
Class of Padding </p>
  <div class="flex justify-center">
    <div class="px-4 bg-green-300 w-24 h-24">
      <div class="border-2 border-green-800
        bg-green-600 w-16 h-16">

    </div>
  </div>
</div>

```

```

<br>
<p class="text-black-400 text-4xl font-bold"> Pt
Class of Padding </p>
  <div class="flex justify-center">
    <div class="pt-4 bg-green-300 w-24 h-24">
      <div class="border-2 border-green-800
        bg-green-600 w-16 h-16">

    </div>
  </div>
</div>

```

```

<br>

```

```

    <p class="text-black-400 text-4xl font-bold"> Pr
Class of Padding </p>
    <div class="flex justify-center">
    <div class="pr-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold"> Pb
Class of Padding </p>
    <div class="flex justify-center">
    <div class="pb-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold"> Pl
Class of Padding </p>
    <div class="flex justify-center">
    <div class="pl-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>
</body>

</html>

```

Tailwind CSS Padding Class

P Class of Padding



Py Class of Padding



Px Class of Padding



Pt Class of Padding



Pr Class of Padding



Pb Class of Padding



Pl Class of Padding



Padding Classes (p)

MARGIN

It is the same as CSS Margin Property that is used to create space around the element, outside any defined border. We can set different margins for individual sides (top, right, bottom, left). It is good to add border properties to implement margin classes. There are various CSS properties used for margin like CSS padding-top, CSS padding-bottom, CSS padding-right, and CSS padding-left.

Margin Classes:

- `m-0`: Used to define the negative margin on all the sides.
- `-m-0`: Used to define the negative margin on all the sides.
- `my-0`: Used to define margin on the y-axis, i.e., `margin-top` and `margin-bottom`.
- `-my-0`: Used to define negative margin on the y-axis, i.e., `margin-top` and `margin-bottom`.
- `mx-0`: Used to define margin on the x-axis, i.e., `margin-left` and `margin-right`.
- `-mx-0`: Used to add a negative margin on right.
- `mt-0`: Used to add a margin on top.
- `-mt-0`: Used to add a negative margin on top.
- `mr-0`: Used to add a margin on right.
- `-mr-0`: Used to add a negative margin on right.
- `mb-0`: Used to add a margin on the bottom.
- `-mb-0`: Used to add a negative margin on the bottom.
- `ml-0`: Used to add a margin on left.
- `-ml-0`: Used to add a negative margin on left.

Example:

```
<!DOCTYPE html>
<head>
  <link href=
"https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
      rel="stylesheet">
  <style>
    .margin{
      height: 200px;
      width:200px;
```

```

        background-color: antiquewhite;
    }
</style>
</head>

<body class=" container text-center">
    <h1 class="text-green-600 text-5xl font-bold">
        Tailwind CSS Padding Class    </h1>
    <br>
    <p class="text-black-400 text-4xl font-bold">M
Class of Padding </p>
    <div class="flex justify-center margin">
    <div class="m-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold">-M
Class of Padding </p>
    <div class=" flex justify-center margin">
    <div class=" -m-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold">My
Class of Padding </p>
    <div class="flex justify-center margin">
    <div class="my-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>
</div>

```

```

    <br>
    <p class="text-black-400 text-4xl font-bold">Ml
Class of Padding </p>
    <div class="flex justify-center margin">
    <div class="ml-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold">Mt
Class of Padding </p>
    <div class="flex justify-center margin">
    <div class="mt-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>
    <p class="text-black-400 text-4xl font-bold">Mr
Class of Padding </p>
    <div class="flex justify-center margin">
    <div class="mr-4 bg-green-300 w-24 h-24">
        <div class="border-2 border-green-800
            bg-green-600 w-16 h-16">

            </div>
        </div>
    </div>

    <br>

</body>

</html>

```

Tailwind CSS Padding Class



M Class of Padding



-M Class of Padding



My Class of Padding



MI Class of Padding



Mt Class of Padding



Mr Class of Padding

Margin Classes

SPACE BETWEEN

It is the same as CSS Space Between property. It is used for controlling the space between child elements.

Space Between classes:

- space-y-0
- space-y-reverse

- -space-y-0
- space-x-0
- space-x-reverse
- -space-x-0

Here is the complete example of spacing right below:

```
<!DOCTYPE html>
<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
      rel="stylesheet">
</head>

<body class="text-center">
  <h class="text-green-600 text-5xl font-bold">

    Tailwind CSS Space Between Class
  </h>

  <div class="m-24 space-y-4">
    <b class="text-green-600 text-5xl font-
    bold">Space y </b>

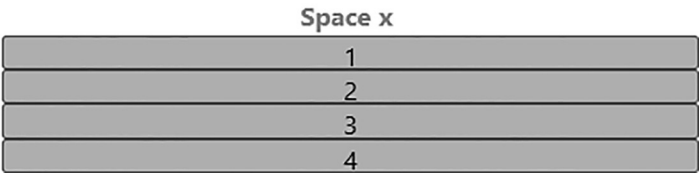
      <div class="bg-green-400 h-16 text-5xl
    rounded-lg
      border-2 border-green-800">1</div>
      <div class="bg-green-400 h-16 text-5xl
    rounded-lg
      border-2 border-green-800">2</div>
      <div class="bg-green-400 text-5xl h-16
    rounded-lg
      border-2 border-green-800">3</div>
      <div class="bg-green-400 h-16 text-5xl
    rounded-lg
      border-2 border-green-800">4</div>
    </div>

  <br>

  <div class="m-24 space-x-4">
```

```
<b class="text-green-600 text-5xl font-  
bold">Space x </b>  
  <div class="bg-green-400 h-16 text-5xl  
rounded-lg  
      border-2 border-green-800">1</div>  
  <div class="bg-green-400 h-16 text-5xl  
rounded-lg  
      border-2 border-green-800">2</div>  
  <div class="bg-green-400 h-16 text-5xl  
rounded-lg  
      border-2 border-green-800">3</div>  
  <div class="bg-green-400 h-16 text-5xl  
rounded-lg  
      border-2 border-green-800">4</div>  
</div>  
</body>  
  
</html>
```

Tailwind CSS Space Between Class



Space Between

TAILWIND CSS BORDERS

Border Radius

It can accept more than one value in tailwind. It is the same as of the CSS border-radius property. This class is used to set the border radius. Here, classes are covered only used to create rounded corners like rounded-sm, rounded-md, and rounded-lg, but not the fully circular or the pill shapes.

Border-Radius Classes:

- rounded-none
- rounded-sm
- rounded
- rounded-md
- rounded-lg
- rounded-xl
- rounded-2xl
- rounded-3xl
- rounded-full
- rounded-t-none
- rounded-r-none
- rounded-b-none
- rounded-l-none
- rounded-t-sm
- rounded-r-sm
- rounded-b-sm
- rounded-l-sm
- rounded-t

- rounded-r
- rounded-b
- rounded-l

Here is the complete example of above border radius right below:

```
<!DOCTYPE html>
<html>

<head>
  <link rel="stylesheet" href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
  >
</head>

<body class="text-center">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind Border Radius Class    </h1>

    <div class="m-5 grid grid-cols-3 gap-2
bg-green-200 p-2">

      <!-- First sub divison is not for roundig-->
      <div class="rounded-none bg-green-400 w-36
h-12">
        rounded-none</div>
      <div class="rounded bg-orange-400 w-36 h-12">
        rounded</div>
      <div class="rounded-sm bg-yellow-400 w-36
h-12">
        rounded-sm</div>
      <div class="rounded-md bg-red-400 w-36 h-12">
        rounded-md</div>
      <div class="rounded-lg bg-purple-400 w-36
h-12">
        rounded-lg</div>
      <div class="rounded-xl bg-gray-400 w-36 h-12">
        rounded-xl</div>
      <div class="rounded-t bg-pink-400 w-36 h-12">
```

```

        rounded-t</div>
<div class="rounded-r bg-blue-400 w-36 h-12">
    rounded-r</div>
<div class="rounded-l bg-green-400 w-36 h-12">
    rounded-l</div>
<div class="rounded-b bg-yellow-400 w-36
h-12">
    rounded-b</div>
</div>
</body>

</html>

```

Tailwind Border Radius Class

rounded-none	rounded	rounded-sm
rounded-md	rounded-lg	rounded-xl
rounded-t	rounded-r	rounded-l
rounded-b		

Border Radius

- rounded-t-md
- rounded-r-md
- rounded-b-md
- rounded-l-md
- rounded-t-lg
- rounded-r-lg
- rounded-b-lg
- rounded-l-lg
- rounded-t-xl
- rounded-r-xl
- rounded-b-xl
- rounded-l-xl

- rounded-t-2xl
- rounded-r-2xl
- rounded-b-2xl
- rounded-l-2xl
- rounded-t-3xl
- rounded-r-3xl
- rounded-b-3xl
- rounded-l-3xl
- rounded-t-full
- rounded-r-full
- rounded-b-full
- rounded-l-full
- rounded-tl-none
- rounded-tr-none
- rounded-br-none
- rounded-bl-none
- rounded-tl-sm
- rounded-tr-sm
- rounded-br-sm
- rounded-bl-sm
- rounded-tl
- rounded-tr
- rounded-br
- rounded-bl

- rounded-tl-md
- rounded-tr-md
- rounded-br-md
- rounded-bl-md
- rounded-tl-lg
- rounded-tr-lg
- rounded-br-lg
- rounded-bl-lg
- rounded-tl-xl
- rounded-tr-xl
- rounded-br-xl
- rounded-bl-xl
- rounded-tl-2xl
- rounded-tr-2xl
- rounded-br-2xl
- rounded-bl-2xl
- rounded-tl-3xl
- rounded-tr-3xl
- rounded-br-3xl
- rounded-bl-3xl
- rounded-tl-full
- rounded-tr-full
- rounded-br-full
- rounded-bl-full

Here is some example of rounded shape border right below:

```
<!DOCTYPE html>
<html>

<head>
  <link rel="stylesheet" href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
  >
</head>

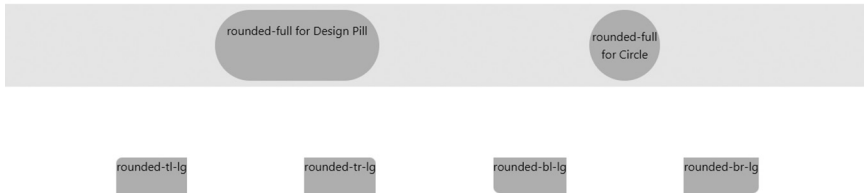
<body class="text-center">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind CSS Border Radius Class    </h1>

    <div class="flex gap-2 m-24 bg-green-200 justify-
    evenly p-2">
      <div class="rounded-full bg-green-400 py-4
      px-4" >
        rounded-full for Design Pill
      </div>
      <div class="rounded-full mx-14 bg-green-400
      h-24 w-24
        flex items-center">
        rounded-full for Circle
      </div>
    </div>
    <!-- For Large -->
    <div class="flex gap-2 m-24 justify-evenly">
      <div class="rounded-tl-lg bg-green-400 h-12">
        rounded-tl-lg</div>
      <div class="rounded-tr-lg bg-green-400 h-12">
        rounded-tr-lg</div>
      <div class="rounded-bl-lg bg-green-400 h-12">
        rounded-bl-lg</div>
      <div class="rounded-br-lg bg-green-400 h-12">
        rounded-br-lg</div>
    </div>

</body>

</html>
```

Tailwind CSS Border Radius Class



Rounded Border

BORDER WIDTH

It is used to set the border width of all four sides of an element. The border-width property is the collection of four property.

Border Width and its Classes are as follows:

- border
- border-0
- border-2
- border-4
- border-8
- border-r
- border-r-0
- border-r-2
- border-r-4
- border-r-8
- border-t
- border-t-0
- border-t-2
- border-t-4
- border-t-8
- border-b
- border-b-0

- border-b-2
- border-b-4
- border-b-8
- border-l
- border-l-0
- border-l-2
- border-l-4
- border-l-8

All sides been used to set the border width for all sides of an element. The classes are given such as borders: border-0, border-2, border-4, and border-8.

Example:

```
<!DOCTYPE html>
<html>

<head>
  <link rel="stylesheet" href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
  >
</head>

<body class="text-center">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind Border Width Class    </h1>

  <div class="my-32 flex gap-4 bg-green-200 p-2">

    <!-- First sub div is not for roundig-->
    <div class="border-l border-green-900
              bg-green-400 w-full h-12">
      border-0
    </div>

    <div class="border border-green-900
              bg-green-400 w-full h-12">
```

```

        border
    </div>

    <div class="border-2 border-green-900
        bg-green-200 w-full h-12">
        border-2
    </div>

    <div class="border-4 border-green-900
        bg-green-200 w-full h-12">
        border-4
    </div>

    <div class="border-8 border-green-900
        bg-green-200 w-full h-12">
        border-8
    </div>
</div>

<div class="mx-4 grid grid-cols-4 gap-2
bg-green-200 p-2">

    <!-- First sub division is not for roundig-->
    <div class="border-b-2 border-green-900
        bg-green-400 w-full h-12">
        border-b-2
    </div>

    <div class="border-r-2 border-green-900
        bg-green-400 w-full h-12">
        border-r-2
    </div>

    <div class="border-l-2 border-green-900
        bg-green-400 w-full h-12">
        border-l-2
    </div>

    <div class="border-t-2 border-green-900
        bg-green-400 w-full h-12">
        border-t-2
    </div>
</div>
</body>

</html>

```

Tailwind Border Width Class



Border Width

BORDER COLOR

This class can accept more than one value in tailwind in. It is the same as CSS border color property. This class is used to specify the border color of an element.

Border Color classes:

- `border-transparent`: The border color will be transparent.
- `border-current`: The border will color depend on the parent element color.
- `border-black`: The border color will be black.
- `border-white`: The border color will be white.
- `border-gray-50`: The border color will be gray.
- `border-red-50`: The border color will be red.
- `border-blue-50`: The border color will be blue.
- `border-indigo-50`: The border color will be indigo.
- `border-purple-50`: The border color will be purple.
- `border-green-50`: The border color will be green.
- `border-yellow-50`: The border color will be yellow.
- `border-pink-50`: The border color will be pink.

```

<!DOCTYPE html>
<head>
  <link rel="stylesheet" href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
  >
</head>

<body class="text-center mx-4">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind Border Color Class
  </h1>
  <div class="mx-14 bg-green-200 grid grid-rows-4
    grid-flow-col gap-2 text-justify p-2">

    <p class="border-2 border-transparent p-2">
      border-transparent p-2
    </p>

    <p class="border-2 border-current text-red-600
p-2">
      text-red-600    </p>

    <p class="border-2 border-white p-2">
      border-white    </p>

    <p class="border-2 border-black p-2">
      border-black    </p>

    <p class="border-2 border-gray-700 p-2">
      border-gray-700  </p>

    <p class="border-2 border-red-700 p-2">
      border-red-700   </p>

    <p class="border-2 border-indigo-700 p-2">
      border-indigo-700 </p>

    <p class="border-2 border-blue-700 p-2">
      border-blue-700  </p>

```

```
<p class="border-2 border-green-700 p-2">
  border-green-700    </p>

<p class="border-2 border-yellow-700 p-2">
  border-yellow-700    </p>

<p class="border-2 border-pink-700 p-2">
  border-pink-700      </p>

<p class="border-2 border-purple-700 p-2">
  border-purple-700    </p>

</div>
</body>

</html>
```

Tailwind Border Color Class		
border-transparent p-2	border-gray-700	border-green-700
text-red-600	border-red-700	border-yellow-700
border-white	border-indigo-700	border-pink-700
border-black	border-blue-700	border-purple-700

Border Color

BORDER OPACITY

The border-opacity is the class of an element that defines as the transparency of the element. It is same as CSS Opacity/Transparency.

Border Opacity class:

- border-opacity-0
- border-opacity-5
- border-opacity-10
- border-opacity-20
- border-opacity-25
- border-opacity-30
- border-opacity-40

- border-opacity-50
- border-opacity-60
- border-opacity-70
- border-opacity-75
- border-opacity-80
- border-opacity-90
- border-opacity-95
- border-opacity-100

Example:

```

<!DOCTYPE html>

<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
      rel="stylesheet">
</head>

<body class="text-center mx-4 space-y-2">
  <h1 class="text-green-600 text-5xl font-bold">
    GeeksforGeeks
  </h1>

  Tailwind CSS Border Opacity Class

  <div class="mx-14 bg-green-200 grid grid-rows-4
    grid-flow-col gap-2 text-justify p-4">

    <p class="border-green-800 border-4
      border-opacity-100 p-2">
      border-opacity-100
    </p>

    <p class="border-green-800 border-4
      border-opacity-75 p-2">
      border-opacity-75    </p>

```

```

    <p class="border-green-800 border-4
        border-opacity-50 p-2">
        border-opacity-50          </p>

    <p class="border-green-800 border-4
        border-opacity-25 p-2">
        border-opacity-25          </p>

    <p class="border-yellow-800 border-4
        border-opacity-100 p-2">
        border-opacity-100        </p>

    <p class="border-yellow-800 border-4
        border-opacity-75 p-2">
        border-opacity-75         </p>

    <p class="border-yellow-800 border-4
        border-opacity-50 p-2">
        border-opacity-50         </p>

    <p class="border-yellow-800 border-4
        border-opacity-25 p-2">
        border-opacity-25         </p>

    <p class="border-pink-800 border-4
        border-opacity-100 p-2">
        border-opacity-100        </p>

    <p class="border-pink-800 border-4
        border-opacity-75 p-2">
        border-opacity-75         </p>

    <p class="border-pink-800 border-4
        border-opacity-50 p-2">
        border-opacity-50         </p>

    <p class="border-pink-800 border-4
        border-opacity-25 p-2">
        border-opacity-25         </p>
</div>
</body>

</html>

```

Tailwind Border Opacity Class

border-opacity-100	border-opacity-100	border-opacity-100
border-opacity-75	border-opacity-75	border-opacity-75
border-opacity-50	border-opacity-50	border-opacity-50
border-opacity-25	border-opacity-25	border-opacity-25

Opacity Class

BORDER STYLE

It is the same as CSS border-style property. It is used for controlling the style of an element's borders.

Border-Style Classes:

- border-solid
- border-dashed
- border-dotted
- border-double
- border-none

Example:

```
<!DOCTYPE html>
<html>

<head>
  <link href=
"http://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
css"
      rel="stylesheet">
</head>

<body class="text-center">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind Border Style Class
  </h1>
```

```

<div class="mx-4 grid grid-cols-5 gap-2
bg-green-200 p-2">

    <!-- First sub div is not for roundig-->
    <div class="border-4 border-green-900
border-solid
                bg-green-400 w-full h-12">
        Solid
    </div>

    <div class="border-4 border-green-900
border-dashed
                bg-green-400 w-full h-12">
        Dashed
    </div>

    <div class="border-4 border-green-900
border-dotted
                bg-green-400 w-full h-12">
        Dotted
    </div>

    <div class="border-4 border-green-900
border-double
                bg-green-400 w-full h-12">
        Double
    </div>

    <div class="border-4 border-green-900
border-none
                bg-green-400 w-full h-12">
        None
    </div>
</div>
</body>

</html>

```

Tailwind Border Style Class

Solid	Dashed	Dotted	Double	None
-------	--------	--------	--------	------

Border Style

RING OPACITY

By using this class, we can set the opacity of any ring. In CSS, we do it by using the CSS Opacity property.

Ring Opacity class:

ring-opacity-0: It control the opacity of an element's placeholder color using the ring-opacity-{amount} utilities.

RING WIDTH

All the properties have covered as in class form. The class is used to set the ring width of buttons.

Ring – Width Classes

This class is used to set the box-shadow 0 0 #000 classes as follows:

- ring: It is used to set the ring.
- ring-0: It defines no ring.
- ring-1: It is link to be created by using this class.
- ring-2: It is used to create the medium ring around the button.
- ring-4: It is used to create the large ring around the button.
- ring-8: It is used to create the widest ring around the button.
- ring-inset: It is used to generate ring inside, like padding works on any element.

Ring: It is used to apply solid box-shadow of a specific thickness to an element. These rings are a semi-transparent blue color by default, similar to the default focus ring style in many systems.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet"
    href="https://unpkg.com/tailwindcss@^1.0/dist/
tailwind.min.css"
  >
</head>
```

```

<body class="text-center">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind Ring Opacity Class
  </h1>
  <div class="mx-16 grid grid-cols-5 gap-2 p-2">
    <button class="ring-0 ring-green-600
bg-green-400
                                ring-opacity-25 w-full h-12
rounded-lg">
      Ring-0
    </button>
    <button class="ring-1 ring-green-600
bg-green-400
                                ring-opacity-25 w-full h-12
rounded-lg">
      Ring-1
    </button>
    <button class="ring-2 ring-green-600
bg-green-400
                                ring-opacity-25 w-full h-12
rounded-lg">
      Ring-2
    </button>
    <button class="ring-4 ring-green-600
bg-green-400
                                ring-opacity-25 w-full h-12
rounded-lg">
      Ring-4
    </button>
    <button class="ring-8 ring-green-600
bg-green-400
                                ring-opacity-25 w-full h-12
rounded-lg">
      Ring-8
    </button>
  </div>

</body>
</html>

```

Tailwind Ring Opacity Class

Ring-0

Ring-1

Ring-2

Ring-4

Ring-8

Box-Shadow and Opacity Class

TAILWIND EFFECTS

This shadow Effect properties of box-shadow in CSS.

Box-Shadow Classes:

- shadow-sm: Used to create a faded or small shadow effects on the box.
- shadow: Used to create normal shadow effects on the box.
- shadow-md: Used to create md effects on the box.
- shadow-lg: Used to create lg shadow effects on the box.
- shadow-xl: Used to create xl shadow effects on the box.
- shadow-2xl: Used to create 2xl shadow effects on the box.
- shadow-inner: Used to create shadow effects inside the box.
- shadow-none: Used to dilute the shadow effects.

Example of Box-shadow class with its opacity class.

```
<!DOCTYPE html>
<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
      rel="stylesheet">
</head>

<body class="text-center mx-4 space-y-2">
  <h1 class="text-green-600 text-4xl font-bold ">
    Tailwind Box Shadow Class
  </h1>
  <div class="flex gap-4 text-center p-2
  justify-evenly">
```

```

    <div class="shadow-sm w-24 h-24 bg-green-200
        rounded-lg">shadow-sm
    </div>
    <div class="shadow w-24 h-24 bg-green-200
        rounded-lg">shadow
    </div>
    <div class="shadow-md w-24 h-24 bg-green-200
        rounded-lg">shadow-md
    </div>
    <div class="shadow-lg w-24 h-24 bg-green-200
        rounded-lg">shadow-lg
    </div>
    <div class="shadow-xl w-24 h-24 bg-green-200
        rounded-lg">shadow-xl
    </div>
    <div class="shadow-2xl w-24 h-24 bg-green-200
        rounded-lg">shadow-2xl
    </div>
</div>

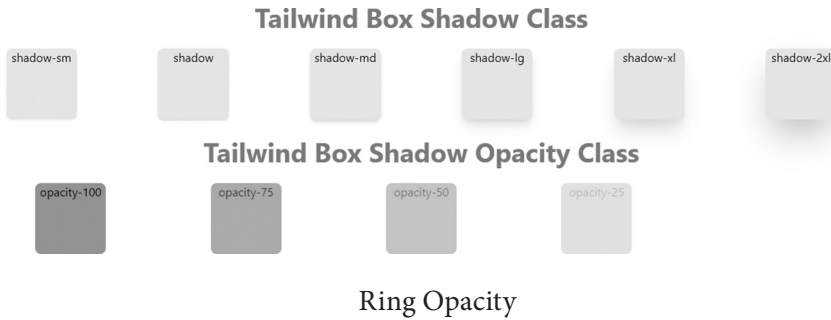
<h1 class="text-green-600 text-4xl font-bold ">
    Tailwind Box Shadow Opacity Class
</h1>
<div class="flex gap-2 text-center p-2
justify-evenly">
    <div class="opacity-100 w-24 h-24
bg-green-500
        rounded-lg <>opacity-100
    </div>
    <div class="opacity-75 w-24 h-24 bg-green-500
        rounded-lg">opacity-75
    </div>
    <div class="opacity-50 w-24 h-24 bg-green-500
        rounded-lg">opacity-50
    </div>
    <div class="opacity-25 w-24 h-24 bg-green-500
        rounded-lg">opacity-25
    </div>
    <div class="opacity-0 w-24 h-24 bg-green-500
        rounded-lg">opacity-0
    </div>
</div>

</div>

```

```
</body>
```

```
</html>
```



TAILWIND TABLES

Border Collapse

It accepts lots of value in tailwind. It is used to set the borders of the cell present inside the table and tells whether these cells will share a common border or not. We can do that by using CSS border-collapse property in CSS.

Table Layout

We can set the display of the layout of the table using this class. In CSS, we do it using the CSS table-layout property.

Table Layout classes:

- table-auto
- table-fixed
- table-auto: Used to allow the table to automatically size columns to fit the contents of the cell

Example:

```
<html>
<head>
  <link href=
  "https://unpkg.com/tailwindcss@^1.0/dist/tailwind.min.
  css"
```

```

        rel="stylesheet">
</head>

<body class="text-center mx-4 space-y-2">
  <h1 class="text-green-600 text-3xl font-bold">
    Tailwind CSS Table Collapse Class    </h1>

    <div class="grid grid-flow-col">
      <table class=" table-auto border-Collapse
border border-green-900 text-center my-8">
        <thead>
          <tr>
            <th class="border text-red-600 text-
2xl border-green-600 ">Serial Number</th>
            <th class="border text-red-600 text-2xl
border-green-600">Name</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td class="border border-green-600">1</td>
            <td class="border border-green-600">A</td>
          </tr>
          <tr>
            <td class="border border-green-600">2</td>
            <td class="border border-green-600">B</td>
          </tr>
          <tr>
            <td class="border border-green-600">3</td>
            <td class="border border-green-600">C</td>
          </tr>
        </tbody>
      </table>
    </div>

    <h1 class="text-green-600 text-3xl font-bold
">Tailwind Table Layout Class</h1>
    <div class="flex justify-evenly">
      <table class="table-auto border-separate border
border-green-900 my-6">

```

```

<thead>
  <tr>
    <th class="border border-green-600">Framework</th>
    <th class="border border-green-600">Description</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td class="border border-green-600">Tailwind
CSS</td>
    <td class="border border-green-600">
      Tailwind is a highly customizable,
      low-level CSS framework that gives you all
      of the building blocks
    </td>
  </tr>
  <tr>
    <td class="border border-green-600">Bulma</td>
    <td class="border border-green-600">
      Bulma is Simple, easily customizable and
      doesn't
      impose Javascript implementations.
    </td>
  </tr>
  <tr>
    <td class="border border-green-600">Bootstrap</td>
    <td class="border border-green-600">
      Bootstrap is a free and open-source CSS
      framework directed at responsive,
      mobile-first
      front-end web development.
    </td>
  </tr>
</tbody>
</table>
</div>
</body>
</html>

```

Tailwind CSS Table Collapse Class

Serial Number	Name
1	A
2	B
3	C

Tailwind Table Layout Class

Framework	Description
Tailwind CSS	Tailwind is a highly customizable, low-level CSS framework that gives you all of the building blocks
Bulma	Bulma is Simple, easily customizable and doesn't impose Javascript implementations.
Bootstrap	Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development.

Table Collapse and Layout

BENEFITS OF USING TAILWIND CSS

Now that you are familiar with why and why with Tailwind CSS, let's take a look at some of its outstanding benefits.

1. Tailwind CSS is a highly customizable framework. Although it comes with automatic configuration, it is easy to extract it with a tailwind.config.js file. The configuration file enables customization of color palettes, style, spaces, themes, etc.
2. It also contains simple use patterns and eliminate the difficulty of designing classes with Tailwind CSS. Availability of standard help patterns solves many problems such as classifying, editing, extracting, and much more. App classes simplify the process of creating custom sections. You do not need to encrypt solid Tailwind CSS. You can use the theme function () to extract values from configuration files.
3. It can be improved by using PurgeCSS. The great advantage of Tailwind is that the configuration can be done using PurgeCSS. PurgeCSS can significantly reduce file size by scanning HTML and removing unused classes. It is easy to set up PurgeCSS integrated with Tailwind CSS and it is highly recommended to do so before removing the site. As the size of the project grows, the size of the CSS file also increases. However, this does not happen when you are using Tailwind. The use of a standard set of classes keeps the file size smaller as long as the project is in operation.
4. It allows the building of complex responsive structures. The Tailwind CSS framework uses the default mobile method. The availability of resource classes makes it easy to build complex responsive structures with ease. The service categories can be used in all different

intersections with a condition that helps to build complex responsive structures without hassle.

5. The Tailwind CSS community can be the perfect solution. Be a part of this community and get full help from other users if needed. Get answers to all your CSS related questions while on the go and create special apps/websites without hindrance. The Tailwind CSS team provides instant solutions for quick fixes.

WHY USE TAILWIND CSS?

Tailwind CSS is a low-level, customized CSS framework that rises. It is often compared to Bootstrap. While, they are fundamentally different you can use instead of delivering pre-designed components as Bootstrap does, Tailwind provides building blocks that allow the engineer to build designs quickly.

Here are a few captions:

Syntax

Tailwind syntax is a very basic add-on to HTML or JSX element category names that represent one or more CSS declarations. Here is what it looks like:

```
<div class = "flex p-6 bg-white rounded-lg shadow-xl">
  <div class = "ml-6 pt-1">
    <h4 class = "text-xl text-gray-900 leading-tight">
text </h4>
    <p class = "text-base text-gray-600 leading-
normal"> You have a new message! </p>
  </div>
</div>
```

Small Styles

The advantage of using Tailwind to write CSS is that it does not grow in sequence. By having resource class names to be shared between elements, it ensures a consistent batch size. Additionally, Tailwind paired with Purge CSS can remove any unused styles for codebase.

Less Cognitive Load

Tailwind is designed to fit partially. It is very easy to break up site elements into smaller components and not contaminate the codebase with external

CSS objects or classes. In addition, each class is lined up in a section, making it easy to read and understand.

Better Composition

Tailwind automatically makes the engineer think of parts. It makes us want to keep it small and avoid duplication. This may be due to the low amount of space we need to achieve the building; how it alerts us when styles are repeated or exploded; and how easy it is to learn and remember styles when divided into smaller parts.

Consistency

In addition, Tailwind has a number of pre-designed units that balance colors that will reduce or eliminate the need to implement a design program. This also means that there is a low learning curve for an experienced engineer at Tailwind to learn the styles in an existing project.

Customization

Tailwind provides the developer with an open source theme that will work best for implemented projects. However, it encourages developers to extend or modify these classes with the `tailwind.config.js` file.

Tailwind configuration file allows developers to interfere directly with Tailwind. Any changes will be reflected in the final style sheet. It also allows you to add custom plugins, which opens up a whole world of opportunities for third parties. Since Tailwind uses PostCSS as its pre-processor, it means it can expand its capabilities well without its configuration file. Most codebases will have a CSS file importing Tailwind styles via PostCSS.

Example:

```
@tailwind base;
@tailwind parts;
@tailwind resources;

@recomponent sections {
  .btn-blue {
    @use bg-blue-500 text-white font-bold py-2 px-4
    round;
  }
  .btn-blue: hover {
    @use bg-blue-700;
  }
}
```

Play CDN

The Play CDN allows including Tailwind CSS in your projects without downloading the complete package via NPM or Yarn. This is not a CSS-based CDN, instead, it is a script-based CDN. Add it to the head of your HTML file, and start using Tailwind's utility classes to style your content.

Example:

```
<!-- Import Tailwind Through Play CDN -->
  <script src="https://cdn.tailwindcss.com/"></script>
```

Fancy Underline Styles

The fancy underline styles allow for changing the color, style, thickness, offset of the underlines. Lets' try each of them concisely.

- Text Decoration Color {decoration-sky-500}
- Text Decoration Style {decoration-dotted, decoration-wavy}
- Text Decoration Thickness {decoration-2}
- Text Underline Offset {underline-offset-8}

TAILWIND CSS: DISADVANTAGES

Style and HTML Are Mixed

Since you do not have to write your own CSS rules, Tailwind works differently than most CSS frameworks. While this is good for people unfamiliar with CSS, it also means that Tailwind integrates style rules with your HTML files.

This is contrary to the principle of “anxiety separation”. Many developers prefer to separate the layout and style of the page, claiming that the classes make the tailwind markup process more efficient.

Learning Takes Time

Tailwind CSS is very readable because of its built-in class. Even for experienced engineers, learning to use and make full use of pre-built classrooms can be challenging. However, like any other development project, practice makes perfect. However, when it comes to writing CSS classes you are confident and fast, Tailwind may not be the best option for you. While that may be true, Tailwind tends to speed up CSS style over time.

Lack of Essential Style Elements

Unlike Blum and Bootstrap, Tailwind does not have many style items. Unfortunately, this means you have to add features such as titles, buttons, and navigation bar to web applications.

Documentation

Although it has made great struggle in finding and installing guides and video tutorials, it still lags behind competitors like Bootstrap. Keep in mind, though, that this may take some time. For this reason, you may need to customize the draft to suit your needs.

CHAPTER SUMMARY

Here, we have learned that Tailwind can be used with CSS to enhance its properties we can make pages more attractive. It can be added just by installing or adding CDN in you html file and you can start working on it. Also, you will get to know about its advantages, disadvantages, and its features.

FURTHER READING

Awesome Tailwind CSS - <https://github.com/aniftyco/awesome-tailwindcss>, accessed on (2022 May 24).

Build your next idea even faster - <https://tailwindui.com/>, accessed on (2022 May 24).

daisyUI components - <https://daisyui.com/>, accessed on (2022 May 24).

Flowbite - Tailwind CSS component library - <https://flowbite.com/docs/getting-started/introduction/>, accessed on (2022 May 24).

Free Tailwind Components - <https://dev.to/narottam04/free-tailwind-components-for-your-next-project-2gka>, accessed on (2022 May 24).

Introduction to Tailwind CSS - <https://www.geeksforgeeks.org/introduction-to-tailwind-css/>, accessed on (2022 May 24).

Rapidly build modern websites without ever leaving your HTML - <https://tailwindcss.com/>, accessed on (2022 May 24).

TAILWIND CSS - <https://tailwinduikit.com/>, accessed on (2022 May 24).

Tailwind CSS UI - <https://www.figma.com/community/file/768809027799962739>, accessed on (2022 May 24).

Tailwind CSS UI Components, Blocks and Templates - <https://tailgrids.com/>, accessed on (2022 May 24).

NOTE

1 Get started with Tailwind CSS – Tailwind CSS.

Foundation

IN THIS CHAPTER

- Introduction
- Versions of Foundation
- Foundation installation
- Usage of Foundation

In the last chapter, you learned about a famous CSS framework, i.e., Tailwind, and now, we will discuss another CSS framework that is highly popular, Foundation.

INTRODUCTION

The Foundation is one of the key frameworks for designing great responsive websites. It works on all others types of devices and provides with HTML, CSS, and JavaScript plugins.

The Foundation started as an interior style guide and developed by ZURB in 2008. ZURB is a product design based company in Campbell, California. It is released Foundation 2.0 as an open source in October 2011. The latest version of the Foundation says 6.1.1, released in December 2015. It is used by many companies like Facebook, eBay, Mozilla, Adobe,

and Disney. Frame built on SaaS-like bootstrap. It is very complex, flexible, and easily customizable. It comes with CLI, so it's easy to use with lots of modules. It provides Fastclick.js tool for immediate delivery on mobile devices.

A tooltip is a small text or a word that appears when a user moves a mouse over a specific word or button or controls a website. Tooltip is a very common feature of modern websites.

The framework is based on Bootstrap, similar to SaaS. It is very complex, flexible, and adjustable. It comes with a command line interface, making it easy to use with module bundlers. Email Frame gives you responsive HTML emails, which can be read on any device.

WHY FOUNDATION?

- Allows faster development using Sass compiler, much faster than default compiler.
- Add price tables, switches, joyride, distance slides, lightbox, and more to your website.
- By making quick codes and controls, including development packages like Grunt and Libsass.
- Foundation for Apps allows to build fully responsive web applications.
- It is an open source and has good community support.

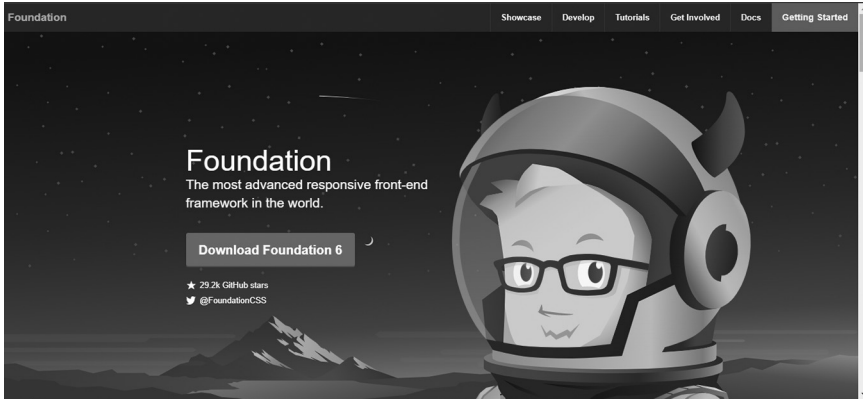
INSTALLATION

The Foundation is one of the key frameworks for designing great responsive websites. It works on all types of devices and provides with HTML, CSS, and JavaScript plugins.

The Foundation started as an interior style guide and developed by ZURB in 2008. ZURB is a product design based company in Campbell, California. The latest version of the Foundation says 6.1.1, released in December 2015. Here, we will discuss about how to install and use Foundation on website.

Download Foundation

When you open the link <https://foundation.zurb.com>, you will see a screen as shown right below:



Foundation Official Site

Click the Download Foundation 6 button; you will be get you Foundation zip file. It will automatically download once you click that button. As the files consist of all things in the framework so every time you don't need to include separate files for individual functionality. At this time, the latest version (Foundation 6) was downloaded.

Template

A basic HTML template using Foundation is as follows:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Foundation Template</title>
    <meta content = "width = device-width, initial-
scale = 1" name = "viewport">

    <link href="https://cdn.jsdelivr.net/npm/
foundation-sites@6.5.1/dist/css/foundation.min.css"
rel="stylesheet">

    <!-- Compressed JavaScript CDN Link -->

  </head>
  <script type = "text/javascript"
src="http://cdn.jsdelivr.net/npm/foundation-
sites@6.5.0/dist/js/foundation.min.js"></script>
  <script type = "text/javascript">
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/
foundation/6.0.0/js/vendor/jquery.min.js"> </script>
```

```
<body>
  <p>You can write content</p>
</body>
</html>
```

FILE STRUCTURE OF FOUNDATION FOLDER

Once Foundation is downloaded, extract the files, and also you will see the following file/directory structure. As you can see, there are CSS and JS compiled as well as compiled and minified CSS and JS.

We are using the CDN link of the library throughout this lesson.

HTML5 Doctype

Foundation consists of various HTML elements and CSS properties that require the use of the HTML5 doctype. The following code for HTML doctype such as `<doctype html>` should be included at the beginning of all projects using Foundation.

MOBILE RESPONSIVE FIRST

It helps to be as responsive to mobile devices. You can include the viewport meta tag to the `<head>` element, to ensure proper rendering and touch zooming on mobile devices.

- `<meta content="width=device-width, initial-scale=1" name="viewport">`

Its width property controls the width of the device. Set it to device-width will make sure that it is rendered across various devices (mobiles, desktops, tablets...) properly.

- `initial-scale=1.0`

It ensures that when loaded, the web page will be rendered at a 1:1 scale, with no zooming applied out of the box.

Example:

```
<!DOCTYPE html>
<html>
  . . . .
</html>
```

INITIALIZATION OF COMPONENTS

The jQuery is required in Foundation for its components like modals and dropdown. You can add them in the head section using script tag.

```
<script src = "http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.0.0/js/vendor/jquery.min.js">
```

```
<script src = "http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.0.0/js/foundation.min.js">
</script>
```

```
</script>
```

WHAT COMES WITH FOUNDATION?

Foundation has a number of components and frameworks to help you build a responsive site without having to worry about your entire foundation, the base (see what we did there?) Code. You can see all parts of the Foundation on a single page called Kitchen Sink or check out a brief summary of what the Foundation includes:

- Grid: Our grid works on almost on all the device and has support for breeding, source ordering, offsets, and device launch.
- Buttons: There are lightweight button styles, presentation, and color to customize your button as easy as adding a class.
- Navigation: Basic navigation styles include a solid top bar with a dropdown; button search bars; cool icon bar; implementation of bitchin 'off-canvas'; and dozens of other navigation structures.
- Plugins: We've included a JavaScript plugin written for Basic only: pop-up modals (Reveal); add the required verification of forms (Abide); create content tabs; warnings; and tons more.

COMPATIBILITY

Foundation CSS is tested across various browsers and devices.

Example:

- Chrome
- Firefox

- Safari
- Opera
- Mobile Safari
- IE Mobile
- Edge
- Internet Explorer
- Android Browser

WHAT WILL NO LONGER WORK?

- The grid: Foundation's grid uses box-sizing: border-box to apply gutters to columns, but this property now isn't supported in IE8.
- Desktop styles: Because the framework is written mobile-first, browsers that don't support media queries will display the mobile styles of the site.
- Java Script: Our plugins use a number of handy ECMAScript 5 features that aren't supported in IE8.

BASIC PRINCIPLES

- Use proper HTML tags in the work when you mark navigation, lists, links, controls, and more.
- If the controller or form element does not have a text label, add one. You can use visual classes to hide labels visually while keeping accessibility. Use alt attribute in all images to define what they are.
- Page content should make sense even if the page is being read by a user, or if the user doesn't see the color and cannot use color-based labeling.
- Make everything workable on the keyboard and mouse, and all of our components work with keyboards, mice, and touch screens outside the box.

BASIC CSS KITCHEN SINK

Foundation CSS is a pre-open and responsive source framework developed by ZURB foundation in September 2011, making it easy to organize amazing responsive websites, apps, and emails that look amazing and accessible on any device. It is used by various companies such as Facebook, Mozilla, Adobe, and Disney. Frame built on SaaS-like bootstrap. It is very complex, flexible, and easily customizable. It comes with CLI, so it's easy to use with lots of modules. Provides Fastclick.js tool for immediate delivery on mobile devices. In this section, we will discuss Kitchen Sink Switch in Foundation CSS.

Kitchen Sink is a page on Foundation CSS where all the major components are listed with their codes. Contains all the basic templates for each component and its code. The Interchange Component in Foundation CSS is useful for creating responsive website designs. It is helpful to upload different web formats or different images to the same website based on screen size. By using this component, we can upload a large size screen size, medium size image size, and a small size screen size.

Accordion

Accordions are elements that help to organize and navigate multiple documents in a single container. They can be used for switching between items in the container.

Syntax:

```
<div class="accordion" data-accordion>
  <div class="accordion-item" data-accordion-item>
    <a href="#" class="accordion-title"></a>
    <div class="accordion-content" data-tab-content>
      <p>Content </p>
    </div>
  </div>
</div>
```

Foundation CSS Kitchen Sink Accordion Classes:

- **Accordion:** Used to create the accordion and is linked with the data-accordion attribute.

- **Accordion-item:** Used to create the different panes inside an accordion and is linked with the `data-accordion-item` attribute. Any pane or accordion item can be made active by using the `is-active` class along with the `accordion-item` class.
- **Accordion-title:** Used to create an accordion title. It is used along with `<a>` tag with the class name of `accordion-title`.
- **Accordion-content:** Used to display all the content inside each accordion tab and is linked with the `data-tab-content` attribute.

FOUNDATION CSS KITCHEN SINK ACCORDION ADVANCED OPTIONS

- **Foundation CSS** provides the users with some advanced options as well to increase accessibility. These can be used with the `data-accordion` attribute.
- **All-closed:** By default, Accordion item remains active. One can change by turning on the `allowAllClosed` option to 'true'. Therefore, the div would have attribute as `data-accordion data-allow-all-closed="true"`.
- **Multi-expand:** By default, Accordion item is allowed to remain active. But, it can change by adding the `Multi-expand` option and switching its value to 'true'. While, the div would have attribute as `data-accordion data-multi-expand="true"`.
- **Disabled:** It is a unique property by which the navigation in the Accordion can be disabled. The div would have an attribute as `data-accordion disabled`.

Syntax:

```
<div class="accordion" attribute-name></div>
```

Example of Accordion:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
```

```

"http://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/css/foundation.min.css" />

    <!-- Compressed JavaScript -->
    <script type = "text/javascript"
src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.0/
jquery.min.js">
    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center">Foundation CSS Kitchen Sink
    Accordion</h1>
    <div class="accordion" data-accordion>
        <br>
        <div class="accordion-item" data-accordion-item>
            <a href="#" class="accordion-title">
                Foundation Accordion 1
            </a>

            <div class="accordion-content" data-tab-
            content>
                <p>Subheading</p>

                <a href="#">Link</a>
            </div>
        </div>
        <div class="accordion-item" data-accordion-item>
            <a href="#" class="accordion-title">
                Foundation Accordion 2
            </a>
            <div class="accordion-content"
data-tab-content>

```

```

        <p>Subheading</p>

        <a href="#">Link</a>
    </div>
</div>
<div class="accordion-item" data-accordion-item>
    <a href="#" class="accordion-title">
        Foundation Accordion 3
    </a>
    <div class="accordion-content"
        data-tab-content>
        <p>Subheading</p>

        <a href="#">Link</a>
    </div>
</div>
</div>
<script>
    $(document).foundation();
</script>
</body>
</html>
```



Foundation Accordion

Here is the example of Foundation with disable option:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"http://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/css/foundation.min.css" rel="stylesheet" />
```

```

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center"> Disbale option in
Foundation</h1>

    <div class="accordion" data-accordion disabled>
        <div class="accordion-item" data-accordion-item>
            <a href="#" class="accordion-title">
                Foundation Accordion 1
            </a>
            <div class="accordion-content" data-tab-content>
                <p>Accordion 1 content</p>
                <a href="#">Link</a>
            </div>

            <div class="accordion-item is-active" data-
accordion-item>
                <a href="#" class="accordion-title">
                    Foundation Accordion 2
                </a>
                <div class="accordion-content" data-tab-
content>
                    <p>Accordion 2</p>

                    <a href="#">Link</a>
                </div>
            </div>
        </div>
    </div>
</div>

```

```

<script>
    $(document).foundation();
</script>
</body>
</html>

```

Disbale option in Foundation

Foundation Accordion 1	+
Foundation Accordion 2	-
Accordion 2	
Link	

Foundation Accordion Disable Option

ADVANCED OPTIONS

Multi-Expand

By default, only a pane of an accordion open at a time. This can be changed by setting the multi-expand option to true.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS Link -->
    <link rel="stylesheet" href=
"http://cdn.jsdelivr.net/npm/foundation-sites@6.7.3/
dist/css/foundation.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.0/
jquery.min.js">
    </script>
    <script src=

```

```

"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.3/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:10%
  }
</style>

<body>
  <h1 class="text-center"> Accordion Foundation
  - Multi Expand  </h1>

  <ul class="accordion" data-accordion
data-multi-expand="true">
    <li class="accordion-item is-active"
data-accordion-item>
      <a href="#" class="accordion-title">Foundation
Accordion 1</a>
      <div class="accordion-content" data-tab-
content >
        <p>Subheading</p>
      </div>
    </li>

    <li class="accordion-item" data-accordion-item>
      <a href="#" class="accordion-title">Foundation
Accordion 2</a>
      <div class="accordion-content"
data-tab-content>
        Enter your name
        <input type="text"></input>
      </div>
    </li>
  </ul>
  <script>
    $(document).foundation();
  </script>
</body>
</html>

```

Accordion Foundation - Multi Expand

Foundation Accordion 1	–
Subheading	
Foundation Accordion 2	+

Foundation Accordion Multi Expand

All Closed

By default, at least a pane in an accordion must open. This can be changed by allowing allow All Closed option to true.

ACCORDION AND URLs

Deep Linking

When the data-deep-link is set to true, the default state of the accordion is recorded by adding a hash with the accordion panel ID to the URL when the accordion opens.

By default, accordion replaces browser history (using `history.replaceState()`). Correct this behavior by using attribute `data-update-history="true"` to merge browser history (using `history.pushState()`). In the latest case, the browser back button will track every click that opens the accordion panel.

Here is the complete example of deep linking as given below:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS Link-->
    <link rel="stylesheet" href=
"http://cdn.jsdelivr.net/npm/foundation-sites@6.7.3/
dist/css/foundation.min.css" />

    <!-- Compressed JavaScript Link - - >
    <script type = "text/javascript"
src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
```

```

    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.3/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center"> Accordion Foundation-
Multi Expand    </h1>

    <ul class="accordion" data-deep-link-smudge="true"
data-deep-link-smudge-delay="500" data-accordion
data-deep-link="true" data-update-history="true"
id="deeplinked-accordion">
        <li class="accordion-item is-active"
data-accordion-item>
            <a href="#deeplink1" class="accordion-
title">Accordion 1</a>
            <div class="accordion-content" data-tab-
content id="deeplink1">
                Panel 1. Lorem ipsum dolor
            </div>
        </li>
        <li class="accordion-item" data-accordion-item>
            <a href="#deeplink2" class="accordion-
title">Accordion 2</a>
            <div class="accordion-content" data-tab-
content id="deeplink2">
                Panel 2. Lorem ipsum dolor
            </div>
        </li>
        <li class="accordion-item" data-accordion-item>
            <a href="#deeplink3" class="accordion-
title">Accordion 3</a>
            <div class="accordion-content" data-tab-
content id="deeplink3">
                Panel 3. Lorem ipsum dolor

```

```

        </div>
    </li>
</ul>
<script>
    $(document).foundation();
</script>
</body>
</html>

```

Accordion Foundation- Deep Linking

Accordion 1	+
Accordion 2	+
Accordion 3	-
Panel 3	

Accordion Foundation – Deep Linking

CALLOUT

It is just an element with a `.callout` class applied. You can put any content inside. It combines panels and alerts from Foundation 5 into one generic container component.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css"
rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.0/
jquery.min.js">

```

```

    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Callout    </h1>

    <div class="row small-up-2 medium-up-3 large-up-5
align-center">
        <div class="column">
            <div class="callout">
                <h5>Callout.</h5>
                <a href="#">Link.</a>
            </div>
        </div>
        <div class="column">
            <div class="callout ">
                <h5>Callout.</h5>
                <a href="#">Link.</a>
            </div>
        </div>
        <div class="column">
            <div class="callout">
                <h5>Callout.</h5>
                <a href="#">Link.</a>
            </div>
        </div>
        <div class="column">
            <div class="callout">
                <h5>Callout.</h5>
                <a href="#">Link.</a>
            </div>
        </div>
        <div class="column">
            <div class="callout">
                <h5>Callout.</h5>
                <a href="#">Link.</a>
            </div>
        </div>
    </div>

```

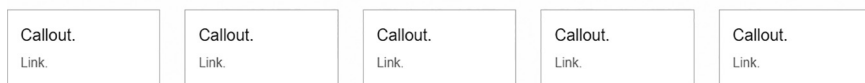
```

        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>

    </div>
    <script>
      $(document).foundation();
    </script>
  </body>
</html>

```

Accordion Foundation - Callout



Callout

Coloring

It can be colored using the `.secondary`, `.primary`, `.success`, `.warning`, or `.alert` classes. Links inside the callout will be tinted to match the color of the callout.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Compressed CSS -->
  <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css" />

  <!-- Compressed JavaScript -->
  <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
  </script>
  <script src=

```

```

"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:10%
  }
</style>

<body>
  <h1 class="text-center"> Foundation - Callout  </h1>

  <div class="row small-up-2 medium-up-3 large-up-5
align-center">
    <div class="column">
      <div class="callout secondary">
        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
    <div class="column">
      <div class="callout primary">
        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
    <div class="column">
      <div class="callout success">
        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
    <div class="column">
      <div class="callout warning">
        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
    <div class="column">
      <div class="callout alert ">
        <h5>Callout.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
  </div>

```

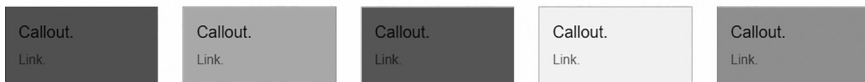
```

        </div>
    </div>

    </div>
    <script>
        $(document).foundation();
    </script>
</body>
</html>

```

Accordion Foundation - Callout



Callout Coloring

SIZING

Callouts can be sized using the `.large` and `.small` classes. These will affect the padding around content to be smaller and larger, respectively.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js" type = "text/javascript" >
    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>

```

```

</head>
<style>
  body{
    margin:10%
  }
</style>

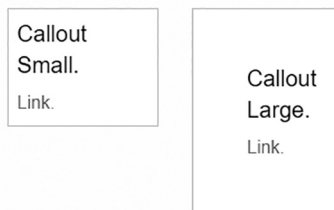
<body>
  <h1 class="text-center"> Foundation - Callout  </h1>

  <div class="row small-up-2 medium-up-3 large-up-5
  align-center">
    <div class="column">
      <div class="callout small ">
        <h5>Callout Small.</h5>
        <a href="#">Link.</a>
      </div>
    </div>
    <div class="column">
      <div class="callout large">
        <h5>Callout Large.</h5>
        <a href="#">Link.</a>
      </div>
    </div>

  </div>
  <script>
    $(document).foundation();
  </script>
</body>
</html>

```

Foundation - Callout



Callout Sizing

MAKING CLOSABLE

Adding the attribute `data-close` to element within the callout will turn it into a close trigger. When using the `data-closable` attribute, you can add Motion UI classes to the attribute to change the closing animation.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script type = "text/javascript" src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script type = "text/javascript" src=
"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Closable
</h1>

    <div class="row medium-unstack">
        <div class="column medium-4">
            <div class="alert callout" data-closable>
                <h5>Heading!</h5>
                <p>Subheading.</p>
```

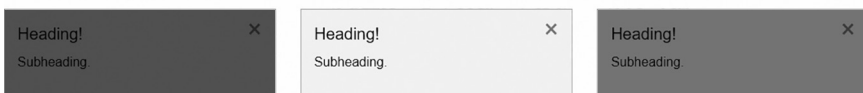
```

        <button class="close-button" aria-label="Dismiss alert" type="button" data-close>
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
</div>
<div class="column medium-4">
    <div class="success callout" data-closable>
        <h5>Heading!</h5>
        <p>Subheading.</p>
        <button class="close-button" aria-label="Dismiss alert" type="button" data-close>
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
</div>
<div class="column medium-4">
    <div class="primary callout" data-closable>
        <h5>Heading!</h5>
        <p>Subheading.</p>
        <button class="close-button" aria-label="Dismiss alert" type="button" data-close>
            <span aria-hidden="true">&times;</span>
        </button>
    </div>
</div>
</div>

<script>
    $(document).foundation();
</script>
</body>
</html>

```

Foundation - Closable



Foundation -Closable

CARD

A card is just an element that uses a card class applied. You can add any type of content inside. Be sure to wrap your content in the `.card-section` section to get a normal card view.

The card container has no padding, which allows you to put pictures inside. Use the `.card-divider` and `card section` sections to split the card slightly.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"https://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css"
rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Card    </h1>

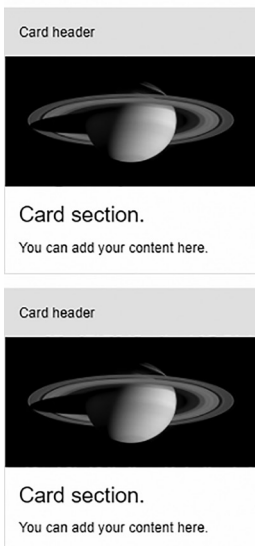
    <div class=" card" style="width: 300px;">
        <div class="card-divider">
            Card header
        </div>
```

```


<div class="card-section">
  <h4>Card section.</h4>
  <p>You can add your content here.</p>
</div>
</div>
<div class="card" style="width: 300px;">
  <div class="card-divider">
    Card header
  </div>
  
  <div class="card-section">
    <h4>Card section.</h4>
    <p>You can add your content here.</p>
  </div>
</div>
<script>
  $(document).foundation();
</script>
</body>
</html>

```

Foundation - Card



Foundation -Card

CARD DIVIDER

You can also include a class `card-divider` element as a card title, footer, or break up content.

Images

Images can work nicely with cards including one outside of the `.card-section` element to span well to the edges. or you can move the image inside the class `card section` to have padding around the image.

Sizing

You can either set the width of cards with custom CSS or add them into the grid.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <linkhref=
"http://cdnjs.cloudflare.com/ajax/libs/foundation/6.3.1/
css/foundation-flex.min.css" rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }

    .img{
```

```

    height: 100px;
    background-color: aquamarine;
  }
  .card-divider{
    background-color: white;
  }
</style>

<body>
  <h1 class="text-center"> Foundation - Card Sizing  </h1>

  <div class="row">
    <div class="column medium-4">
      <div class="card">
        <div class="card-section">
          <div class="img" "></div>
          <h4>This is a card.</h4>
          <p>Add your extra content here.</p>
        </div>
        <div class="card-divider">
          <ul class="menu callout primary">
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
          </ul>
        </div>
      </div>
    </div>
    <div class="column medium-4">
      <div class="card">
        <div class="card-section">
          <div class="img" "></div>
          <h4>This is a card.</h4>
          <p>Add your extra content here.</p>
        </div>
        <div class="card-divider">
          <ul class="menu callout primary">
            <li><a href="#">Link</a></li>
            <li><a href="#">Link</a></li>
          </ul>
        </div>
      </div>
    </div>
    <div class="column medium-4">
      <div class="card">

```

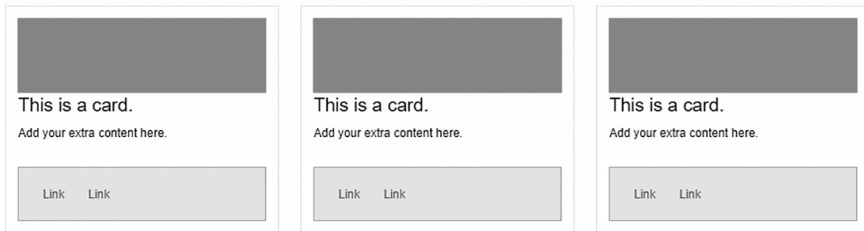
```

<div class="card-section">
  <div class="img" "></div>
  <h4>This is a card.</h4>
  <p>Add your extra content here.</p>
</div>
<div class="card-divider">
  <ul class="menu callout primary">
    <li><a href="#">Link</a></li>
    <li><a href="#">Link</a></li>
  </ul>
</div>
</div>
</div>
</div>

<script>
  $(document).foundation();
</script>
<!-- <script src="https://cdnjs.cloudflare.com/
ajax/libs/foundation/6.3.1/js/foundation.min.js"></
script> -->
</body>
</html>

```

Foundation - Card Sizing



Foundation Card Sizing

DROPDOWN

If you want to create a dropdown pane, class “dropdown-pane” and the attribute “data-dropdown” to an element. It also gives the dropdown a unique ID as well or if you want to create the dropdown trigger, add data-toggle to a <button>. The value of data-toggle is the ID of the dropdown.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"https://cdnjs.cloudflare.com/ajax/libs/foundation/6.3.1/
css/foundation-flex.min.css" rel="stylesheet"/>

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTP://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }

</style>

<body>
    <h1 class="text-center"> Foundation - Dropdown </h1>

    <button class="button" type="button" data-
toggle="example-dropdown">Toggle Dropdown</button>
    <div class="dropdown-pane" id="example-dropdown"
data-dropdown data-auto-focus="true">
        Your Information:<br>
        <form>
            <div class="grid-container">
                <div class="grid-x grid-margin-x">
                    <div class="cell medium-6">
                        <label>Name

```

```

        <input type="text" placeholder="John">
      </label>
    </div>
    <div class="cell medium-6">

      </div>
    </div>
  </div>
</form>
</div>

  <button class="button" type="button" data-
toggle="example-dropdown-1">Hoverable Dropdown</
button>
  <div class="dropdown-pane" id="example-dropdown-1"
data-dropdown data-hover="true"
data-hover-pane="true">
    Available courses are:
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>

  </div>

<script>
  $(document).foundation();
</script>
<!-- <script src="https://cdnjs.cloudflare.com/
ajax/libs/foundation/6.3.1/js/foundation.min.js"></
script> -->
</body>
</html>

```

Foundation - Dropdown



Foundation Dropdown

POSITIONING

By default, a dropdown anchors below the button that opened it. Add the class `.top`, `.right`, or `.bottom` to the dropdown to change this.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"https://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css"
rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script type = "text/javascript" src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script type = "text/javascript" src=
"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }

</style>

<body>
    <h1 class="text-center"> Foundation - Dropdown    </h1>

    <br>
<div class="row">
    <div class="columns">
        <button class="button" type="button" data-
toggle="example-dropdown3" style="float: right;">Left
Aligned</button>
        <div class="dropdown-pane left" id="example-
dropdown3" data-dropdown>
            This is Right Aligned

```

```

    </div>
    <button class="button" type="button" data-
toggle="example-dropdown4">Right Aligned</button>
    <div class="dropdown-pane right" id="example-
dropdown4" data-dropdown>
        This is Left Aligned
    </div>
</div>
</div>
</div>

<div class="row">
    <div class="columns">
        <button class="button float-right" type="button"
data-toggle="example-dropdown5">Bottom-right Aligned</
button>
        <div class="dropdown-pane bottom" id="example-
dropdown5" data-dropdown>
            This is Bottom-right Aligned
        </div>

        <button class="button float-left" type="button" data-
toggle="example-dropdown6">Bottom-left Aligned</button>
        <div class="dropdown-pane bottom" id="example-
dropdown6" data-dropdown>
            This is Bottom-left Aligned
        </div>
    </div>
</div>
</div>

<script>
    $(document).foundation();
</script>
<!-- <script src="https://cdnjs.cloudflare.com/
ajax/libs/foundation/6.3.1/js/foundation.min.js"></
script> -->
</body>
</html>

```

Foundation - Dropdown



Foundation - Dropdown Position

OTHER POSITIONING

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
    "https://cdnjs.cloudflare.com/ajax/libs/
    foundation/6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
    jquery.min.js">
    </script>
    <script src=
    "HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
    dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }
    .dropdown{
        display: flex;

        justify-content: space-evenly;
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Dropdown
    Positioing </h1>

    <br>
    <!-- Bottom Left -->
    <div class="dropdown">

```

```

    <button class="button" type="button" data-
toggle="example-dropdown-bottom-left">Toggle
Dropdown</button>
<div class="dropdown-pane" data-position="bottom"
data-alignment="left" id="example-dropdown-bottom-
left" data-dropdown data-auto-focus="true">
    Dropdown content in here
</div>

<!-- Bottom Center -->
<button class="button" type="button" data-
toggle="example-dropdown-bottom-center">Toggle
Dropdown</button>
<div class="dropdown-pane" data-position="bottom"
data-alignment="center" id="example-dropdown-bottom-
center" data-dropdown data-auto-focus="true">
    Dropdown content in here
</div>

<!-- Bottom Right -->
<button class="button" type="button" data-
toggle="example-dropdown-bottom-right">Toggle
Dropdown</button>
<div class="dropdown-pane" data-position="bottom"
data-alignment="right" id="example-dropdown-bottom-
right" data-dropdown data-auto-focus="true">
    Dropdown content in here
</div>

<!-- Top Left -->
<button class="button" type="button" data-
toggle="example-dropdown-top-left">Toggle Dropdown</
button>
<div class="dropdown-pane" data-position="top" data-
alignment="left" id="example-dropdown-top-left" data-
dropdown data-auto-focus="true">
    Dropdown content in here
</div>

<!-- Top Center -->
<button class="button" type="button" data-
toggle="example-dropdown-top-center">Toggle Dropdown</
button>

```

```

<div class="dropdown-pane" data-position="top" data-
alignment="center" id="example-dropdown-top-center"
data-dropdown data-auto-focus="true">
  Dropdown content in here
</div>

<!-- Top Right -->
<button class="button" type="button" data-
toggle="example-dropdown-top-right">Toggle Dropdown</
button>
<div class="dropdown-pane" data-position="top" data-
alignment="right" id="example-dropdown-top-right"
data-dropdown data-auto-focus="true">
  Dropdown content in here
</div>
</div>

<script>
  $(document).foundation();
</script>
<!-- <script src="https://cdnjs.cloudflare.com/
ajax/libs/foundation/6.3.1/js/foundation.min.js"></
script> -->
</body>
</html>

```

Foundation - Dropdown Positioning



Foundation Dropdown Advance Positioning

MEDIA OBJECT

It will help you create this common repeatable pattern and also can be used in several different ways. It is a container with the class `.media-object`, and two or three sections with the class `.media-object-section`.

Example:

```

<!DOCTYPE html>
<html>

```

```

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
    "https://cdnjs.cloudflare.com/ajax/libs/
    foundation/6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
    jquery.min.js">
    </script>
    <script src=
    "HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
    dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }
    .media-object{
        margin:10px;
        padding:20px;
        box-shadow: 18px 10px 10px 0px grey;
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Media Object </h1>

    <br>
    <!-- Bottom Left -->

    <div class="media-object">
        <div class="media-object-section">
            <div class="thumbnail">
                
            </div>
        </div>
        <div class="media-object-section main-section">

```

```

    <h4>Heading Text</h4>
    <p>Paragraph Content</p>
  </div>
</div>

  <script>
    $(document).foundation();
  </script>
</body>
</html>

```

Foundation - Media Object



Media Object

SECTION ALIGNMENT

Every section aligns to the top by default, but individual sections can also be middle- or bottom-aligned with the `.middle` and `.bottom` classes.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Compressed CSS -->
  <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css" />

  <!-- Compressed JavaScript -->
  <script src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
  </script>
  <script src=
"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">

```

```

    </script>
</head>
<style>
    body{
        margin:10%;

    }
    .media-object{
        margin:10px;
        padding:20px;
        box-shadow: 18px 10px 10px 0px grey;
    }
</style>

<body>
    <h1 class="text-center"> Foundation - Media Object
</h1>

    <br>
<!-- Bottom Left -->

<p>Middle alignment</p>

<div class="media-object">
    <div class="media-object-section middle">
        <div class="thumbnail">
            <img src= "assets/img/media-object/avatar-2.
jpg">
        </div>
    </div>
    <div class="media-object-section">
        <h4>Heading</h4>
        <p>Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut
quis nostrud exercitation laboris nisi ut aliquip ex
ea consequat. Duis reprehenderit in voluptate velit
esse cillum dolore eu fugiat nulla pariatur. aute
irure dolor in reprehenderit in voluptate non
proident, sunt in culpa qui officia mollit anim id est
laborum. velit esse cillum dolore eu fugiat labore et
dolore magna aliqua. Ut enim ad minim veniam,nulla
pariatur. Excepteur sint occaecat cupidatat

```

```

    </div>
    <div class="media-object-section bottom">
        <div class="thumbnail">
            <img src= "assets/img/media-object/avatar-3.
jpg">
        </div>
    </div>
</div>

<div class="media-object">
    <div class="media-object-section bottom">
        <div class="thumbnail">
            <img src= "assets/img/media-object/avatar-2.
jpg">
        </div>
    </div>
    <div class="media-object-section">
        <h4>Heading</h4>
        <p> Duis aute irure dolor in in voluptate velit
esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint non proident, sunt in culpa qui
exercitation ullamco laboris ut aliquip ex ea
commodo officia deserunt mollit anim id est
laborum. Lorem ipsum dolor sit amet, consectetur
ut labore et dolore magna aliqua. Ut enim ad minim
veniam, quis elit, sed do eiusmod tempor
incididunt
        </p>
    </div>
    <div class="media-object-section bottom">
        <div class="thumbnail">
            <img src= "assets/img/media-object/avatar-3.
jpg">
        </div>
    </div>
</div>

    <script>
        $(document).foundation();
    </script>
</body>
</html>

```

OFF-CANVAS

Its panels are set outside the viewing hole and slide inward when activated. Setting up an off-canvas format in the Foundation is very easy. The Foundation's Off-canvas is a source of mobile navigation pattern that can also be used to create a responsive sidebar. It can open in any direction, left, right, up, and down. There are options to allow Off-canvas to push your page or forward to your page along with a few other tricks.

For setting up the Off-canvas requires two elements first create an off-canvas container with the class ".off-canvas" and the attribute data-off-canvas. This is the container that holds Off-canvas content.

The Off-canvas container needs a positioning class to determine which side of the viewport it opens from:

- .position-left
- .position-right
- .position-top
- .position-bottom

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
    "https://cdnjs.cloudflare.com/ajax/libs/
    foundation/6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
    jquery.min.js">
    </script>
    <script src=
    "HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
    dist/js/foundation.min.js">
    </script>
```

```

</head>
<style>
  body{
    margin:10%;

  }

</style>

<body>

  <br>
  <!-- Bottom Left -->

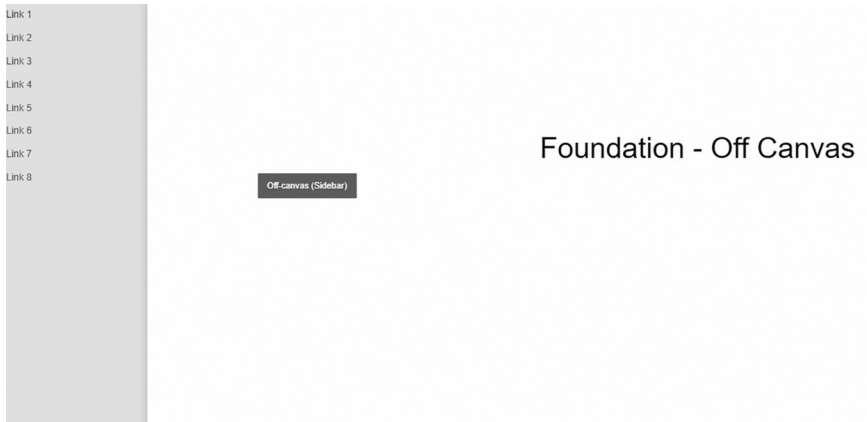
  <div class="off-canvas position-left" id="offCanvas"
  data-off-canvas>
    <!-- Off-canvas content goes here -->

    <ul class="menu vertical">
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
      <li><a href="#">Link 6</a></li>
      <li><a href="#">Link 7</a></li>
      <li><a href="#">Link 8</a></li>
    </ul>
  </div>
  <div class="off-canvas-content"
  data-off-canvas-content>
    <!-- Your page content lives here -->
    <h1 class="text-center"> Foundation - Off Canvas </h1>

    <button type="button" class="button" data-
    toggle="offCanvas">Off-canvas (Sidebar)</button>

    <script>
      $(document).foundation();
    </script>
  </body>
</html>

```



Off - Canvas

OFF-CANVAS POSITION

It is used to set position: fixed by default using the `.off-canvas` class and makes the off-canvas panel in relation to the viewport. However, you can set an off-canvas container to position: absolute by using the alternative class `off-canvas-absolute`. Also use the `.off-canvas-wrapper` when using this method.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"http://cdnjs.cloudflare.com/ajax/libs/
foundation/6.3.1/css/foundation-flex.min.css"
rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
```

```

"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:10%;

  }

</style>

<body>

  <h1> Off-canvas Position </h1>
  <button type="button" class="button" data-toggle="off
CanvasLeftSplit1">Open Left</button>
  <button type="button" class="button" data-toggle="off
CanvasRightSplit2">Open Right</button>

  <div class="grid-x grid-margin-x">
    <div class="cell small-6">
      <div class="off-canvas-wrapper">
        <div class="off-canvas-absolute position-left"
id="offCanvasLeftSplit1" data-off-canvas>
          Off-canvas content goes here
        </div>
        <div class="off-canvas-content" style="min-
height: 300px;" data-off-canvas-content>
          <p>It's off-canvas container</p>
        </div>
      </div>
    </div>
    <div class="cell small-6">
      <div class="off-canvas-wrapper">
        <div class="off-canvas-absolute position-right"
id="offCanvasRightSplit2" data-off-canvas>
          Off-canvas content goes here
        </div>
        <div class="off-canvas-content" style="min-
height: 300px;" data-off-canvas-content>
          <p>It's off-canvas container</p>
        </div>
      </div>
    </div>
  </div>

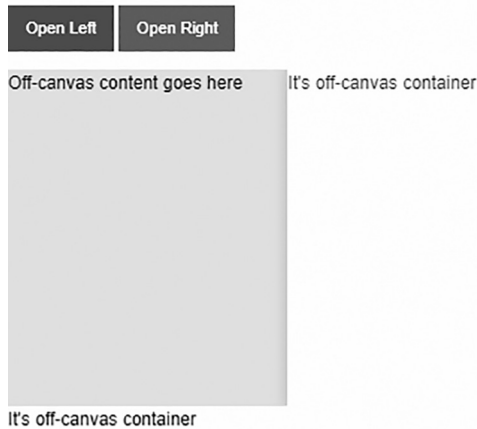
```

```

    </div>
  </div>
</div>
  <script>
    $(document).foundation();
  </script>
</body>
</html>

```

Off-canvas Position



Off canvas Position

REVEAL (MODEL)

A modal is an empty container, so that user can put content inside it, from text to forms to video to a fully grid. It is used for modal dialogs, or pop-up windows, that are handy for prototyping and production. Foundation also includes Reveal, our jQuery modal plugin, to make this easy for you.

To create a modal, add the class `reveal`, the attribute `data-reveal`, and a unique ID to a container.

Example:

```

<!DOCTYPE html>
<html>

<head>

```

```

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"https://cdnjs.cloudflare.com/ajax/libs/foundation/
6.3.1/css/foundation-flex.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }
</style>

<body>

<!-- This demo uses flex grid but you can use float
grid too -->

<div class="row">
    <div class="columns">
        <h2>Reveal (Modal)</h2>
        <p>Heading.</p>
        <div class="callout primary">
            <p>Write paragrapgh here.</p>
        </div>
    </div>
</div>

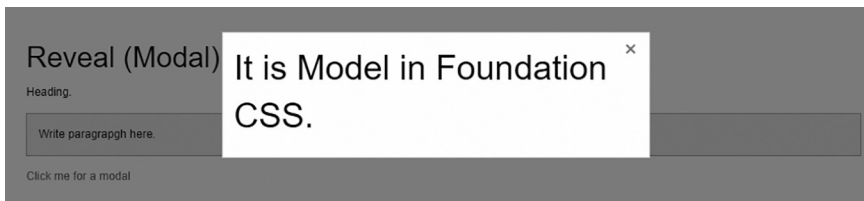
<div class="row">
    <div class="columns">
        <p><a data-open="exampleModall">Click me for a
modal</a></p>

```

```

    <div class="reveal" id="exampleModal1"
data-reveal>
    <h1>It is Model in Foundation CSS.</h1>
    <button class="close-button" type="button"
data-close aria-label="Close reveal" >
    <span aria-hidden="true">&times;</span>
    </button>
  </div>
</div>
</div>
  <script>
    $(document).foundation();
  </script>
</body>
</html>

```



Foundation -Reveal (Model)

You'll also need to open the modal and add the attribute `data-open` to any element and the value of `data-open` could be the ID of the modal.

```

<p><button class="button" data-
open="exampleModal1">Click me for a modal</button></p>

```

SIZING

For small screens, a modal is always fully 100% of the width of the screen. For medium-sized screens and larger one, the width changes to 600px. The size of a modal can be changed with these classes, which are added to the modal container:

- `.tiny`: 30% wider
- `.small`: 50% wider
- `.large`: 90% wider
- `.full`: 100% width and height, defaults the `escClose` opt to true, as well as creates a close button.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
    "https://cdnjs.cloudflare.com/ajax/libs/
    foundation/6.3.1/css/foundation-flex.min.css"
    rel="stylesheet"/>

    <!-- Compressed JavaScript -->
    <script src=
    "https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
    jquery.min.js">
    </script>
    <script src=
    "HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
    dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;

    }

    .columns{
        padding:50px;
        display: flex;
        justify-content: space-evenly;
    }
</style>

<body>

<div class="row">
    <div class="columns">
        <h2>Reveal (Modal - Sizing)</h2>
    </div>

```

```

</div>

<div class="row ">
  <div class="columns">
    <p><a data-toggle="exampleModal5" aria-
controls="exampleModal5">Click me for a tiny modal</
a></p>

    <div class="tiny reveal" id="exampleModal5"
data-reveal>
      <p>This is TINY modal</p>
      <button class="close-button" type="button" data-
close aria-label="Close reveal" >
        <span aria-hidden="true">&times;</span>
      </button>
    </div>

    <p><a data-toggle="exampleModal6">Click me for a
small modal</a></p>

    <div class="small reveal" id="exampleModal6"
data-reveal>
      <p>This is SMALL modal</p>
      <button class="close-button" type="button" data-
close aria-label="Close reveal" >
        <span aria-hidden="true">&times;</span>
      </button>
    </div>

    <p><a data-toggle="exampleModal7">Click me for a
large modal</a></p>

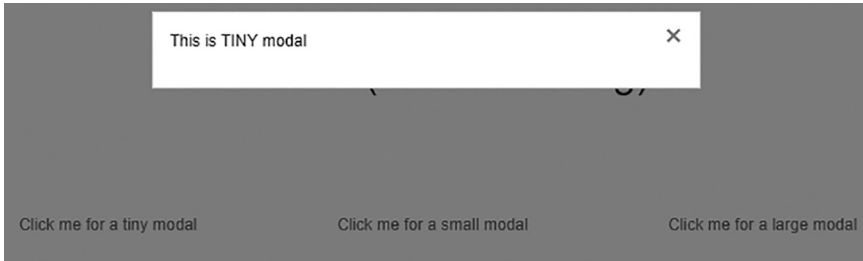
    <div class="large reveal" id="exampleModal7"
data-reveal>
      <p>This is LARGE modal</p>
      <button class="close-button" type="button" data-
close aria-label="Close reveal" >
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
  </div>
</div>
<script>

```

```

    $(document).foundation();
  </script>
</body>
</html>

```



Reveal (Modal Sizing)

FEATURES

1. **Grid:** Responsive web design is typically 1200 pixels wide, a flexible grid structure. You can also opt-out of this module, but the system is the basis for a variety of designs, the opposite solution for your web project.
2. **Navigation:** You have a wide selection of modules for ZURB Foundation project with which you can create menu and menu bars (accordion, dropdown), add navigation modes, or mark pages – including page numbers.
3. **Containers:** A container represents a first-class method of integrating multiple elements, such as text, images, or videos, into a commonplace. Here, you can find styles for common content widgets, such as accordion, tabs, dropdown locations, or modal chats.
4. **Media:** Under the “Media” you can find the Basic Framework modules needed to integrate multimedia features. The “Flex Video” component supports, for example, video embedding, and ensures that your content is adapted to suit various display sizes and resolutions. There are also photo preview sections and tips.
5. **Plugins:** For example, with the “Motion UI” library, you can download a useful library (other than Sass only) which makes creating flexible UI animations and animations as easy as pie.

6. Reusable components: In addition to the HTML elements, Foundation contains other commonly used interface elements. These include buttons with advanced features labels, advanced typographic capabilities, and formatting for messages such as warnings.

FOUNDATION INTEGRATION LEVELS

- CSS: To use Foundation CSS, automatic or custom CSS packages can be downloaded from the download page and installed in the appropriate web server folders. The base is then embedded in the pages of the HTML page.
- SASS: Foundation SASS installation uses Ruby, Node.js, and Git to install Foundation resources. The base then provided a visual command line to edit and merge the source into CSS for use in HTML page extensions.
- Rails: The Foundation Rails gem can be added by adding “foundation ‘rails’ to the Rails Application Gemfile.

CHAPTER SUMMARY

The entire chapter is about Foundation we have learned that Foundation can be used with CSS. It can be added just by installing or adding CDN in your HTML file and you can start working on it. Also you will learn coding concept of it how to implement its coding.

FURTHER READING

Foundation - Quick Guide - https://www.tutorialspoint.com/foundation/foundation_quick_guide.htm, accessed on (2022 May 25).

Foundation (framework) - [https://en.wikipedia.org/wiki/Foundation_\(framework\)](https://en.wikipedia.org/wiki/Foundation_(framework)), accessed on (2022 May 25).

Foundation Overview - <https://developer.apple.com/documentation/foundation>, accessed on (2022 May 25).

Put Yourself On the Cutting Edge of Web Design - <https://zurb.com/university/foundation-intro>, accessed on (2022 May 25).

Responsive design gets a whole lot faster - <https://get.foundation/>, accessed on (2022 May 25).

Run locally - <https://github.com/foundation/foundation-sites>, accessed on (2022 May 25).

Skeleton

IN THIS CHAPTER

- Introduction
- Versions of Skeleton
- Skeleton installation
- Usage of Skeleton

In the Chapter 5, we talked about the most famous CSS framework, i.e., Foundation, but now, we will discuss another CSS framework, i.e., Skeleton.

INTRODUCTION

Skeleton is a lightweight, responsive, and customized CSS library built with a cell phone in mind. No merging or installation is required, and it works well with additional CSS style sheets and frames. Use the skeleton to get fast, efficient, and suitable CSS for demos or MVPs. And because it is <12 kb in the minified version, it is also ideal if speed is very important. Remember, the Skeleton, at its core, is the skeletal structure of pure parts and style. With just one link, you can turn dull HTML into a full-page page without the need to sacrifice speed or memorize a classroom library.

Tired of boring HTML, and long CSS forever? Combine Skeleton, an additional, lightweight CSS for every boilerplate code, you have already installed as the starting point for any project. The boilerplate code contains

any repetitive code that appears over and over again to get the result that seems to have to be very simple. And a simple, basic HTML style should be very simple.

Many previous style libraries or frames work on setting styles in classes that require the user to remember class names in order to add style to HTML objects. Skeleton works by applying most of the style directly to HTML tags. Tables, Forms, Lists, Buttons, you name it, they write it.

Installation

It is a simple, responsive boilerplate to kick start any responsive project.

```
$ npm install --save skeleton-CSS
```

The download will include Skeleton's CSS, Normalize CSS as a reset, a sample favicon, and an index.html as a starting point.

FILE STRUCTURE

```
Skeleton/  
├── index.html  
├── CSS/  
│   ├── normalize.min.css  
│   └── skeleton.css  
└── images/  
    └── favicon.icon
```

It is lightweight and simple. It styles only raw HTML elements and provides a responsive grid. Nothing more.

- It has around 400 lines of CSS unminified and with comments.
- It's a point, not a UI framework.
- No compiling or installing...just vanilla CSS.

SUPPORTED BROWSER

- Latest Chrome
- Latest Firefox

- Latest Opera
- Latest Safari
- Latest IE

The above list is non-exhaustive. It works perfectly with almost all older versions of the browsers, though IE certainly has large degradation prior to IE9.

SKELETON – THE GRID¹

Skeleton is a simple CSS framework. Specifically, it has two CSS files: The popular `normalize.css` file and the `skeleton.css` file. The Skeleton grid system is used to create a responsive website so it is a very important part of Skeleton. It also provides basic HTML components such as buttons, lists, tables, and forms.

The grid is a 12-column grid but fluid with a maximum width of 960px, which shrinks the browser/device in smaller sizes. The maximum width can be changed with a single CSS line and all columns will resize accordingly. The syntax is simple and makes encoding much easier. Go ahead, resize the browser. Basically, in this case, the grid is divided in such a way that the resulting sum will be equal to 12. Grid view layout is shown below:

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
```

```

"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:10%;
    text-align: center;
  }

.columns{
  background-color:bisque;
  border:1px solid black
}
.column{
  background-color: aquamarine;
  border:1px solid black
}
</style>

<body>

<!-- .container is main centered wrapper -->
<div class="container">
<h1> Skeleton -The Grid </h1>
  <!-- columns should be the current immediate child
of a .row -->
  <div class="row">
    <div class="one column"> One </div>
    <div class="eleven columns"> Eleven </div>
  </div>

  <!-- you just use a number and class 'column' or
'columns' -->
  <div class="row">
    <div class="two columns"> Two </div>
    <div class="ten columns"> Ten </div>
  </div>

  <div class="row">
    <div class="three columns"> Three </div>

```

```

    <div class="nine columns"> Nine </div>
</div>

<div class="row">
    <div class="four columns"> Four </div>
    <div class="eight columns"> Eight </div>
</div>

<div class="row">
    <div class="five columns"> Five </div>
    <div class="seven columns"> Seven </div>
</div>
<div class="row">
    <div class="six columns"> Six </div>
    <div class="six columns"> Six </div>
</div>

<div class="row">
    <div class="seven columns"> Seven </div>
    <div class="five columns"> Six </div>
</div>

<div class="row">
    <div class="eight columns"> Eight </div>
    <div class="four columns"> Six </div>
</div>

<div class="row">
    <div class="nine columns"> Nine </div>
    <div class="three columns"> Three </div>
</div>

<div class="row">
    <div class="ten columns"> Ten </div>
    <div class="two columns"> Two </div>
</div>

<div class="row">
    <div class="eleven columns"> Eleven </div>
    <div class="one columns"> One </div>
</div>

```

```
<!-- there are a few short hand columns widths as
well -->
<div class="row">
  <div class="one-third column">1/3</div>
  <div class="two-thirds column">2/3</div>
</div>
<div class="row">
  <div class="one-half column">1/2</div>
  <div class="one-half column">1/2</div>
</div>

<div class="row">
  <div class="one-half column">1/2</div>
  <div class="one-half column">1/2</div>
</div>
</div>
</body>
</html>
```

Skeleton -The Grid



Skeleton – The Grid

TYPOGRAPHY²

It is all about style, proportions, and spacing. Good typography is not only imperative for aesthetic appeal, but it also improves site usability when text legibility and readability concepts are applied. Typography is also about emotion. It types are all set with the rems, font sizes and special relationships can be responsive sized based on a single `<html>` tags font-size proper. It never changes the `<html>` font size, but it is there in case the user need it for project. All measurements are still base 10 though, an `<h1>` with `5.0remfont-size` just means 50px.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" rel="stylesheet" />

    <!-- Compressed JavaScript -->
    <script type = "text/javascript"
src=
"https://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:10%;
        text-align: center;
    }
</style>

<body>
<h1> Skeleton -TYPOGRAPHY </h1>

<!-- .container is main centered wrapper -->
<div class="container">
    <!-- columns should be the current immediate child
of a .row -->

<!-- Standard Headings -->
<h1> Heading H1 </h1>
<h2> Heading H2 </h2>
<h3> Heading H3 </h3>
<h4> Heading H4 </h4>

```

```

<h5> Heading H5 </h5>
<h6> Heading H6 </h6>

<!-- Base type size -->
<p>The base type is 15px over 1.6 line height (24px)</p>

<!-- Other styled text tags -->
<strong> Bolder </strong>
<em> Italicized </em>
<a> Anchor (Link) </a>
<u> Underlined </u>

</div>
</body>
</html>

```

Skeleton -TYPOGRAPHY

Heading H1

Heading H2

Heading H3

Heading H4

Heading H5

Heading H6

The base type is 15px over 1.6 line height (24px)

Bolder *Italicized* Anchor (Link) Underlined

Skeleton – Typography

Typography Units Used in CSS Frameworks

CSS has several different units for expressing a length.

Absolute Lengths

These units are fixed and a length expressed in any of these appear as exactly that size. These units are not recommended for use on screen, because screen sizes vary so much. They can be used if the output medium is known, such as for print layout.

cm	centimeters
Mm	millimeters
In	inches (1in=96px=2.54 cm)
px	pixels (1px=1/96th of 1 in)
pt	points (1pt=1/72 of 1 in)
pc	picas (1pc=12 pt)

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href="http://cdnjs.
cloudflare.com/ajax/libs/skeleton/2.0.4/skeleton.min.
css" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:5%;
        text-align: center;
    }
    .cm1{ font-size: 1cm; }
    .cm2{ font-size: 1.5cm;}

    .mm1{ font-size: 10mm; }
    .mm2{ font-size: 15mm; }

    .im1{ font-size: 10in; }

```

```

.im2{ font-size: 15in; }

.px1{ font-size: 10px; }
.px2{ font-size: 15px; }

.pc1{ font-size: 10px; }
.pc2{ font-size: 15px; }

.pt1{ font-size: 10pt; }
.pt2{ font-size: 15pt; }

.container{
box-shadow: 0px 3px 3px 2px grey ;
  display: flex;
  justify-content: space-evenly;
  padding: 20px;
}
h3{
  width:200px
}
</style>

<body>
<h1> Absolute Lengths - Font Units </h1>

<div class="container">
  <h3>centimeters</h3>
  <div class="cm1"> This is 1cm</div>
  <div class="cm2"> This is 1.5cm </div>

</div>

<div class="container">
  <h3> millimetres </h3>
  <div class="mm1"> This is 10mm </div>
  <div class="mm2"> This is 15mm </div>
</div>

<div class="container">
  <h3> inches </h3>
  <div class="in1"> This is 1in </div>
  <div class="in2"> This is 0.5in </div>
</div>

```

```
<div class="container">
  <h3> pixels </h3>
  <div class="px1"> This is 1px </div>
  <div class="px2"> This is 0.5px </div>
</div>
```

```
<div class="container">
  <h3> points </h3>
  <div class="pt1"> This is 1pt </div>
  <div class="pt2"> This is 0.5pt </div>
</div>
```

```
<div class="container">
  <h3> picas </h3>
  <div class="pc1"> This is 1pc </div>
  <div class="pc2"> This is 0.5pc </div>
</div>
```

```
</body>
</html>
```

Absolute Lengths - Font Units

centimeters	This is 1cm	This is 1.5cm
millimeters	This is 10mm	This is 15mm
inches	This is 1in	This is 0.5in
pixels	This is 1px	This is 0.5px
points	This is 1pt	This is 0.5pt
picas	This is 1pc	This is 0.5pc

Absolute Lengths

RELATIVE LENGTHS

Its units specify a length relative to another length property. Its units scale better between different rendering medium.

Units are as follows:

em	It is relative to the font-size of the element (2em means two times the size of the current font)
ex	It is relative to the x-height of the current font (rarely used)
ch	It is relative to the width of the “0” (zero)
VW	It is relative to 1% of the width of the viewport
vh	It is relative to 1% of the height of the viewport
vmin	It is relative to 1% of viewport’s smaller dimension
vmax	It is relative to 1% of viewport’s larger dimension
%	It is relative to the parent element

BUTTONS³

It comes in two basic flavors in CSS Framework Skeleton. The standard `<button>` element is simple, whereas the `class.button-primary` button is vibrant and prominent. The most button styles are applied to a number of appropriate form elements but can also be arbitrarily attached to anchors with a `button` class.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" />
```

```

    <!-- Compressed JavaScript -->
    <script type = "text/javascript"
src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
  body{
    margin: 5%
  }
  .button-primary{
    color: #2980b9
  }
  .button-success{
    background: #27ae60;
  }
  .button-info{
    background-color: #3498db;
  }
  .button-warning{
    background-color: #f39c12;
  }
  .button-danger{
    background-color: #c0392b;
  }
  .container{
    display: grid;
    text-align: center;
    padding: 10px;
  }
  button{
    width: 250px;
  }
  h1{
    text-align: center;
  }

```

```
</style>
```

```
<body>
```

```
  <h1>Skeleton Buttons</h1>
```

```
  <div class="container text-center">
```

```
    <button class="button"> Default </button>
```

```
    <button class="button button-primary"> Primary </button>
```

```
    <button class="button button-success"> Success </button>
```

```
    <button class="button button-info"> Info </button>
```

```
    <button class="button button-warning"> Warning </button>
```

```
    <button class="button button-danger"> Danger </button>
```

```
    <br />
```

```
    <button class="button button-primary button-block"> button block </button>
```

```
  </div>
```

```
</body>
```

```
</html>
```

Skeleton Buttons



Skeleton – Buttons

There are two types of button available:

- Primary Buttons
- Standard Button

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/
skeleton/2.0.4/skeleton.min.css" />

    <!-- Compressed JavaScript -->
    <script type = "text/javascript"
s rc=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:5%
    }
    .button-primary{
        color:#2980b9
    }
    .button-success{
        background: #27ae60;
    }
    .button-info{
        background-color: #3498db;
    }
    .button-warning{
```

```

        background-color: #f39c12;
    }
    .button-danger{
        background-color: #c0392b;
    }
</style>

<body>

    <div class="container">
        <h1>Skeleton Buttons - Primary & Standard </h1>

        <a class="button" href="#">Anchor button</a>
        <button>Button element</button>
        <input type="submit" value="submit input">
        <input type="button" value="button input">

        <!-- Primary buttons -->
        <a class="button button-primary" href="#">Anchor
button</a>
        <button class="button-primary">Button element</
button>
        <input type="submit" value="submit input
class="button-primary" ">
        <input type="button" value="button input"
class="button-primary">
    </div>
    <!-- Standard buttons -->
</body>
</html>

```

Example:

Skeleton Buttons - Primary & Standard



Skeleton Buttons – Primary and Standard

FORM⁴

Generally, Forms can make page effectively. But CSS Framework Skeleton allows user to create different forms using different elements and styles. All inputs, select, buttons are normalized for a common min height cross-browser so inputs can be stacked or placed alongside each other.

Example:

```
<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->
    <link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.3/
skeleton.min.css" />

    <!-- Compressed JavaScript -->
    <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
    </script>
    <script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
    </script>
</head>
<style>
    body{
        margin:5%
    }
    .button-primary{
        color:#2980b9
    }
    .button-success{
        background: #27ae60;
    }
    .button-info{
        background-color: #3498db;
    }
    .button-warning{
```

```

        background-color: #f39c12;
    }
    .button-danger{
        background-color: #c0392b;
    }
</style>

<body>

    <div class="container">
        <!-- The form above looks like this -->
    <form>
        <h2> Skeleton Forms </h2>

        <div class="row">
            <div class="six columns">
                <label for="exampleEmailInput"> Your email</
label>
                <input type="email" placeholder="test@mailbox.
com" class="u-full-width" id="exampleEmailInput">
            </div>
            <div class="six columns">
                <label for="exampleRecipientInput"> Reason for
contacting </label>
                <select class="u-full-width"
id="exampleRecipientInput">
                    <option value="Option 1"> Option 1 </option>
                    <option value="Option 2"> Option 2 </option>
                    <option value="Option 3"> Option 3 </option>
                </select>
            </div>
        </div>
        <label for="exampleMessage"> Message </label>
        <textarea class="u-full-width" placeholder="Write
your Messages" id="exampleMessage"></textarea>
        <label class="example-send-yourself-copy">
            <input type="checkbox">
            <span class="label-body"> Terms & Conditions </
span>
        </label>
        <input class="button-primary" type="submit"
value="Submit">

```

```

</form>
  </div>
<!-- Standard buttons -->
</body>
</html>

```

Skeleton Forms

Your email

Reason for contacting

Option 1 ▾

Message

Write your Messages

☐ Terms & Conditions

SUBMIT

Skeleton – Form

LISTS⁵

It behaves like other text for the most part, but there are some CSS properties specific to lists that you need to know about, as well as some best practices to consider. This article explains all.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Compressed CSS -->
  <link href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" rel="stylesheet" />

  <!-- Compressed JavaScript -->
  <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
  </script>
  <script src=

```

```

"HTTPS://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:5%
  }
  h2{
    text-align: center;
  }
</style>

<body>

  <div class="container">
    <h2 >Skeleton Lists </h2>
</form>

  <h4>Unordered List</h4>
  <ul>
    <li> Item 1 </li>
    <li>
      Item 2
      <ul>
        <li> Item 2.1 </li>
        <li> Item 2.2 </li>
      </ul>
    </li>
    <li> Item 3 </li>
  </ul>

  <h4>Ordered List</h4>
  <ol>
    <li> Item 1 </li>
    <li>
      Item 2
      <ul>
        <li> Item 2.1 </li>
        <li> Item 2.2 </li>
      </ul>
    </li>
  </ol>

```

```

    <li> Item 3 </li>
  </ol>

</div>

<!-- Standard buttons -->
</body>
</html>

```

Skeleton Lists

Unordered List

- Item 1
- Item 2
 - Item 2.1
 - Item 2.2
- Item 3

Ordered List

1. Item 1
2. Item 2
 - Item 2.1
 - Item 2.2
3. Item 3

Skeleton – Lists

CODE⁶

It is used to keep basic styling, you can just wrap anything in a `<code>` and it will appear like this. For blocks of code, wrap a `<code>` with a `<pre>`.

Example:

```

<!DOCTYPE html>
<html>

<head>

    <!-- Compressed CSS -->

```

```

<link rel="stylesheet" href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" />

<!-- Compressed JavaScript -->
<script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
</script>
<script src=
"HTTPS://cdn.jsdelivrivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
</script>
</head>
<style>
  body{
    margin:15%
  }

  .some-class {
    padding:30px;
    background-color: red;
  }
</style>

<body>

  <h1>Skeleton - Code</h1>
  <div class="container">
    <code class="some-class "> This is how we can use
Code tag in Skeleton.</code>
  </div>
</body>
</html>

```

Skeleton - Code

This is how we can use Code tag in skeleton.

Skeleton – Code

Benefits of Skeleton CSS Framework

- Very simple, with about 400 lines of code.
- It has a mobile-first approach and has designs that can be used as a starting point, not as a full frame.
- Style for a few HTML objects and includes Grid support. It should be used for small projects where the use of a broad framework is not available.
- Because of its minimal footprint, it is fast.

Disadvantages of Skeleton CSS Framework

- The skeletal structure should be considered only for use in small projects; in terms of improved performance, there will be a major effort from the developer side to use Skeleton as the only CSS library in the project.
- Small companies such as Steelkiwi Inc., Spirit Pixels use Skeleton in their technology stack.

CHAPTER SUMMARY

The entire chapter is about Skeleton we have learned that Skeleton can be used with CSS to enhance its. Skeleton can make it possible to do so. It can be added just by installing or adding CDN in your HTML file. You also find advantages and disadvantage.

FURTHER READING

A dead simple, responsive boilerplate - <http://getskeleton.com/>, accessed on (2022 May 26).

Getting Started with Skeleton, the Simple CSS Boilerplate - <https://www.sitepoint.com/getting-started-with-skeleton-simple-css-boilerplate/>, accessed on (2022 May 26).

Skeleton — The Lightweight CSS Framework - <https://medium.com/@jithamolns/skeleton-the-light-weight-css-framework-2f5e73aca563>, accessed on (2022 May 26).

Skeleton Block - <https://framework7.io/docs/skeleton>, accessed on (2022 May 26).

Skeleton Block - <https://medium.com/@jithamolns/skeleton-the-light-weight-css-framework-2f5e73aca563>, accessed on (2022 May 26).

SKELETON FRAMEWORK TUTORIALS - <https://ieatcss.com/skeleton-tutorial.html>, accessed on (2022 May 26).

NOTES

- 1 The Grid – Skeleton.
- 2 The Typography – Skeleton.
- 3 The Buttons – Skeleton.
- 4 The Forms – Skeleton.
- 5 The List – Skeleton.
- 6 The Code – Skeleton.

Milligram

IN THIS CHAPTER

- ▶ Introduction
- ▶ Milligram installation
- ▶ Usage of Milligram

In the previous chapter, we talked about Skeleton, and now, we will discuss another CSS Framework, Milligram.

INTRODUCTION

It provides a setup of styles for a fast and clean starting point, and it's not about a UI framework. It is designed for better performance and higher productivity with properties to reset resulting in cleaner code.

MILLIGRAM INSTALLATION

Install with Bower

```
$ bower install milligram
```

Install with npm

```
$ npm install milligram
```

Install with Yarn

```
$ yarn add milligram
```

Once downloaded, unzip the compressed folder to see the main file in the uncompressed and minified version.

```
milligram
  dist
milligram.css
milligram.min.css
example.html
license
readme.md
```

First, use method mentioned above to download Milligram. Then, add these tags in the head. Milligram is also available via CDN.

```
<!-- Google Fonts -->
<link href="https://fonts.googleapis.com/css?family=R
oboto:300,300italic,700,700italic" rel="stylesheet">

<!-- CSS Link For normalize-->
<link href="http://cdnjs.cloudflare.com/ajax/libs/
normalize/8.0.1/normalize.css" rel="stylesheet" >

<!-- CSS -->
<link href="http://cdnjs.cloudflare.com/ajax/libs/
milligram/1.4.1/milligram.css" rel="stylesheet" >
```

CLI

It offers a simple boilerplate project with Milligram CSS.

```
$ npm install -g milligram-cli
```

Create a new application using the command `milligram init`, then install dependencies, and run with `npm start`.

```
$ milligram init newapp
$ cd newapp
$ npm start
```

Typography Units Used in CSS Frameworks

CSS has various different units for expressing a length.

Absolute Lengths

The absolute length units are fixed and a length expressed in any of these appear as exactly that size. These units are not recommended for use on screen, because screen sizes vary so much. They can be used if the output is known, such as for print layout.

cm	centimeters
mm	millimeters
in	inches (1 in=96px=2.54 cm)
px	pixels (1 px=1/96th of 1 in)
pt	points (1 pt=1/72 of 1 in)
pc	picas (1 pc=12 pt)

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Compressed CSS -->
  <link href=
"http://cdnjs.cloudflare.com/ajax/libs/skeleton/2.0.4/
skeleton.min.css" rel="stylesheet" />

  <!-- Compressed JavaScript -->
  <script src=
"http://cdnjs.cloudflare.com/ajax/libs/jquery/3.2.1/
jquery.min.js">
  </script>
  <script src=
"https://cdn.jsdelivr.net/npm/foundation-sites@6.7.4/
dist/js/foundation.min.js">
  </script>
</head>
<style>
  body{
    margin:5%;
    text-align: center;
  }
```

```

.cm1{ font-size: 1cm; }
.cm2{ font-size: 1.5cm;}

.mm1{ font-size: 10mm; }
.mm2{ font-size: 15mm; }

.im1{ font-size: 10in; }
.im2{ font-size: 15in; }

.px1{ font-size: 10px; }
.px2{ font-size: 15px; }

.pc1{ font-size: 10px; }
.pc2{ font-size: 15px; }

.pt1{ font-size: 10pt; }
.pt2{ font-size: 15pt; }

.container{
box-shadow: 0px 3px 3px 2px grey ;
  display: flex;
  justify-content: space-evenly;
  padding: 20px;
}
h3{
  width:200px
}
</style>

<body>
<h1> Absolute Lengths - Font Units </h1>

<div class="container">
  <h3>centimeters</h3>
    <div class="cm1"> This is 1cm</div>
    <div class="cm2"> This is 1.5cm </div>

</div>

<div class="container">
  <h3> millimetres </h3>
    <div class="mm1"> This is 10mm </div>
    <div class="mm2"> This is 15mm </div>
</div>

```

```
<div class="container">
  <h3> inches </h3>
  <div class="in1"> This is 1in </div>
  <div class="in2"> This is 0.5in </div>
</div>
```

```
<div class="container">
  <h3> pixels </h3>
  <div class="px1"> This is 1px </div>
  <div class="px2"> This is 0.5px </div>
</div>
```

```
<div class="container">
  <h3> points </h3>
  <div class="pt1"> This is 1pt </div>
  <div class="pt2"> This is 0.5pt </div>
</div>
```

```
<div class="container">
  <h3> picas </h3>
  <div class="pc1"> This is 1pc </div>
  <div class="pc2"> This is 0.5pc </div>
</div>
```

```
</body>
</html>
```

Absolute Lengths - Font Units

centimeters	This is 1cm	This is 1.5cm
millimeters	This is 10mm	This is 15mm
inches	This is 1in	This is 0.5in
pixels	This is 1px	This is 0.5px
points	This is 1pt	This is 0.5pt
picas	This is 1pc	This is 0.5pc

Absolute Lengths

RELATIVE LENGTHS

Relative length units define a length relative to another length property. These units scale better between different rendering mediums.

Units are as follows:

em	It is relative to font-size of the element (2em means two times the size of the current font)
ex	It is relative to the x-height of the current font (rarely used)
ch	It is relative to the width of the “0” (zero)
vw	It is relative to 1% of the width of the viewport
vh	It is relative to 1% of the height of the viewport
vmin	It is relative to 1% of viewport’s smaller dimension
vmax	It is relative to 1% of viewport’s larger dimension
%	It is relative to the parent element

TYPOGRAPHY¹

CSS3 introduced several new units, including the rem unit, representing “root em”. The rem unit is related to the html element font size of the root.

That means we can define one font size in the root segment, and define all rem units as a percentage. Typography has a font size defined at 1.6 rem (16 px) and line length at 1.6 (24 px). Milligram uses the Roboto family font, developed by Christian Robertson, and provided by Google.

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin: 15%
  }
</style>

<body>
  <div class="container">
    <h1> Milligram - Typography </h1>

    <!-- Base font-size and line-height -->
    <p> The base type is 1.6rem (16px) over 1.6 line height (24px) </p>

    <!-- Other elements to text markup -->
    <a> Anchor </a>
    <em> Emphasis </em>
```

```
<small> Small </small>
<strong> Strong </strong>
<u> Underline </u>
```

```
<!-- Default Headings -->
<h1> Heading H1 </h1>
<h2> Heading H2 </h2>
<h3> Heading H3 </h3>
<h4> Heading H4 </h4>
<h5> Heading H5 </h5>
<h6> Heading H6 </h6>
  </div>
</body>
</html>
</div>
```

Milligram - Typography

The base type is 1.6rem (16px) over 1.6 line height (24px)

Anchor *Emphasis* small **Strong** Underline

Heading

Heading

Heading

Heading

Heading

Heading

Milligram – Typography

BLOCKQUOTES²

The Blockquote represents that is quoted from another source.

Example:

```
<!DOCTYPE html>
<html>

<head>
```

```

<!-- Google Fonts -->
<link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

<!-- CSS Reset -->
<link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

<!-- CSS -->
<link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:15%
  }

</style>

<body>
  <div class="container">
    <h1> Milligram - Blockquotes </h1>
    <blockquote>
      <p><em> You write any of your favourite quotes.
</em></p>
    </blockquote>

  </div>
</body>
</html>
</div>

```

Milligram - Blockquotes

You write any of your favourite quotes.

BUTTONS³

Button styles can be applied to a number of various form elements but can also be attached to anchors with a `button` class. The Button, an essential part of user experience. Buttons consists in three basic styles in Milligram: The button element has color by default, whereas `.button-outline` has an outline around, and `.button-clear` is entirely clear.

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:15%
  }
</style>

<body>
  <div class="container">
    <!-- Default Button -->
    <h1> Milligram - Button </h1>
    <a class="button" href="#">Default Button</a>

    <!-- Outlined Button -->
    <button class="button button-outline">Outlined Button</button>
```

```

    <!-- Clear Button -->
    <input class="button button-clear" type="submit"
value="Clear Button">

    <!-- Easily apply the `.button` for button style
in the anchor element. -->

</div>
</body>
</html>
</div>

```

Milligram - Button



Milligram – Button

```

<!DOCTYPE html>
<html>

<head>

    <!-- Google Fonts CSS-->
    <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

    <!-- CSS Reset -->
    <link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css">

    <!-- CSS -->
    <link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet"
rel="stylesheet" >

</head>
<style>
    body{
        margin:10%;
        font-size: 24px;
        text-align: center;

```

```

    }
    .button-black {
      background-color: black;
      border-color: black;
    }
    .button-black.button-clear,
    .button-black.button-outline {
      background-color: transparent;
      color: black;
    }
    .button-black.button-clear {
      border-color: transparent;
    }

    /* Custom size */
    .button-small {
      font-size: .8rem;
      height: 2.8rem;
      line-height: 2.8rem;
      padding: 0 1.5rem;
    }

    .button-large {
      font-size: 1.4rem;
      height: 4.5rem;
      line-height: 4.5rem;
      padding: 0 2rem;
    }
  }
</style>

<body>
  <div class="container">
    <h1> Milligram - Extending The Inheritances Buttons
  </h1>
    <button class="button-black"> Button Black </button>
    <button class=".button-black button-clear "> Button-
black Button-clear  </button>
    <button class="button-small"> Button Smalll </
button>
    <button class="button-large"> Button Large </button>
    <button class="button-black button-outline ">
Button-black Button-outline </button>
  </div>

```

```

</body>
</html>
</div>

```

Milligram - Extending The Inheritances Buttons



Milligram – Extending the Inheritances Buttons

LISTS⁴

It behaves like other text for the most part, but there are some CSS properties specific to lists that need to know about, as well as some best practices to consider.

The List is a very common way to display items. It has three types of lists: The unordered list uses `<ul>` tags marked with a circles, the ordered list uses `ol` element and items will be marked as numbers, and the description list uses `dl` element and items will not be marking, only spacings.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{

```

```

        margin:15%
    }
</style>

<body>
    <div class="container">
        <h1> Milligram - List</h1>
        <!-- Default Button -->
        <!-- Unordered list -->
    <ul>
        <li>Unordered list item 1</li>
        <li>Unordered list item 2</li>
    </ul>

    <!-- Ordered list -->
    <ol>
        <li>Ordered list item 1</li>
        <li>Ordered list item 2</li>
    </ol>

    <!-- Description list -->
    <dl>
        <dt>Description list item 1</dt>
        <dd>Description list item 1.1</dd>
    </dl>

    </div>
</body>
</html>
</div>

```

Milligram - List

- Unordered list item 1
- Unordered list item 2

1. Ordered list item 1
2. Ordered list item 2

Description list item 1

Description list item 1.1

Milligram – List

FORMS⁵

Generally, Forms can make page effectively. But CSS Framework Milligram allows user to create different forms using different elements and styles. All inputs, select, and buttons are normalized for a height cross-browser so inputs can be placed alongside each other.

The Form has never been exactly, and it can be downright on a mobile device with its on-screen keyboard. It helps to make this much easier with design focused on the user experience.

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts Link-->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:15%
  }
</style>

<body>
  <div class="container">
    <h1> Milligram - Forms </h1>
    <form>
      <fieldset>
        <label for="nameField"> Name </label>
```

```

        <input type="text" placeholder="John Swift"
id="nameField">
        <label for="ageRangeField"> Age Range </label>
        <select id="ageRangeField">
            <option value="0-13"> 0-13 </option>
            <option value="14-17"> 14-17 </option>
            <option value="18-23"> 18-23 </option>
            <option value="24+"> 24+ </option>
        </select>
        <label for="commentField"> Comment</label>
        <textarea placeholder="Write your message
here" id="commentField"></textarea>
        <div class="float-right">
            <input type="checkbox" id="confirmField">
            <label class="label-inline"
for="confirmField">Send a copy to yourself</label>
        </div>
        <input class="button-primary" type="submit"
value="Send">
    </fieldset>
</form>
</div>
</body>
</html>
</div>

```

Milligram - Forms

Milligram – Form

TABLES⁶

Its element represents data in two dimensions. We can use of proper formatting with `<thead>` and `<tbody>` to create a table. The code becomes cleaner without disturbing understanding.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts CSS -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:15%
  }
</style>

<body>
  <div class="container">
    <h1> Milligram - Table </h1>
    <table>
      <thead>
        <tr>
          <th>A</th>
          <th>B</th>
          <th>C</th>
          <th>D</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>W</td>
          <td>X</td>
          <td>Y</td>

```

```
        <td>Z</td>
      </tr>
      <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
        <td>4</td>
      </tr>
    </tbody>
  </table>
</div>
</body>
</html>
</div>
```

Milligram - Table

A	B	C	D
W	X	Y	Z
1	2	3	4

Milligram – Table

GRIDS⁷

The grid is a liquid system with a diameter of 112.0 rem (1120 px), which shrinks the browser/device in smaller sizes. The maximum width can be changed with a single CSS line and all columns will resize accordingly. Milligram is different from most due to its use of the standard CSS Flexible Box Layout Layout Module. The advantage is simplicity, and we know that effective code is very important in maintaining and balancing. Just add the columns you want in sequence, and they will take up the available space equally. If you want three columns, add three columns; if you want five columns, add five columns. There is no limit to the number of columns, but we advise you to follow the design pattern of the grid system. See more tips for good habits in tips.

Example:

```
<!DOCTYPE html>
<html>
```

```

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:15%
  }
  .column{
    border:1px solid black;
  }

</style>

<body>
  <div class="container">
    <h1> Milligram - Grid System </h1>
    <div class="row">
      <div class="column"> .column </div>
      <div class="column"> .column </div>
      <div class="column"> .column </div>
      <div class="column"> .column </div>
    </div>
    <br>
    <div class="row">
      <div class="column"> .column </div>
      <div class="column column-50 column-offset-25">
        .column column-50 column-offset-25 </div>
      </div>
    </div>
  </body>

```

Milligram - Grid System

.column	.column	.column	.column
.column	.column column-50 column-offset-25		

Milligram – Grid System

```
</html>
</div>
```

CODE⁸

The Code element represents a fragment of code. It just wraps anything in a code and it will appear like this. If you need a block default, enter the code within the `<pre>` element.

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=R
oboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/
normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.
com/ajax/libs/milligram/1.4.1/milligram.css">

</head>
<style>
  body{
    margin:15%
  }
  .milligram {
    color: #9b4dca;
  }
}
```

```

</style>

<body>
  <div class="container">
    <h1>Milligram - Code </h1>
    <code>
      .milligram {
        color: #9b4dca;
      }
    </code>
  </div>
</body>
</html>
</div>

```

Milligram - Code

```
.milligram { color: #9b4dca; }
```

Milligram - Code

UTILITIES⁹

Milligram Framework has some functional classes to improve the performance and productivity.

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts CSS -->
  <link href="https://fonts.googleapis.com/css?family=R
oboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/
normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->

```

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/
milligram/1.4.1/milligram.css" rel="stylesheet" >
```

```
</head>
<style>
  body{
    margin:10%;
    font-size: 24px;
    text-align: center;
  }
  .left{
    border:1px solid brown;
  }
  .right a{
    border:1px solid black;
    padding-left: 18px;
  }
</style>

<body>
  <div class="container">
    <!-- Functional Classes -->
    <h1> Milligram - Utilities </h1>
    <!-- Clear any float -->
    <div class="clearfix">

      <!-- Float either directions -->
      <div class="float-left left">Brand Logo
      </div>
      <div class="float-right right" >
        <ul>
          <a> Home </a>
          <a> About </a>
          <a> Contact</a>
        </ul>
      </div>
    </div>
  </div>
</body>
</html>
</div>
```

Milligram - Utilities

Brand Logo

Home

About

Contact

Milligram – Utilities

COLUMN WIDTHS

The Grid system also supports specifying column widths as a ratio of the parent row's width. It can append one of the percentages listed below to a column-class to achieve the desired layout.

Example:

```
<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->
  <link href="https://fonts.googleapis.com/css?family=Roboto:300,300italic,700,700italic" rel="stylesheet">

  <!-- CSS Reset -->
  <link href="http://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.css" rel="stylesheet" >

  <!-- CSS -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/milligram/1.4.1/milligram.css" rel="stylesheet" >

</head>
<style>
  body{
    margin:10%;
    font-size: 24px;
    text-align: center;
  }
  .column {
    border: 1px solid black;
  }
</style>
```

```

<body>
  <div class="container">
    <h1> Milligram - Utilities </h1>
    <div class="row">
      <div class="column column-10"> 10% </div>
      <div class="column column-90"> 90% </div>

    </div>
    <div class="row">
      <div class="column column-20"> 20% </div>
      <div class="column column-80"> 80% </div>

    </div>
    <div class="row">
      <div class="column column-25"> 25% </div>
      <div class="column column-75"> 75% </div>
    </div>
    <div class="row">
      <!-- 3 column width */ -->
      <div class="column column-33"> 33% </div>
      <div class="column column-33"> 33% </div>

      <div class="column column-33"> 33% </div>
    </div>
    <div class="row">
      <div class="column column-40"> 40% </div>
      <div class="column column-60"> 60% </div>
    </div>
    <div class="row">
      <div class="column column-50"> 50% </div>
      <div class="column column-50"> 50% </div>

    </div>
    <div class="row">
      <div class="column column-60"> 60% </div>
      <div class="column column-40"> 40% </div>

    </div>
    <div class="row">
      <div class="column column-67"> 67% </div>
      <div class="column column-37"> 37% </div>

```

```

</div>
<div class="row">
  <div class="column column-75"> 75% </div>
  <div class="column column-25"> 25% </div>
</div>
<div class="row">
  <div class="column column-80"> 80% </div>
  <div class="column column-20"> 20% </div>
</div>
<div class="row">
  <div class="column column-90"> 90% </div>
  <div class="column column-10"> 10% </div>
</div>
<div class="row">
  <div class="column">100%</div>
</div>
</div>
</body>
</html>
</div>

```

RESPONSIVENESS

The Mobile First is the design strategy that takes priority development for mobile devices like smartphones and tablets. It means all styles outside of a media queries apply to all devices, and then larger screens are targeted for enhancement. This prevents small devices from having to parse tons of unused CSS. Milligram uses some breakpoints like these:

- Larger than Mobile screen: 40.0 rem (640 px)
- Larger than Tablet screen: 80.0 rem (1280 px)
- Larger than Desktop screen: 120.0 rem (1920 px)

Example:

```

<!DOCTYPE html>
<html>

<head>

  <!-- Google Fonts -->

```

```
<link rel="stylesheet" href="https://fonts.googleapis.
com/css?family=Roboto:300,300italic,700,700italic">
```

```
<!-- CSS Reset -->
```

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/
normalize/8.0.1/normalize.css" rel="stylesheet" >
```

```
<!-- CSS -->
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.
com/ajax/libs/milligram/1.4.1/milligram.css">
```

```
</head>
```

```
<style>
```

```
  body{
    margin:10%;
    font-size: 24px;
    text-align: center;
  }
```

```
/* Mobile First Media Queries */
```

```
/* Base style */
```

```
.container{
  background-color: o;
  color: brown;
}
```

```
/* Larger than mobile screen */
```

```
@media (min-width: 40.0rem) {  .container{
  background-color: pink;
  color: black;
} }
```

```
/* Larger than tablet screen */
```

```
@media (min-width: 80.0rem) {
  .container{
    background-color: aquamarine;
    color: brown;
  } }
```

```
/* Larger than desktop screen */
```

```
@media (min-width: 120.0rem) {
  .container{
```

```

        background-color: salmon;
        color: brown;
    }
}
</style>

<body>
  <div class="container">
    <h1> Milligram - Responsiveness Screen </h1>
  </div>
</body>
</html>
</div>

```

REASONS TO USE MILLIGRAM

- Small CSS frame: Milligram is easy to set up and get started with. Although it offers powerful features to improve productivity, it comes with a very low weight of 2 KB when pressed.
- No idea: Unlike other frames, Milligram does not come with a default style. You will not need to reset or remove structures that do not comply with your terms when using your custom styles.
- Easy to read: Milligram is so easy to read during the day. Reading the official texts is more than enough to get you started.

WHY NOT TO USE?

- No templates: If you want something pre-made or similar to a template, it means Milligram is not yours. But if you want to use a specific design, you can greatly improve your product.
- Small community: Milligram has a small but strong community. Finding community support will not be as easy as the well-known CSS frameworks, but Milligram's simplicity means you probably won't need much help anyway.

CHAPTER SUMMARY

Here, we covered CSS framework that is Milligram, which can be used with CSS to enhance its features. Also, we covered how to add it just by installing or adding CDN in your html file with its advantages and disadvantage.

FURTHER READING

A minimalist CSS framework - <https://cdnjs.com/libraries/milligram>, accessed on (2022 May 27).

A Minimalist CSS Framework - <https://gitpiper.com/resources/css/framework-sart/milligram-a-minimalist-css-framework>, accessed on (2022 May 27).

CSS Framework Sample - <https://www.tohoho-web.com/ex/css-framework/milligram.html>, accessed on (2022 May 27).

Getting Started - <https://www.npmjs.com/package/milligram-less>, accessed on (2022 May 27).

Lightweight CSS Frameworks - <https://dev.to/sm0ke/lightweight-css-framework-a-curated-list-4hc3>, accessed on (2022 May 27).

Milligram - Minimalist CSS Framework - <https://www.drupal.org/project/milligram>, accessed on (2022 May 27).

Who's using Milligram - <https://milligram.io/>, accessed on (2022 May 27).

Why it's awesome - <https://github.com/milligram/milligram>, accessed on (2022 May 27).

NOTES

- 1 Typography – Milligram.
- 2 Blockquotes – Milligram.
- 3 Buttons – Milligram.
- 4 List – Milligram.
- 5 Forms – Milligram.
- 6 Tables – Milligram.
- 7 Grids – Milligram.
- 8 Code – Milligram.
- 9 Utilities – Milligram.

Pure CSS

IN THIS CHAPTER

- Introduction
- Pure installation
- Usage of Pure CSS concepts

In the last chapter, we covered Milligram Framework with all its features, but now, our next framework is Pure CSS.

INTRODUCTION

It is a CSS framework. It is a set of free tools and open source for building responsive websites and web applications. Yahoo has improved this, which is used to make websites faster, better, and more responsive. It is an active form of Bootstrap. Pure CSS is built with keeping the design responsive in mind. So, we get pre-built reaction styles that are always the same in all forums.

Pure is a Cascading Sheet (CSS) designed by Yahoo. It helps in making faster, more beautiful, and responsive websites. It is extremely space savvy and is very tiny in size comparable to 4 KB. This section explains all the fundamental concepts of Pure.

Why Pure CSS?

- Designs responsive webpages for mobile devices.
- It is a free framework and open source available at <https://purecss.io/>.

- No dependence on JavaScript and its library.
- It also supports shadows and colors.
- Another great alternative to Bootstrap.
- It's a Quick and Easy Way to Web Development.

INSTALLING PURE CSS

There are two methods to use Pure:

- Locally installation: You can download the file on locally and include it in your HTML code.
- CDN version: You can include the its file into HTML code directly from the Content Delivery Network (CDN).

There are two ways we can apply code:

1. Using NPM install: You can also add Pure CSS to the project at npm. This is our recommended way to combine styles in the construction process of your project with a series of tools. To use this command, first of all, you need to enter a node.

```
$ npm install purecss --save
```

2. Using CDN: You can add Pure CSS to web page via the free CDN link. Just add the following `<link>` element into page `<head>` section, before your project's style sheets.

```
<link rel="stylesheet" href="https://unpkg.com/purecss@2.0.5/build/pure-min.css" integrity="sha384-LTIdEidl25h2dPxrB2Ekgc9c7sEC3CWGM6HeFmuDNUjX76Ert4Z4IY714dhZHPLd" crossorigin="anonymous">
```

Example: The following example shows a dropdown menu and a drop-down with background colors created by using the background-color property of CSS.

Example:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link rel="stylesheet" href=
    "HTTPS://unpkg.com/purecss@2.0.5/build/pure-min.css"
    integrity=
    "sha384-LTIDeidl25h2dPxRB2Ekgc9c7sEC3CWGM6HeFmuDNUjX76
    Ert4Z4IY714dhZHPLd"
    crossorigin="anonymous">
</head>

<style>
  body{
    margin: 10%;
    font-size: 24px;
  }
  .container{
    text-align: center;
  }
</style>

<body>
<div class="container">
  <h2 style="color:green;">
    Pure CSS - Simple dropdown menu
  </h2>

  <div class="pure-menu pure-menu-horizontal">
    <ul class="pure-menu-list">
      <li class="pure-menu-item pure-menu-selected">
        <a href="#" class="pure-menu-link"> Home </
a>
      </li>
      <li class="pure-menu-item pure-menu-has-children
        pure-menu-allow-hover">
        <a href="#" class="pure-menu-link"> Blog </a>
        <ul class="pure-menu-children">
          <li class="pure-menu-item">

```

```

        <a href="#" class="pure-menu-link">
            Notes </a>
    </li>
    <li class="pure-menu-item">
        <a href="#" class="pure-menu-link">
            Tutorial </a>
    </li>
</ul>
</li>
<li class="pure-menu-item pure-menu-has-children
pure-menu-allow-hover"
    style="background-color: green;">

    <a href="#" class="pure-menu-link"
        style="color:white;">
        Contact US
    </a>

    <ul class="pure-menu-children">
        <li class="pure-menu-item">
            <a href="#" class="pure-menu-link">
                Email
            </a>
        </li>
        <li class="pure-menu-item">
            <a href="#" class="pure-menu-link">
                Google Form
            </a>
        </li>
    </ul>
</li>
</ul>
</div>
</div>
</body>
</html>

```

Pure CSS - Simple dropdown menu

[Home](#)
[Blog](#)
[Contact US](#)

Pure CSS – Simple Dropdown Menu

OVERVIEW

Pure Cascading Style Sheet (CSS) created by Yahoo. It helps to create fast, beautiful, and responsive websites.

Some of its notable features are the following:

- Built-in responsive design
- Normal CSS with a minimal footprint
- A set of small, responsive CSS modules
- It's free to use

RESPONSIVE DESIGN

Pure has a responsive design built in so that a website created using Pure will redesign itself according to device size. Pure has a 12-grid mobile-first fluid that supports responsive classes of small, large, and medium screen sizes. Pure classes are designed in such a way that the website can fit any screen size. Websites created using Pure are compatible with PCs, tablets, and mobile devices.

Standard CSS

Pure uses only standard CSS and is very easy to learn. There is no reliance on any external JavaScript library like jQuery.

Extensible

Pure is a very small and flat design. It is built with the fact that it is much easier to add new CSS rules than to write over existing CSS rules. By adding a few lines of CSS, Pure look can be customized to work with an existing web project. It supports bright shadows and colors. The colors and shades remain the same across the various platforms and devices. And most importantly, it's completely free to use.

PURE CSS MENU RESPONSIVE CSS MENU¹

The Horizontal-to-Scrollable menu is a type of responsive menu bar in Pure.CSS, which converts the horizontal menu bar into a horizontal scroll bar. Used to make the menubar responsive when website width increases or decreases from one device to another, for example, Tablets and Smartphones. Pure.CSS Horizontal-to-Scrollable menu does not require any JavaScript functions, and CSS Categories built are enough.

Classes are mentioned below and described classes are required to create this layout:

- **Pure-menu:** It is the default that generates the horizontal menu. It can contain the menu items and headings in the form of a list.
- **Pure-menu-horizontal:** It is added to the default vertical “pure-menu” to make it a horizontal menu.
- **Pure-menu-list:** It is for the unordered list that contains the menu items. The list items inside the list must have the class “pure-menu-item”.
- **Pure-menu-item:** It is added to the list items inside this list.
- **Pure-menu-heading:** It is added for the headings inside or outside the menu list. By default, it capitalizes the text inside.
- **Pure-menu-scrollable:** It makes the pure-menu-horizontal be scrolled or flicked when the width of the website is not fit to the menu size.
- **Pure-menu-link:** It is added to the links inside the menu items.

Syntax:

```
<div class="pure-menu pure-menu-horizontal
pure-menu-scrollable">

</div>
```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link rel="stylesheet" href= "HTTPS://unpkg.com/
purecss@2.0.5/build/pure-min.css">
```

```

<link
    href=
"HTTPS://unpkg.com/purecss@1.0.0/build/pure-min.css"
rel="stylesheet"
    integrity=
"sha384-nn4HPE8lTHyVtFCBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
    crossorigin="anonymous">
</head>

```

```

<style>
    .grids-example {
        background: RGB(250, 250, 250);
        margin: 2em auto;
        font-family: Consolas, 'Liberation Mono',
Courier, monospace;
        text-align: center;
    }

```

```

    .graybox {
        background: rgb(240, 240, 240);
        border: 1px solid #ddd;
    }

```

```

</style>

```

```

</head>

```

```

<body>

```

```

<div class = "grids-example">

```

```

    <h1> Pure CSS - Responsive Horizontal-to-Scrollable
Menu Layout

```

```

    </h1>

```

```

    <div class = "pure-g">

```

```

        <div class = "pure-u-1">

```

```

            <div class = "graybox">

```

```

                <p >This column is to occupy the complete
space of a row. </p>

```

```

            </div>

```

```

        </div>

```

```

    </div>

```

```

</div>

```

```

<div class = "grids-example">

```

```

<div class = "pure-g">
  <div class = "pure-u-2-5">
    <div class = "graybox">
      <p> This column is to occupy the two-fifth
of the space of a row. </p>
    </div>
  </div>
</div>
</div>

```

```

<div class = "grids-example">
  <div class = "pure-g">
    <div class = "pure-u-3-5">
      <div class = "graybox">
        <p> This column is to occupy the three-
fifth of the space of a row. </p>
      </div>
    </div>
  </div>
</div>

```

```

<div class = "grids-example">
  <div class = "pure-g">
    <div class = "pure-u-1-3">
      <div class = "graybox">
        <p> Column 1: This column is to occupy the
one-third of the
        space of a row on all devices. </p>
      </div>
    </div>
  </div>

```

```

    <div class = "pure-u-1-3">
      <div class = "graybox">
        <p> Column 2: This column is to occupy the
one-third of the space
        of a row on all devices. </p>
      </div>
    </div>
  </div>

```

```

    <div class = "pure-u-1-3">
      <div class = "graybox">

```

```

        <p> Column 3: This column is to occupy the
one-third of the space of a
        row on all devices. </p>
    </div>
</div>
</div>
</div>
</body>
</html>

```

Pure CSS - Responsive Horizontal-to-Scrollable Menu Layout		
This column is to occupy the complete space of a row.		
This column is to occupy the two-fifth of the space of a row.		
This column is to occupy the three-fifth of the space of a row.		
Column 1: This column is to occupy the one-third of the space of a row on all devices.	Column 2: This column is to occupy the one-third of the space of a row on all devices.	Column 3: This column is to occupy the one-third of the space of a row on all devices.

Pure - Responsive Design

Another Example:

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content=
        "width=device-width, initial-scale=1">
    <link rel="stylesheet" href= "HTTPS://unpkg.com/
pure-css@2.0.5/build/pure-min.css">
    <link
        href=
"HTTPS://unpkg.com/purecss@1.0.0/build/pure-min.css"
        rel="stylesheet" integrity=
"sha384-nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
        crossorigin="anonymous">
</head>

<style>
    .menubar {
        background-color:brown;
        position: relative;
    }

```

```

.menu {
    display: inline-block;
    width: auto;
    vertical-align: middle;
}
.menu a {
    color: WhiteSmoke;
}
.menu a:hover {
    background-color: white;
    color: green;
}
.website_name {
    display: block;
    text-align: center;
    position: relative;
}
body{
    background-color: antique white;
    background-repeat: no-repeat;
    background-position: center;
    background-attachment: fixed;
}
/* To toggle the menu bar to
Horizontal-scrollable menubar
when the window size reduced to particular size*/
@media (max-width: 72em)
{

    .scrollable_links
    {
        position: absolute;
        width: 100%;
        background-color: transparent;
        background-color: chocolate;
    }
    .menu
    {
        display: block;
    }
    .menubar_top

```

```

        {
            position: relative;
            padding: 0.5em;
        }
    }
</style>
</head>
<body>

<div class="menubar">
    <div class="pure-menu menu menubar_top">
        <a href="#" class="pure-menu-heading
website_name">
            Brand LOGO
        </a>
    </div>
    <div class="pure-menu pure-menu-horizontal
pure-menu-scrollable
            menu scrollable_links">
        <ul class="pure-menu-list">
            <li class="pure-menu-item">
                <a href="#" class="pure-menu-link">
                    Home
                </a>
            </li>
            <li class="pure-menu-item">
                <a href="#" class="pure-menu-link">
                    About Us
                </a>
            </li>
            <li class="pure-menu-item">
                <a href="#" class="pure-menu-link">
                    Blog
                </a>
            </li>
            <li class="pure-menu-item">
                <a href="#" class="pure-menu-link">
                    Services
                </a>
            </li>
            <li class="pure-menu-item">

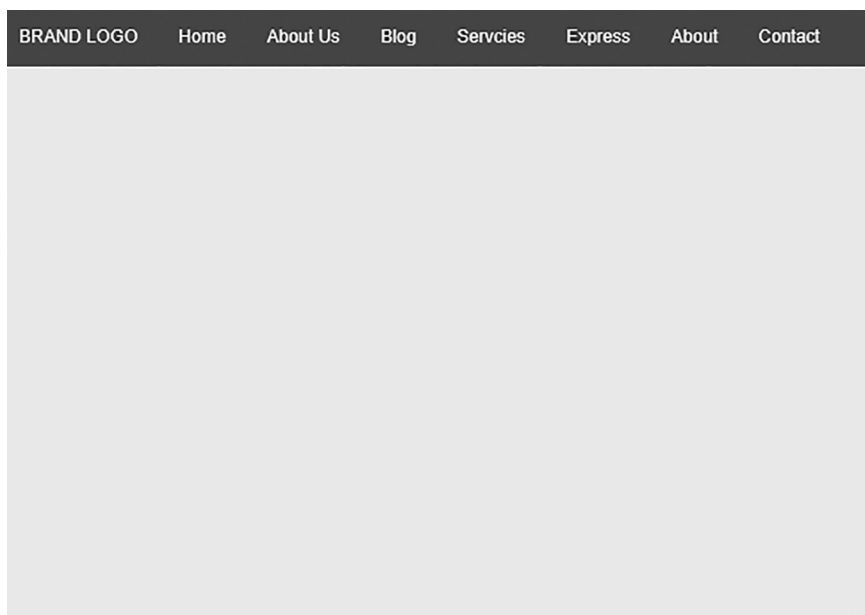
```

```

        <a href="#" class="pure-menu-link">
            Express
        </a>
    </li>

    <li class="pure-menu-item">
        <a href="#" class="pure-menu-link">
            About
        </a>
    </li>
    <li class="pure-menu-item">
        <a href="#" class="pure-menu-link">
            Contact
        </a>
    </li>
</ul>
</div>
</div>
</body>
</html>

```



PURE CSS FOUNDATION

When creating a collaborative website, there may be things that need to be automatically hidden or by a specific user action. In this section, we can see how you hide things in Pure CSS.

Pure CSS Hiding Elements

- **hidden:** This section is used to hide features while supporting older browsers such as Internet Explorer.

Include Pure CSS Base Module: Encryption comes with a basic Pure CSS module. So to use the encryption utility, we need to install the basic module in our HTML file. A link to install the basic module is provided below:

```
<link href="HTTPS://unpkg.com/purecss@1.0.1/build/
base-min.css" rel="stylesheet" >
```

Syntax:

1. Using the hidden attribute:

```
<element hidden> ... </element>
```

2. Using the hidden class:

```
<element class="hidden"> ... </element>
```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link rel="stylesheet" href= "HTTPS://unpkg.com/
purecss@2.0.5/build/pure-min.css">
  <link
    href=
```

```

"HTTPS://unpkg.com/purecss@1.0.1/build/pure-min.css"
rel="stylesheet"
    integrity=
"sha384-nn4HPE8lTHyVtFCBiyW9d20Fjt8BJwUXyWZT9InLYax14R
DjBj46LmSztkmNP9w"
    crossorigin="anonymous">
<style>
    body{
        margin:10%;
        text-align: center;
    }
    h2{
        margin-bottom: 10px;
    }
    h3{
        margin-top: 0px;
    }
    .header{
        margin-bottom: 30px;
    }
</style>
</head>

<body>
    <div class="header">
        <h3> Pure CSS Hiding Elements using
            <i>hidden </i>class
        </h3>
    </div>
    <!-- This will be not hidden -->
    <label>Input without hidden class:</label>
    <input type="text" name="name"
        placeholder="Enter Name">

    <br>
    <!-- This will be hidden -->
    <label>Input with hidden class:</label>

    <input type="text" name="name"
        placeholder="Enter Name" class="hidden">

    <!-- -->
</body>
</html>

```

Pure CSS Hiding Elements using *hidden* class

Input without hidden class:

Input with hidden class:

Pure – Hidden Attribute**PURE CSS IMAGES**

Pure CSS is a CSS framework. A collection of free and open tool for creating responsive websites and web applications. Pure CSS is developed by Yahoo and is used to create fast, beautiful, and responsive websites. It is used as an alternative to Bootstrap.

Pure CSS provides a pure-img class that makes the image responsive. This section can be used in the object to create a scale with the view hole. This is useful if you are building responsive websites.

Rounded Corners Image: The border-radius property is used to create a rounded corner image.

Syntax:

```

```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link rel="stylesheet" href= "HTTPS://unpkg.com/
purecss@2.0.5/build/pure-min.css">
  <link rel="stylesheet"
    href=
"HTTPS://unpkg.com/purecss@1.1.0/build/pure-min.css"
    integrity=
"sha384-nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
    crossorigin="anonymous">
  <style>
```

```

    body{
        margin:10%;
        text-align: center;
    }
    h2{
        margin-bottom: 10px;
    }
    h3{
        margin-top: 0px;
    }
    .header{
        margin-bottom: 30px;
    }
    img{
        border-radius: 12%;
    }
</style>
</head>

<body>
    <!DOCTYPE html>
    <html lang="en">

    <head>
        <title> PureCSS </title>
        <meta charset="utf-8">
        <meta name="viewport" content=
            "width=device-width, initial-scale=1">
        <link rel="stylesheet" href=
            "HTTPS://unpkg.com/purecss@2.0.5/build/pure-min.css"
            integrity=
            "sha384-LTIDeidl25h2dPxrb2Ekgc9c7sEC3CWGM6HeFmuDNUjX
76Ert4Z4IY714dhZHPLd"
            crossorigin="anonymous">
    </head>

    <body>
        <center>
            <div style="width: 500px">
                <h1>Responsive Image</h1>

```

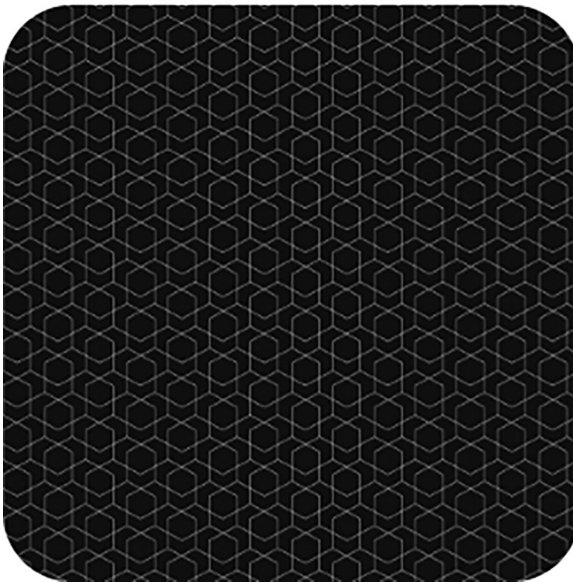
```

        
    </div>
</center>
</body>

</html>
</body>
</html>

```

Responsive Image



Pure – Image Responsive

PURE CSS BUTTONS

Buttons are one of the common UI elements. They made it possible for others to interact and take action by making selections. We can create types of buttons with the help of Pure CSS.

- Primary buttons
- Active buttons
- Default Buttons

- Disabled buttons
- Customizing buttons

Default Buttons:

You can add the “pure-button” class name to any `<a>` or `<button>` element to create a default button.

Syntax:

```
// Pure CSS - Link Button
<a class="pure-button" href="#">
  Pure Button1
</a>

// Pure CSS - Normal Button
<button class="pure-button">
  Pure- Button2
</button>
```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <linkhref= "HTTPS://unpkg.com/purecss@2.0.5/build/
pure-min.css" rel="stylesheet" >
  <link rel="stylesheet"
    href=
"HTTPS://unpkg.com/purecss@1.0.0/build/pure-min.css"
    integrity=
"sha384-nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
    crossorigin="anonymous">
  <style>
    body{
      margin:10%;
```

```

        text-align: center;
    }

</style>
</head>

<body>
    <!--Default buttons-->
    <h1>Pure CSS Default Buttons</h1>
    <br><br>
    <!-- Link Button -->
    <a class="pure-button" href="#">Pure Button1</a>

    <!-- Normal Button -->
    <button class="pure-button">Pure Button2</button>
</body>
</html>

```

Pure CSS Default Buttons

Pure Button1

Pure Button2

Pure – Buttons

Disabled Buttons: To mark a button as disabled add “pure-button-disabled” with class pure-button. You can use the disabled attribute directly.

Syntax:

```

<button class="pure-button pure-button-disabled">
Disabled Button1
</button>
<button class="pure-button" disabled>
Disabled Button2
</button>

```

Example:

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link href= "HTTPS://unpkg.com/purecss@2.0.5/
build/pure-min.css" rel="stylesheet">
  <link rel="stylesheet"
    href=
    "HTTPS://unpkg.com/purecss@1.0.1/build/pure-min.css"
    integrity=
    ""
    crossorigin="anonymous">
  <style>
    body{
      margin:10%;
      text-align: center;
    }

  </style>
</head>

<body>
  <!--Default buttons-->
  <h1>Pure CSS Disabled Buttons</h1>
  <br><br>
  <!-- Link Button -->
  <a class="pure-button pure-button-disabled"
href="#">Pure Button1</a>

  <!-- Normal Button -->
  <button class="pure-button" disabled="">Pure
Button2</button>
</body>
</html>

```

Pure CSS Disabled Buttons

Pure Button1

Pure Button2

Pure – Disabled Buttons

Customized Buttons: With the help of Pure, it is easy to customize buttons for your own application because of its minimal styling. Group custom CSS into a class such as `button-success`, which you can then add to an element that has a `pure-button` class.

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content=
    "width=device-width, initial-scale=1">
  <link rel="stylesheet" href= "HTTPS://unpkg.com/
purecss@2.0.5/build/pure-min.css">
  <link rel="stylesheet"
    href=
"HTTPS://unpkg.com/purecss@1.0.0/build/pure-min.css"
    integrity=
"sha384-nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
    crossorigin="anonymous">
  <style>
    body{
      margin:10%;
      text-align: center;
    }

  </style>
</head>
```

```

<body>
  <style>
    /* Styling for Customized Buttons */

    .button-warning,
    .button-
secondary ,
    .button-success,
    .button-error,{
      color: white;
      border-radius: 14px;
      text-shadow: 1px 2px 2px rgba(0, 0, 0, 0.2);
    }
    .button-error {
      background: rgb(202, 60, 60);
    }

    .button-success {
      background: rgb(28, 184, 65);
    }

    .button-warning {
      background: rgb(223, 117, 20);
    }

    .button-secondary {
      background: rgb(66, 184, 221);
    }
  </style>
</head>

<body style="text-align: center">
  <!--Customized buttons-->
  <h1>Pure CSS Customized Buttons </h1>
  <br /><br />
  <button class="button-success pure-button">
    Success Button
  </button>

  <button class="button-secondary pure-button">
    Secondary Button
  </button>

```

```

<button class="button-warning pure-button">
    Warning Button
</button>

<button class="button-error pure-button">
    Error Button
</button>
</body>
</html>
</body>
</html>

```

Pure CSS Customized Buttons



Pure – Customized Buttons

BUTTONS SIZES

Buttons are one of the common UI elements. They made it for users to interact with a system and take action by making selections. Sometimes we required various size buttons, there it comes, with the help of the Pure CSS Button Size class. Pure CSS Button Sizes class: There is no predefined class for button size we have to do that by using CSS. Button Size percentage depends on the parent:

- `button-xsmall`: It is used to create extra small buttons (70%).
- `button-small`: It is used to create small buttons (85%).
- `pure-button`: It is used to create normal buttons.
- `button-large`: It is used to create large buttons (110%).
- `button-xlarge`: It is used to create extra-large buttons (125%).

Syntax:

```
<button class="button-xsmall pure-button">
    .....
</button>
```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content=
        "width=device-width, initial-scale=1">
    <link rel="stylesheet" href= "HTTPS://unpkg.com/
purecss@2.0.5/build/pure-min.css">
    <link rel="stylesheet"
        href=
"HTTPS://unpkg.com/purecss@1.0.0/build/pure-min.css"
        integrity=
"sha384-nn4HPE8lTHyVtfcBi5yW9d20FjT8BJwUXyWZT9InLYax14
RDjBj46LmSztkmNP9w"
        crossorigin="anonymous">
    <style>

        body{
            margin:10%
        }
        .button-xsmall {
            font-size: 70%;
            background-color: aqua;
        }

        .button-small {
            font-size: 85%;
            background-color: tan;
        }
    </style>
```

```

.button-large {
    font-size: 110%;
    background-color: palegreen;
}

.button-xlarge {
    font-size: 125%;
    background-color: aquamarine;
}
</style>
</head>

<body>
  <center>
    <h1>Pure - Button Sizes</h1>
    <br><br>

    <button class="button-xsmall pure-button
                button-">Pure Button2</button>
    <button class="button-small pure-button
                button-">Pure Button2</button>
    <button class="pure-button
                button-">Pure Button2</button>
    <button class="button-large pure-button
                button-">Pure Button2</button>
    <button class="button-xlarge pure-button
                button-">Pure Button2</button>

  </center>
</body>
</html>

```

Pure - Button Sizes

Pure Button2

Pure Button2

Pure Button2

Pure Button2

Pure Button2

Pure – Button Sizes

PURE CSS TABLES

Before we start with Pure, we need to know the basics of transparent CSS. Basically, the Pure CSS is a Cascading Style Sheet framework designed by Yahoo. The main reason for inventing Pure CSS is used to develop responsive and functional websites like Bootstrap, which is also compatible with mobile devices and is an open source and easy-to-use CSS framework. It would be one of the best ways to create Materialize CSS. In this lesson, we will understand what the Pure CSS Table is and learn how to use it in our project.

As we consider creating web pages, we know that tables are a great and easy way to organize a lot of data on websites into rows and columns. Using the Pure CSS framework, we will be able to design different types of tables.

Pure CSS provides a number of useful classes for style tables, there are five major classes as listed below:

- **pure-table:** This section is used to style a table with default pads and assign a border to table elements with a highlighted theme.
- **pure-table-bordered:** This section is used to draw borders on a straight and horizontal table across all table cells.
- **pure-table-horizontal:** This section is used to draw a table with only horizontal lines.
- **pure-table-odd:** This section is used to create a linear table with a zebra effect that is more visible and attractive to users.
- **pure-table-striped:** Basically used to display tables with stripes. Automatically clear the table after passing the `<table>` element next to the clean table section.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Import Pure CSS -->
    <link rel="stylesheet"
          href="HTTPS://unpkg.com/purecss@2.0.6/build/
pure-min.css"
```

```

        integrity="sha384-Uu6IeWbM+gzNVXJcM9XV3SohHt
mWE+3VGi496jvgX1jyvDTXfdK+rfZc8C1Aehk5"
        crossorigin="anonymous"
        origin="anonymous"
    />

    <!-- Used to optimized Website for mobile -->
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />

    <style>
        .container {
            margin-top: 20px;
            display: flex;
            align-items: center;
            justify-content: center;
            flex-direction: column;
        }
        h1 {
            color: green;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Pure Tables</h1>
        <h2> Default Table</h2>

        <!-- Here pure-table class is used -->
        <table class="pure-table">
            <thead>
                <tr>
                    <th>Rank</th>
                    <th>Name</th>
                </tr>
            </thead>

            <tbody>
                <tr>
                    <td>1</td>
                    <td>Rohit</td>

```

```

        </tr>

        <tr>
            <td>2</td>
            <td>Virat</td>

        </tr>

        <tr>
            <td>3</td>
            <td>Rahul</td>

        <tr>
            <td>4</td>
            <td>Rishabh</td>

        </tr>
    </tbody>
</table>

<br> <br>
<h2> Bordered Table </h2>
    <table class="pure-table pure-table-bordered">
        <thead>
            <tr>
                <th>Number</th>
                <th>Name</th>
                <th>Age</th>

            </tr>
        </thead>

        <tbody>
            <tr>
                <td>1</td>
                <td>Rohit</td>
                <td>14</td>

            </tr>

            <tr>

```

```

        <td>2</td>
        <td>Virat</td>
        <td>12</td>

</tr>

<tr>
    <td>3</td>
    <td>Rahul</td>
    <td>18</td>

</tr>

<tr>
    <td>4</td>
    <td>Rishabh</td>
    <td>16</td>

</tr>
</tbody>
</table>
<br> <br>

<h2> Table with Horizontal Borders </h2>
    <table class="pure-table
pure-table-horizontal">
        <thead>
            <tr>
                <th>Number</th>
                <th>Name</th>
                <th>Age</th>

            </tr>
        </thead>

        <tbody>
            <tr>
                <td>1</td>
                <td>Rohit</td>
                <td>14</td>

            </tr>

```

```

        <tr>
            <td>2</td>
            <td>Virat</td>
            <td>12</td>

        </tr>

        <tr>
            <td>3</td>
            <td>Rahul</td>
            <td>18</td>

        </tr>

        <tr>
            <td>4</td>
            <td>Rishabh</td>
            <td>16</td>

        </tr>
    </tbody>
</table>
<br> <br>

<h2> Striped Table</h2>

<table class="pure-table">
    <thead>
        <tr>
            <th>Number</th>
            <th>Name</th>
            <th>Age</th>

        </tr>
    </thead>

    <tbody>
        <tr>
            <td>1</td>
            <td>Rohit</td>
            <td>14</td>

```

```
</tr>
```

```
<tr class="pure-table-odd">
  <td>2</td>
  <td>Virat</td>
  <td>12</td>
```

Pure Tables

Default Table

Rank	Name
1	Rohit
2	Virat
3	Rahul
4	Rishabh

Bordered Table

Number	Name	Age
1	Rohit	14
2	Virat	12
3	Rahul	18
4	Rishabh	16

Table with Horizontal Borders

Number	Name	Age
1	Rohit	14
2	Virat	12
3	Rahul	18
4	Rishabh	16

Striped Table

Number	Name	Age
1	Rohit	14
2	Virat	12
3	Rahul	18
4	Rishabh	16

Pure – Table

```

        </tr>

        <tr>
            <td>3</td>
            <td>Rahul</td>
            <td>18</td>

        </tr>

        <tr class="pure-table-odd">
            <td>4</td>
            <td>Rishabh</td>
            <td>16</td>

        </tr>
    </tbody>
</table>
</div>
</body>
</html>

```

CHAPTER SUMMARY

Here, we covered CSS framework that is Pure. It told you the way you can add it just by installing or adding CDN in your HTML file with its advantages and disadvantages.

FURTHER READING

Features of Pure.CSS - <https://www.javatpoint.com/what-is-pure-css>, accessed on (2022 May 28).

Get Started - <https://www.npmjs.com/package/purecss>, accessed on (2022 May 28).

Pure CSS - <https://www.geeksforgeeks.org/pure-css/>, accessed on (2022 May 28).

Pure.CSS - <https://purecss.io/>, accessed on (2022 May 28).

Pure.CSS Tutorial - <https://www.tutorialspoint.com/purecss/index.htm>, accessed on (2022 May 28).

NOTE

- 1 Pure CSS Responsive Horizontal-to-Scrollable Menu Layout – Geeksforgeeks.

Materialize

IN THIS CHAPTER

- Introduction
- Materialize installation
- Usage of Materialize

In the previous chapter, we talked about the most famous CSS framework, i.e., Pure, but now, we will discuss another CSS framework, Materialize.

INTRODUCTION

Materialize CSS is a design language that combines the basic principles of design as well as innovation and technology. It is created and designed by Google itself. Google is to create a design system that allows for integrated user information across all of its products in any platform.

WHY USE MATERIALIZE CSS?

- It is easy to use and makes the web page dynamic and responsive.
- Creates a webpage compatible with mobile phones, tablets, and laptops.
- It is available for free.

INSTALLING MATERIALIZE CSS

Method 1

Go to the Official Materialize Documents <https://materializecss.com/getting-started.html> and download the latest version available. Then, add the downloaded `materialize.min.js` file and the `materialize.min.css` file to your project directory.

Method 2

You can also include the `materialize.min.js` and `materialize.min.css` files into your HTML code file directly from the Content Delivery Network (that is CDN). `cdnjs.cloudflare.com` provides you content for the latest version.

You can use CDN link to run the code without installing it. Include the following CDN links inside the head tag.

```
<!-- Compiled and minified CSS -->
<link href="https://cdnjs.cloudflare.com/ajax/libs/
materialize/1.0.0/css/materialize.min.css"
rel="stylesheet" >

<!-- Compiled and minified JavaScript -->
<script type = "text/javascript"
src="http://cdnjs.cloudflare.com/ajax/libs/
materialize/1.0.0/js/materialize.min.js"></script>
```

Method 3

Install Materialize using npm. Before installing it, first install `node.js` and `npm` and then run the following command:

```
$ npm install materialize-css@next
```

Method 4

You can install Materialize using Bower package manager. Use the following command to install it.

```
$ bower install materialize
```

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <meta initial-scale = 1" name = "viewport"
content = "width = device-width,>
    <link
      href = "htrel = "stylesheet"
      http://fonts.googleapis.com/
icon?family=Material+Icons"
      rel = "stylesheet" >
    <link
      href = "http://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css"
      rel = "stylesheet">
    <script src = "https://code.jquery.com/jquery-
2.1.1.min.js" type = "text/javascript">
    </script>
    <script type = "text/javascript"
      src = "http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js">
    </script>
  </head>
  <body>
    <div class = "card-panel pink lighten-
2"><h3>Materialize CSS </h3></div>
  </body>
</html>
</html>

```

Materialize CSS

Materialize CSS

MATERIALIZE COLOR

It can support a rich set of color classes. These color classes are inspired and designed by considering the colors used in marketing, road signs, and sticky notes.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <meta name = "viewport" content = "width =
device-width, initial-scale = 1">
    <link
      href = "http://fonts.googleapis.com/
icon?family=Material+Icons"
      rel = "stylesheet" >
    <link
      href = "http://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css" rel =
"stylesheet">
    <script src = "https://code.jquery.com/jquery-
2.1.1.min.js" type = "text/javascript">
    </script>
    <script src = "http://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/js/materialize.min.js">
    </script>
    <style>
```

```

        .container{
            display: flex;
            justify-content: space-evenly;
            flex-wrap: 1;
        }
        .card-panel{
            width: 300px;
        }
        h1{
            text-align: center;
        }
    </style>
</head>

<body>
    <h1> Materialize Color - Lightness/Darkness
classes</h1>
    <div class="container">
        <div class = "card-panel red lighten-1"><h4>
Red </h4></div>
        <div class = "card-panel pink lighten-2"><h4>
Pink </h4></div>
        <div class = "card-panel purple lighten-
3"><h4> Purple </h4></div>
        <div class = "card-panel deep-purple lighten-
4"><h4> Deep Purple </h4></div>
        <div class = "card-panel indigo lighten-
5"><h4>Indigo </h4></div>
        <div class = "card-panel blue darken-1"><h4>
Blue </h4></div>
        <div class = "card-panel light-blue darken-
2"><h4> Light Blue </h4></div>
        <div class = "card-panel cyan darken-3"><h4>
Cyan </h4></div>
        <div class = "card-panel teal darken-4"><h4>
Teal </h4></div>
    </div>
    <div class="container">
        <div class = "card-panel light-green accent-1
"><h4> Light Green </h4></div>
        <div class = "card-panel lime accent-
2"><h4>Lime </h4></div>

```

```

    <div class = "card-panel orange accent-3
"><h4> Orange </h4></div>
    <div class = "card-panel deep-orange accent-4
"><h4> Deep Orange </h4></div>
  </div>
</body>
</html>

```

Materialize Color



Materialize Color

Here, we have a complete list of lightness/darkness classes, which can be used to vary the color applied.

- lighten-1
- lighten-2
- lighten-3
- lighten-4
- lighten-5
- darken-1
- darken-2
- darken-3
- darken-4
- accent-1
- accent-2
- accent-3
- accent-4

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <link
      href = "http://fonts.googleapis.com/
icon?family=Material+Icons" rel = "stylesheet">
    <meta name = "viewport" content = "width =
device-width, initial-scale = 1">

    <link rel = "stylesheet"
      href = "http://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css">

    <script type = "text/javascript"
src = "http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js">
    </script>
    <script type = "text/javascript" src = "http://code.
jquery.com/jquery-2.1.1.min.js">
    </script>

    <style>
      .container{
        display: flex;
        justify-content: space-evenly;
        flex-wrap: 1;
      }
      .card-panel{
        width: 300px;
      }
      h1{
        text-align: center;
      }
    </style>
  </head>

  <body>
    <h1> Materialize Color </h1>
    <div class="container">

```

```

        <div class = "card-panel red lighten-1"><h4>
Red </h4></div>
        <div class = "card-panel pink lighten-2"><h4>
Pink </h4></div>
        <div class = "card-panel purple lighten-
3"><h4> Purple </h4></div>
        <div class = "card-panel deep-purple lighten-
4"><h4> Deep Purple </h4></div>
        <div class = "card-panel indigo lighten-
5"><h4>Indigo </h4></div>
        <div class = "card-panel blue darken-1"><h4>
Blue </h4></div>
        <div class = "card-panel light-blue darken-
2"><h4> Light Blue </h4></div>
        <div class = "card-panel cyan darken-3"><h4>
Cyan </h4></div>
        <div class = "card-panel teal darken-4"><h4>
Teal </h4></div>
    </div>
    <div class="container">
        <div class = "card-panel light-green accent-1
"><h4> Light Green </h4></div>
        <div class = "card-panel lime accent-
2"><h4>Lime </h4></div>
        <div class = "card-panel orange accent-3
"><h4> Orange </h4></div>
        <div class = "card-panel deep-orange accent-4
"><h4> Deep Orange </h4></div>
    </div>
</body>
</html>

```

Materialize Color - Lightness/Darkness classes



Materialize Color – Lightness/Darkness Classes

MATERIALIZED CSS GRIDS

Materialize provides a 12 column-fluid responsive grid system. It uses the row and column styles classes to define rows and columns, respectively.

- Row: It is used to define a padding-less container to be used for responsive columns. It is mandatory for responsive classes to be fully responsive.
- Col: It is used to specify a column with sub-classes.

COLUMNS FOR SMALL SCREEN DEVICES

Here is the list of column-level styles for small screen devices, typically smartphones.

- s1: It is used to describe 1 of 12 columns with width as 08.33%.
- s2: It is used to describe 2 of 12 columns with width as 16.66%.
- s3: It is used to describe 3 of 12 columns with width as 25.00%.
- s4: It is used to describe 4 of 12 columns with width as 33.33%. It can contain s4, s5, s6, s7, s8, s9, s10, and s11.
- s12: It is used to describe the 12 of 12 columns with width as 100%. The default class for small screen phones.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <meta initial-scale = 1" name = "viewport"
content = "width = device-width"

    <link
      href = "http
://cdnjs.cloudflare.com/ajax/libs/materialize/0.97.3/
css/materialize.min.css" rel = "stylesheet">
    <link rel = "stylesheet"
```

```

        href = "http://fonts.googleapis.com/
icon?family=Material+Icons"    >
        <script src = "https://code.jquery.com/jquery-
2.1.1.min.js" type = "text/javascript" >
        </script>
        <script type = "text/javascript"
src = "http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js">
        </script>
        <style>
        .row{
            margin:10%;
        }
        .container{
            display: flex;
            justify-content: space-evenly;
            flex-wrap: 1;
        }
        .card-panel{
            width: 300px;
        }
        h1{
            text-align: center;
        }
        .col{
            height:100px;
        }
    </style>
</head>

<body>

    <div class = "row">
        <h1> Materialize CSS - The Grid System (for
small)</h1>
        <div class = "col s3 red center">1</div>
        <div class = "col s2 pink center">2</div>
        <div class = "col s4 blue center">3</div>
        <div class = "col m1 orange center">4</div>
        <div class = "col m4 grey center">5</div>
        <div class = "col m1 light-blue center">6</
div>

```

```


## Materialize CSS - The Grid System (for small)



### Materialize CSS – The Grid System (for Small)



#### COLUMNS FOR MEDIUM SIZE SCREEN DEVICES



Here is the list of column-level styles for medium screen devices, i.e., tablets:



- m1: It is used to describe 1 of 12 columns with width as 08.33%.
- m2: It is used to describe 2 of 12 columns with width as 16.66%.
- m3: It is used to describe 3 of 12 columns with width as 25.00%.
- m4: It is used to describe 4 of 12 columns with width as 33.33%.
- m5- m11.
- m12: It is used to describe 12 of 12 columns with width as 100%. It is a default class for medium screen phones.



#### Example:



```

<!DOCTYPE html>
<html>
 <head>
 <title>The Materialize Example</title>

```


```

```
<meta name = "viewport" content = "width =
device-width, initial-scale = 1">
```

```
<link rel = "stylesheet"
      href = "https://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css">
<link rel = "stylesheet"
      href = "http://fonts.googleapis.com/
icon?family=Material+Icons">
```

```
</script>
<script type = "text/javascript" src = "http://
cdnjs.cloudflare.com/ajax/libs/materialize/0.97.3/js/
materialize.min.js">
```

```
</script>
<style>
.row{
    margin:10%;
}
.container{
    display: flex;
    justify-content: space-evenly;
    flex-wrap: 1;
}
.card-panel{
    width: 300px;
}
h1{
    text-align: center;
}
.col{
    height:100px;
}
</style>
</head>
```

```
<body>
```

```
<div class = "row">
  <h1> Materialize CSS - The Grid System (for
medium)</h1>
```

```

        <div class = "col m1 red center">1</div>
        <div class = "col m1 pink center">2</div>
        <div class = "col m1 blue center">3</div>
        <div class = "col m1 orange center">4</div>
        <div class = "col m3 grey center">5</div>
        <div class = "col m1 light-blue center">6</
div>
        <div class = "col m1 teal center">7</div>
        <div class = "col m1 center purple">8</div>
        <div class = "col m1 center light-green">9</
div>
        <div class = "col m1 center lime ">10</div>
        <div class = "col m1 center teal ">11</div>
        <div class = "col m1 center cyan ">12</div>
    </div>
</body>
</html>

```

Materialize CSS - The Grid System (for medium)



Materialize CSS – The Grid System (for Medium)

COLUMNS FOR LARGE SIZE SCREEN DEVICES

Here you can see the list of column-level styles for large screen devices, i.e., laptops and desktops.

- 1l: It is used to describe 1 of 12 columns with width as 08.33%.
- 12: It is used to describe 2 of 12 columns with width as 16.66%.
- 13: It is used to describe 3 of 12 columns with width as 25.00%.
- 14: It is used to describe 4 of 12 columns with width as 33.33%.
- 112: It is used to describe 12 of 12 columns with width as 100%. It is default class for large screen devices.

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <meta initial-scale = 1", name = "viewport"
content = "width = device-width,>
<link

  href = "http
://cdnjs.cloudflare.com/ajax/libs/materialize/0.97.3/
css/materialize.min.css"  rel = "stylesheet"
  >
  <script  type = "text/javascript"
  src = "http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js">
    <link rel = "stylesheet"
      href = "http://fonts.googleapis.com/
icon?family=Material+Icons">

    <scriptsrc = "https://code.jquery.com/jquery-
2.1.1.min.js">
    </script>

  </script>
  <style>
    .row{
      margin:10%;
    }
    .container{
      display: flex;
      justify-content: space-evenly;
      flex-wrap: 1;
    }
    .card-panel{
      width: 300px;
    }
    h1{
      text-align: center;
    }
    .col{

```

```

        height:100px;
    }
</style>
</head>

<body>

    <div class = "row">
        <h1> Materialize CSS  - The Grid System (for
long)</h1>
        <div class = "col l3 red center">1</div>
        <div class = "col l2 pink center">2</div>
        <div class = "col l4 blue center">3</div>
        <div class = "col m1 orange center">4</div>
        <div class = "col m1 grey center">5</div>
        <div class = "col m1 light-blue center">6</
div>
        <div class = "col m1 teal center">7</div>
        <div class = "col m1 center purple">8</div>
        <div class = "col m1 center light-green">9</
div>
        <div class = "col m5 center lime ">10</div>
        <div class = "col m1 center teal ">11</div>
        <div class = "col m1 center cyan ">12</div>
    </div>
</body>
</html>

```

Materialize CSS - The Grid System (for long)



Materialize CSS – The Grid System (for Long)

MATERIALIZE CSS UTILITY CLASSES

There are various utility classes in Materialize which are used for designing. such as color utility classes – .red, .green, .grey, etc., alignment utility classes – .valign-wrapper, .left-align, .right-align, .center-align, .left, .right, hiding also, and content utility classes as per device size – .hide,

.hide-on-small-only, .hide-on-med-only, .hide-on-med-and-down, .hide-on-med-and-up, and .hide-on-large-only.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Example</title>
    <meta name = "viewport" content = "width =
device-width, initial-scale = 1">
    <script src = "https://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/js/materialize.min.js">
  <link rel = "stylesheet"
    href = "https://fonts.googleapis.com/
icon?family=Material+Icons">
    <script type = "text/javascript" src = "https://
code.jquery.com/jquery-2.1.1.min.js">
    </script>

    </script>
    <style>
      .row{
        margin:10%;
      }
      .container{
        display: flex;
        justify-content: space-evenly;
        flex-wrap: 1;
      }
    </style>
  </head>

  <body>

    <div class = "row">

      <h3> Materialize - Alignment Utilities </h3>
      <div class = "card-panel valign-wrapper">
        <p class = "valign">Vertically Aligned
Text</p>
      </div>
```

```

        <div class = "card-panel">
            <div> <p class = "left-align">
Left    Aligned Text </p> </div>
            <div> <p class = "right-align"> Right
Aligned Text </p> </div>
            <div> <p class = "center-align"> Center
Aligned Text </p> </div>
            </div>
        </div>
    </body>
</html>

```

Materialize - Alignment Utilities



Materialize – Alignment Utilities

MATERIALIZE CSS MEDIA

It specifies several classes to make images and videos responsive to different sizes.

- **Responsive-img:** Used to make an image to resize itself based on the screen size.
- **Video-container:** Used for responsive container having embedded videos.
- **Responsive-video:** Used to make HTML5 videos responsive.

Example:

```

<!DOCTYPE html>
<html>
  <head>

```

```

<title>The Materialize Example</title>
<meta name = "viewport" content = "width =
device-width, initial-scale = 1">
<link rel = "stylesheet"
href = "https://fonts.googleapis.com/
icon?family=Material+Icons">
<link rel = "stylesheet"
href = "https://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css">
<script src = "https://code.jquery.com/jquery-
2.1.1.min.js" type = "text/javascript"
>
</script>
<script type = "text/javascript"
src = "https://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js">
</script>
<style>
.row{
margin:10%;
}
.container{
display: flex;
justify-content: space-evenly;
flex-wrap: 1;
}

iframe{
width:540;
height:200
}
</style>
</head>

<body>

<div class = "row">

<h3>Responsive Embedded Video</h3>
<div class = "video-container">
<iframe src = "https://www.youtube.com/"

```

```

        frameborder = "0" allowfullscreen></
iframe>
    </div>

    <div class = "video-container">
        <video width = "300" height = "200"
controls>
        <source src = "movie.mp4" type = "video/
mp4" />
        Your browser does not support the video
element.
    </video>

    </div>
</body>
</html>

```

MATERIALIZE – TABLES

It can be used to display different types of tables using various styles over table.

- None: It represents a basic table without any border.
- Stripped: It displays a stripped table.
- Bordered: It can draw a table with a border across rows.
- Highlight: It can draw a highlighted table.
- Centered: It can draw a table with all the text center aligned in the table.
- Responsive-table: It draws a responsive table to show a horizontal scrollbar, if the screen is too small to display the content.

Example:

```

<!DOCTYPE html>
<html>
    <head>
        <title>The Materialize Example</title>
initial-scale=1"

```

```

    <meta name="viewport"
content="width=device-width>

    <link href="http
://cdnjs.cloudflare.com/ajax/libs/materialize/0.97.3/
css/materialize.min.css" rel="stylesheet">
    <script type = "text/javascript"
src="http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js"></script>
    <script type="text/javascript" src="http://
code.jquery.com/jquery-2.1.1.min.js"></script>
<link href="http://fonts.googleapis.com/
icon?family=Material+Icons" rel="stylesheet">

<style>
    div {
        width : 200px;
        height : 200px;
    }
</style>
</head>
<body class="container">
    <h2> Materialize - Tables </h2>
    <hr/>
    <h3> Simple Table </h3>
    <table>
    <thead>
        <tr>
            <th> Student </th>
            <th> >Class </th>
            <th> Grade </th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td> >Mahesh </td>
            <td> 5 </td>
            <td> A+ </td>
        </tr>
        <tr>
            <td> Rahul </td>
            ><td> 5 </td>

```

```

        <td> B </td>
    </tr>
    <tr>
        <td> Mohan </td>
        <td> 5 </td>
        <td> C </td>
    </tr>
</tbody>
</table>
<h3> Stripped Table </h3>
<table class="striped bordered">
<thead>
    <tr><th> Studen t</th><th> Class </th><th>
>Grade </th></tr>
</thead>
<tbody>
    <tr><td >Mahesh </td><td> 5 </td><td> A+ </
td></tr>
    <tr><td> Rahul </td><td> 5 </td><td> B </td></
tr>
    <tr><td> Mohan </td><td> 5 </td><td> C </td></
tr>
</tbody>
</table>
<hr/>
<h3> Centered Table </h3>
<table class="centered">
<thead>
    <tr><th> Student </th>
        <th> Class </th>
        <th> Grade </th>
    </tr>
</thead>
<tbody>
    <tr>
        <td >Mahesh </td>
        <td> 5 </td>
        <td> A+ </td>
    </tr>
    <tr>
        <td> Rahul </td>
        <td> 5 </td>

```

```

        <td> B </td>
    </tr>
    <tr>
        <td> Mohan </td>
        <td> 5 </td>
        <td> C </td>
    </tr>
</tbody>
</table>
<h3>Responsive table</h3>
<table class="responsive-table">
<thead>
    <tr>
        <th> Student </th>
        <th> Class </th>
        <th> Data 1 </th>
        <th> Data 2 </th>
        <th> Data 3 </th>
        <th> Data 4 </th>
        <th> Data 5 </th>
    </tr>
</thead>
<tbody>
<tr>
    <tr>
        <td>Rahul </td>
        <td>10</td>
        <td> 11 </td>
        <td> 12 </td>
        <td> 13 </td>
        <td> 14 </td>
        <td> 15 </td>
    </tr>
    <tr>
        <td>Mohan </td>
        <td>VI</td>
        <td>10</td>
        <td>11</td>
        <td>12</td>
        <td>13</td>
        <td>14</td>
    </tr>

```

```

        </tbody>
      </table>
    </body>
  </html>

```

MATERIALIZE – BUTTONS

It provides various classes to apply various to the buttons. There are following table mentions the available classes and their effects.

- **btn**: It sets button or anchor as a Materialize button, required. It sets raised display effect to a button.
- **btn-floating**: It creates a circular button.
- **btn-flat**: It sets flat display effect to a button, default.
- **btn-large**: It creates large buttons.
- **Disabled**: It creates a disabled button.
- **Type="submit"**: It represents an anchor or button as a primary button.
- **Waves-effect**: It sets ripple click effect and can be used in combination with any other classes.

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>The Materialize Buttons Example</title>
    <meta content="width=device-width,
initial-scale=1" name="viewport">
    <link rel = "stylesheet" href="https://fonts.
googleapis.com/icon?family=Material+Icons" >
    <script type = "text/javascript" src="https://cdnjs.
cloudflare.com/ajax/libs/materialize/0.97.3/js/
materialize.min.js">
  </script>

    <link href="http://cdnjs.cloudflare.com/ajax/
libs/materialize/0.97.3/css/materialize.min.css"
rel="stylesheet" >

```

```

<script type="text/javascript" src="https://
code.jquery.com/jquery-2.1.1.min.js"></script>

</head>
<body class="container">

    <div class="card-panel">
        <h5> Flat Buttons </h5>
        <button class="btn-flat waves-effect waves-
teal"> Add </button> </td>
        <button class="btn-flat waves-effect waves-
teal disabled" > <i class="material-icons left"> add
</i> Add </button> </td>
    </div>
    <div class="card-panel">
        <h5> Floating Buttons </h5>
        <a class="btn-floating waves-effect waves-
light red"> <i class="material-icons"> add</i> </a>
        <a class="btn-floating waves-effect waves-
light red disabled" > <i class="material-icons"> add
</i> </a>
    </div>
    <div class="card-panel">
        <h5> Raised Button </h5>
        <button class="btn waves-effect waves-teal">
Add </button> </td>
        <button class="btn waves-effect waves-
teal"><i class="material-icons left"> add </i> Add </
button> </td>
        <button class="btn waves-effect waves-
teal"><i class="material-icons right"> add </i> Add </
button> </td>
        <button class="btn waves-effect waves-teal
disabled"> Add </button> </td>
    </div>
    <div class="card-panel">
        <h5> Primary Buttons </h5>
        <button type="submit" class="btn waves-
effect waves-light red"> Send <i class="material-icons
right"> send </i> </button>

```

```

        <button type="submit" class="btn waves-
effect waves-light red disabled"> Cancel <i
class="material-icons right"> cancel </i> </button>
    </div>
    <div class="card-panel">
        <h5> Large Buttons </h5>
        <a class="btn-floating btn-large waves-effect
waves-light red"> <i class="material-icons"> add </i>
</a>
        <a type="submit" class="btn-floating btn-
large waves-effect waves-light red disabled" > <i
class="material-icons"> add </i> </a>
    </div>
</body>
</html>

```

MATERIALIZE – CARDS

Materialize provides various different CSS classes to apply on various predefined visual and behavioral enhancements for displaying various types of cards. The following table mentions all are available classes and their effects:

- Card: It identifies div element as a Materialize card container. It required on “outer” div.
- Card-content: It identifies div as a card content container and is required on “inner” div.
- Card-title: It identifies div as a card title container and is required on “inner” title div.
- Card-action: It identifies div as a card actions container and assigns appropriate text characteristics to actions text.
- Card-image: It identifies div as a card image container and is required on “inner” div.
- Card-reveal: It identifies div as a revealed text container.
- Activator: It identifies div as a revealed text container and img to be revealer. It is used to show contextual information related to image.

- Card-panel: It identifies div as a simple card with shadows and padding.
- Card-small: It identifies div as a small-sized card. Height–300px.
- Card-medium: It identifies div as a medium-sized card. Height–400px.
- Card-larger: It identifies div as a large-sized card. Height–500px.

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title> The Materialize Cards Example </title>
    <meta name="viewport" content="width=device-
width, initial-scale=1">
    <script src="https://cdnjs.cloudflare.com/
ajax/libs/materialize/0.97.3/js/materialize.min.js"></
script>
    <link rel="stylesheet" href="https://cdnjs.
cloudflare.com/ajax/libs/materialize/0.97.3/css/
materialize.min.css">
    <link rel="stylesheet" href="https://fonts.googleapis.
com/icon?family=Material+Icons">
    <script type="text/javascript" src="https://
code.jquery.com/jquery-2.1.1.min.js"></script>

  </head>
  <body class="container">
    <div class="row">
      <div class="col s12 m12 ">
        <div class="card blue-grey lighten-3">
          <div class="card-content">
            <span class="card-title"> <h3> The
Materialize Cards </h3> </span>
            <p> Nunc in elit quis nisl posuere
pulvinar.

                Donec iaculis quam vel ligula
tristique lacinia.

                Morbi vestibulum vel metus eget
dapibus. Fusce ut elementum tortor.
```

Nam eu ultrices odio, quis
 pulvinar magna. In feugiat sem ut congue blandit.
 Nunc vitae sapien nec tellus
 pharetra imperdiet a et tellus.

```

    </p>
  </div>
</div>
</div>

<!-- Row -->
<div class="row">
  <div class="col s12 m3">
    <div class="card-panel orange">
      <span class="white-text"> Simple Card
</span>
    </div>
  </div>
  <div class="col s12 m3">
    <div class="card small pink">
      <span class="white-text"> Small Card
</span>
    </div>
  </div>
  <div class="col s12 m3">
    <div class="card medium teal">
      <span class="white-text"> Medium Card
</span>
    </div>
  </div>
  <div class="col s12 m3">
    <div class="card large purple">
      <span class="white-text"> Large Card
</span>
    </div>
  </div>
</div>
</body>
</html>

```

MATERIALIZE – FORM

Materialize has a so beautiful and responsive CSS for form designing. The following CSS are used:

- **input-field:** It sets the <div> container as an input field container.
- **validate:** It adds validation styles to an input field.
- **active:** It shows an input with active style.
- **materialize-text-area:** It used to style a text-area. The text-areas will automatically resize to the text inside.
- **filled-in:** It shows a checkbox with filled box style.

Example:

```
<html>
  <head>
    <title>The Materialize Form Example</title>
    <meta content="width=device-width, initial-
scale=1" name="viewport">
    <linkhref="https://fonts.googleapis.com/
icon?family=Material+Icons" rel="stylesheet" >

    <script type = "text/javascript"
src="http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/js/materialize.min.js"></script>
    <link href="http://cdnjs.cloudflare.com/ajax/libs/
materialize/0.97.3/css/materialize.min.css"
rel="stylesheet" >

    <script type="text/javascript" src="http://code.
jquery.com/jquery-2.1.1
.min.js"></script>
  </head>
  <body class="container">
    <div class="row">
      <form class="col s12">
        <div class="row">
          <h1>User Data</h1>
          <div class="input-field col s6">
```

```

        <input value="Mahesh" id="name"
type="text" class="active validate"
placeholder="Username" required>
        <label for="name">Username</label>
    </div>
    <div class="input-field col s6">
        <label for="password">Password</
label>
        <input class="validate"
id="password" type="password" placeholder="Password"
required>
    </div>
</div>
<div class="row">
    <div class="input-field col s12">
        <input id="email"
placeholder="Email" type="email" class="validate">
        <label for="email">Email</label>
    </div>
</div>
<div class="row">
    <div class="input-field col s12">
        <textarea id="address"
class="materialize-textarea"></textarea>
        <label for="address">Address</label>
    </div>
</div>
<div class="row">
    <div class="input-field col s12">
        <input placeholder="Comments (You
can disabled it)" id="comments" type="text" >
        <label for="comments">Comments</
label>
    </div>
</div>
<!-- Checkboxes -->
<div class="row">
    <div class="input-field col s12">
        <p>
            <input id="married"
type="checkbox" checked="checked">

```

```

                                <label for="married">Married
</label>
                                </p>
                                <p>
                                    <input id="single"
type="checkbox" class="filled-in" >
                                    <label for="single">Single </
label>
                                </p>
                            </div>
                        </div>

<!-- Radio Buttons -->
<div class="row">
    <div class="input-field col s12">
        <p>
            <input type="radio" name="gender"
id="male" value="male" checked>
            <label for="male">Male</label>
        </p>
        <p>
            <input " type="radio"
name="gender" id="female value="female" checked>
            <label for="female">Female </
label>
        </p>
    </div>
</div>
</div>
</form>
</div>
</body>
</html>

```

User Data

Username	Password
Email	
Address	
Comments (You can disabled it)	
☒ Married ☐ Single	
Male ☒ Female	

Materialize – Form

FEATURES

Speeds up development: It provides default stylings that incorporate custom components. In extra, they refined animations and transitions to provide a smoother experience for developers.

User Experience Focused: By utilizing elements and principles of Material Design, they were able to create a framework that incorporates other components and animations that provide more benefits to users. Additionally, responsive system across all platforms allows for a more unified user experience.

They have provided detailed documentation as well as specific code examples to help new users get started.

CHAPTER SUMMARY

Here, we covered CSS framework Materialize with its basic concepts, which can be used to increase its features. Also, you will get how to add it just by installing or adding CDN in your HTML file with its advantages and disadvantages.

FURTHER READING

Content delivery network - <https://www.belugacdn.com/materialize-cdn/>, accessed on (2022 May 30).

Documentation - <https://github.com/Dogfalo/materialize>, accessed on (2022 May 30).

Getting started with Materialize - <https://medium.com/@swarnakishore/getting-started-with-materialize-css-cf8548969fdb>, accessed on (2022 May 30).

Materialize - <https://materializecss.com/>, accessed on (2022 May 30).

Materialize Css - https://dribbble.com/tags/materialize_css, accessed on (2022 May 30).

Materialize CSS Component Library - <https://www.figma.com/community/file/883610695243366061>, accessed on (2022 May 30).

Materialize CSS Tutorial - <https://www.javatpoint.com/materialize-css-tutorial>, accessed on (2022 May 30).

Materialize CSS Typography - <https://www.geeksforgeeks.org/materialize-css-typography/>, accessed on (2022 May 30).

Materialize Tutorial - <https://www.tutorialspoint.com/materialize/index.htm>, accessed on (2022 May 30).

materialize-css - <https://yarnpkg.com/package/materialize-css>, accessed on (2022 May 30).

Supported Browsers - <https://www.npmjs.com/package/materialize-css>, accessed on (2022 May 30).

CSS Cheat Sheet

CSS IS THE LANGUAGE of designing your website, which you can use to add backgrounds, colors, and even transitions or other interactive features. It will also help you to create a lightweight and responsive SEO-friendly website.

However, it can be difficult to get used to different CSS values, especially if you are new to the language.

To help you, we will share a complete CSS cheat sheet that will be useful for all your web development projects. In addition, you will find 13 pages of CSS and CSS3 announcements as well as a few possible layouts.

GETTING TO KNOW CSS

Cascading Style Sheet or CSS is a style sheet language that determines what features of your website should look like. You can control the structure, layout, font, and color of your website content by embedding a CSS file into HTML document.

Let's take a look at how CSS works its syntax is given below:

```
selector { declaration }
```

The announcement can be divided into

```
selector {property: value;}
```

As you can see, the CSS syntax consists of a selector and a declaration block. The selector is the element that you are about to control. Whereas

the declaration block contains the property name and the value of the element – both of them are separated by a colon (:) and declared within curly brackets ({}).

CSS stands for Cascading Style Sheets. HTML documents are formatted and presented using CSS. There are many features in HTML and CSS is a very popular add-on. Instead of changing the web page itself, with CSS, only styles need to be changed, which means less chance of total code. Additionally, HTML contained multiple duplicate codes on each web page, which can be embedded in a standard file with CSS and applied to all web pages. CSS contains - selector and announcement block. The declaration block contains a pair of property value pairs.

USING CSS IN HTML PAGES

It can be added to HTML documents in three different ways:

- Inline: by using the `<div style="">` attribute inside HTML elements
- Internal: by using a `<style>` and `</style>` element in the `<head>` section
- External: by using a `<link href="" rel="stylesheet">` element to link to an external CSS file.

Inline CSS, by Adding a Style Attribute to the HTML Tag

Example:

```
<!DOCTYPE html>
<html>
  <head></head>
  <title>Add CSS In HTML</title>
</head>
  <body style=" text-align : center ; padding : 50px" >
    <h1 style=" font-size: 2rem ; font-weight:
bolder; margin-bottom: 1rem "> CSS with Inline CSS
method </h1>
    <h2>I am a headline made with HTML</h2>
    <p style=" margin-bottom:1rem ; color: tomato">And
I am a simple text paragraph.
```

```

        <br> The color of this text is styled with
CSS. <br>
    </p>
</body>
</html>

```

CSS with Inline CSS method

I am a headline made with HTML

And I am a simple text paragraph.
The color of this text is styled with CSS.

Cheat Sheet - CSS with Inline CSS Method

Internal CSS, by Inserting a Style Tag in the Head Section

```
<head> <style> p { color:red } </style> </head>
```

Example:

```

<!DOCTYPE html>
<html>
    <head></head>
        <title>Add CSS In HTML</title>
        <style>
            body {
                text-align: center;
                padding: 50px;
            }

            h1 {
                font-size: 2rem;
                font-weight: bolder;
                margin-bottom: 1rem;
            }

            p {
                margin-bottom: 1rem;
                color: tomato;
            }
        </style>

```

```

</head>
<body>
  <h1> CSS with Internal CSS method </h1>
  <h2>I am a headline made with HTML</h2>
  <p> And I am a simple text paragraph.
    <br> The color of this text is styled with
CSS. <br>
  </p>
</body>
</html>

```

CSS with Internal CSS method

I am a headline made with HTML

And I am a simple text paragraph.
The color of this text is styled with CSS.

Cheat Sheet - CSS with Internal CSS Method

External CSS

```
<head><link rel="stylesheet" href="style.css"></head>
```

Example:

Index.html

```

<!DOCTYPE html>
<html>
  <head></head>
    <title> Add CSS In HTML </title>
    <link href="styles.css" rel="stylesheet">
  </head>
  <body>
    <h1> CSS with External CSS method </h1>
    <h2> I am a headline made with HTML </h2>
    <p> And I am a simple text paragraph.
      <br> The color of this text is styled with
CSS. <br>
    </p>
  </body>
</html>

```

Styles.css

```
body {  
    text-align: center;  
    padding: 50px;  
}  
  
h1 {  
    font-size: 2rem;  
    font-weight: bolder;  
    margin-bottom: 1rem;  
}  
  
p {  
    margin-bottom: 1rem;  
    color: tomato;  
}
```

CSS CHEAT SHEET

There are various sites we all use in CSS and here is a complete list of its layouts with their own example.

Font

- font
- font-style
- font-variant
- font-weight
- font-size
- font-family
- font-size-adjust
- font-stretch

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title> Add CSS In HTML </title>
  </head>
  <link href="styles.css" rel="stylesheet">
  <style>
    body {
      text-align: center;
      padding: 50px;
    }

    h1 {
      font-size: 3rem;
      margin-bottom: 1rem;
    }

    p {
      margin-bottom: 0.5rem;
    }

    .p-1{
      font-family: 'Franklin Gothic Medium', 'Aerial
Narrow', Aerial, sans-serif;
      font-size: 23px;
      line-height: 1.8em;
      font-weight: normal;
    }

    .p-2{
      font-size: 28px;
      font-style: italic;
    }

    .p-3{
      font-family: 'Franklin Gothic Medium', 'Aerial
Narrow', Aerial;
      font-size: 26px;
      line-height: 2em;
      font-weight: bolder;
    }

```

```

.p-4{
    font-size: 22px;
    font-variant: small-caps;
    font-stretch: ultra-condensed;
}

</style>
</head>
<body>
    <div class="container">
        <h1> <u> Font Attribute with its properties </
u> </h1>
        <p class="p-1"> The pain itself is love, but
the main customer </p>
        <p class="p-2" > I give such time as low as
toil and great pain </p>
        <p class="p-3" > the pain escapes from the pain
that none can result from. Excepteur are blacks </p>
        <p class="p-4" > The school district's
employment is only to achieve its objectives. homework
</p>
    </div>
</body>
</html>

```

Font Attribute with its properties

The pain itself is love, but the main customer

I give such time as low as toil and great pain

the pain escapes from the pain that none can result from. Excepteur are blacks

THE SCHOOL DISTRICT'S EMPLOYMENT IS ONLY TO ACHIEVE ITS OBJECTIVES. HOMEWORK

Cheat Sheet - Font Attribute with Its Properties

Text

- text-align
- color

- direction
- text-decoration
- white-space
- letter spacing
- text-outline
- text-wrap
- text-indent
- word-spacing
- text-transform
- text-emphasis
- text-justify
- text-shadow

Example:

```
<!DOCTYPE html>
<html>
  <head> </head>
    <title>Add CSS In HTML</title>
    <link href="styles.css" rel="stylesheet">
    <style>
      body {
        text-align: center;
        padding: 50px;
      }

      h1 {
        font-size: 3rem;
        margin-bottom: 1rem;
      }

      p {
        margin-bottom: 0.5rem;
      }
```

```
.p-1{
    font-size: 22px;
    line-height: 1.8em;
color: rgb(145, 36, 36)
}
.p-2{
    color:rgb(151, 151, 77);
    font-size: 28px;
    direction: rtl;
}
.p-3{
    font-size: 26px;
    color: rgb(36, 121, 142);
        line-height: 2em;
    letter-spacing: 5px;
}
.p-4{
    color: rgb(127, 131, 16);
    font-size: 22px;
    word-spacing: 5px;
}
.p-5{
    color: rgb(86, 8, 61);
    font-size: 22px;
    text-indent: 52m;
}
.p-6{
    color: rgb(87, 197, 136);
    font-size: 22px;
    text-align: left;
}
.p-7{
    color:rgb(87, 197, 136);
    font-size: 22px;
    text-align: center;
}
.p-8{
    color:rgb(87, 197, 136);
    font-size: 22px;
    text-align: right;
}
```

```
.p-9{
    color: rgb(211, 60, 222);
    font-size: 22px;
    text-decoration: underline
}
.p-10{
    color: rgb(222, 60, 60);
    font-size: 22px;
    text-decoration: line-through
}
.p-11{
    color:rgb(222, 214, 60);
    font-size: 22px;
    text-decoration: overline
}
.p-12{
    color:rgb(26, 137, 201);
    font-size: 22px;
    text-decoration: blink
}
.p-13{
    color: rgb(223, 98, 45);

    font-size: 22px;
    text-transform: capitalize
}
.p-14{
    color: rgb(224, 170, 229);
    font-size: 22px;
    text-transform: uppercase
}
.p-15{
    color: rgb(18, 112, 32);
    font-size: 22px;
    text-transform: lowercase
}
.p-16{
    font-size: 22px;
    white-space: pre
}
.p-17{
    text-shadow: 4px 4px 8px blue;
}
```

```

</style>
</head>
<body>
  <div class="container">
    <h1> <u> Text Attribute with its properties </
u> </h1>
    <p class="p-1"> This text will be written in
red. </p>
    <p class="p-2"> This text will be rendered
from right to left </p>
    <p class="p-3"> This text is having space
between letters </p>
    <p class="p-4"> This text is having space
between words. </p>
    <p class="p-5"> This text will have first line
indented by 1 cm and this line will remain at
its actual position this is done by
CSS text-indent property. </p>
    <p class="p-6"> This will be left aligned.</p>
    <p class="p-7"> This will be center aligned.</
p>
    <p class="p-8"> This will be right aligned.</
p>
    <p class="p-9"> This will be underlined</p>
    <p class="p-10"> This will be striked through.
</p>
    <p class="p-11"> This will have a over line.
</p>
    <p class="p-12"> This text will have blinking
effect </p>
    <p class="p-13"> This will be capitalized </p>
    <p class="p-14"> This will be in uppercase </
p>
    <p class="p-15"> This will be in lowercase.</
p>
    <p class="p-16">
      This text has a line break and the white-
space pre setting
      tells the browser to honor it just like
the HTML pre tag.
    </p>
    <p class="p-17">

```

If your browser supports the CSS text-shadow property,

this text will have a blue shadow.

</p>

</div>

</body>

</html>

Text Attribute with its properties

This text will be written in red.

This text will be rendered from right to left

This text is having space between letters

This text is having space between words.

This text will have first line indented by 1cm and this line will remain at its actual position this is done by CSS text-indent property.

This will be left aligned.

This will be center aligned.

This will be right aligned.

This will be underlined

~~This will be striked-through.~~

This will have a over line.

This text will have blinking effect

This Will Be Capitalized

THIS WILL BE IN UPPERCASE

this will be in lowercase.

This text has a line break and the white-space pre setting tells the browser to honor it just like the HTML pre tag.

If your browser supports the CSS text-shadow property, this text will have a blue shadow.

Cheat Sheet - Text Attribute with Its Properties

Backgrounds

- background-size
- background-image
- background-repeat
- background-attachment
- background-color

Example:

```

<!DOCTYPE html>
<html>
  <head></head>
    <title>Add CSS In HTML</title>
    <link href="styles.css" rel="stylesheet">
    <style>
      body {
        text-align: center;
        padding: 50px;
      }
      .container{
        max-width: 500px;
        margin: auto;
      }
      h1 {
        font-size: 3rem;
        margin-bottom: 1rem;
      }

      p {
        margin-bottom: 0.5rem;
      }
      .bc-1{
        font-size: 22px;
        background-color: brown;
      }

      .bc-2{
        background-color: #0000ff;
        font-size: 28px;
        direction: rtl;
      }
      .bc-3{
        font-size: 26px;
        background-color: rgb(0, 255, 0);
        letter-spacing: 5px;
      }
      .bc-4{
        background-color: rgba(0, 255, 0, 0.5);
        font-size: 22px;

```

```

        word-spacing: 5px;
    }
    .bc-5{
        background-color: transparent;
        font-size: 22px;
        text-indent: 52m;
    }
    .bc-6{
        background-color: hsl(0, 100%, 50%);
        font-size: 22px;
    }
    .bc-7{
        background-color: hsla(0, 100%, 50%, 0.1);
        font-size: 22px;
    }

</style>
</head>
<body>
    <div class="container">
        <h1><u> Background Attribute with its
different values of the background-color property.</
u></h1>
        <p class="bc-1">background-color: red; </p>
        <p class="bc-2"> background-color using Hex
value</p>
        <p class="bc-3"> background-color using
rgb </p>
        <p class="bc-4"> background-color using
rgba </p>
        <p class="bc-5"> background-color using
transparent </p>
        <p class="bc-6"> background-color using hsl </
p>
        <p class="bc-7"> background-color using hsla
</p>
        </div>

</body>
</html>

```

Background Attribute with its different values of the background-color property.

```
background-color: red;
```

```
background-color using Hex value
```

```
background-color using rgb
```

```
background-color using rgba
```

```
background-color using transparent
```

```
background-color using hsl
```

```
background-color using hsla
```

Background Attribute with Its Different Values of the Background-Color Property

- background-position

```
background-position: left top;
background-position: left center;
background-position: left bottom;
background-position: right top;
background-position: right center;
background-position: right bottom;
background-position: centre top;
background-position: 10% 40%;
background-position: 50px 100px;
```

- background-origin
- background-clip
- background image

Example:

```

<!DOCTYPE html>
<html>
  <head></head>
    <title>Add CSS In HTML</title>
    <link href="styles.css" rel="stylesheet">

    <style>
      body {
        text-align: center;
        padding: 50px;
      }
      .container{
        display: flex;
        justify-content: space-evenly;
      }
      h1 {
        font-size: 3rem;
        margin-bottom: 1rem;
      }

      p {
        margin-bottom: 0.5rem;
      }
      .bc-1{
        font-size: 22px;
        height:500px;
        width:500px;
        background-image: url('./Laptop-Pic.jpg');
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-position: center;
      }

      .bc-1{
        font-size: 22px;
        height:500px;
        width:500px;
        background-image: url('./Laptop-Pic.jpg');
        background-repeat: no-repeat;
        background-attachment: fixed;

```

```

background-position: center;
}

.bc-1{
    font-size: 26px;
    height:500px;
    width:500px;
background-image: url('../Laptop-Pic.jpg');
background-repeat: no-repeat;
background-attachment: fixed;
background-position: center;
}

.bc-2{
    font-size: 26px;
    height:500px;
    width:500px;
    background-image: conic-gradient(red, yellow,
green);
    background-repeat: no-repeat;
background-attachment: fixed;
background-position: left;
}

.bc-3{
    font-size: 26px;
    height:500px;
    width:500px;
    background-image: radial-gradient(red, green,
blue);
background-repeat: no-repeat;
background-attachment: fixed;
background-position: left;
}

</style>
</head>
<body>
    <h1><u> Background Image Attributes</u></h1>

    <div class="container">

```

```

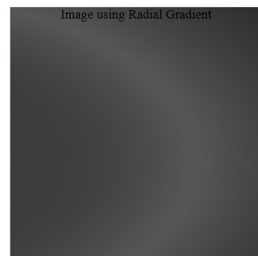
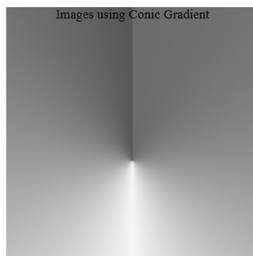
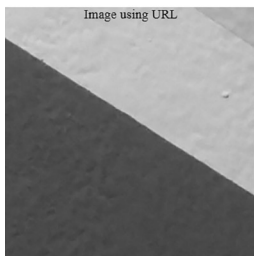
        <p class="bc-1"> Image using URL </p>
        <p class="bc-2"> Images using Conic Gradient
</p>
        <p class="bc-3"> Image using Radial Gradient
</p>

</div>

</body>
</html>

```

Background Image Attributes



Cheat Sheet - Background Image Attributes

Borders

- border

Example:

```

<!DOCTYPE html>
<html>
  <head></head>
  <title>Add CSS In HTML</title>
  <link href="styles.css" rel="stylesheet">

  <style>
  body {
    text-align: center;
    padding: 50px;
  }
  .container{

```

```
    display: flex;
    justify-content: space-evenly;
  }
  h1 {
    font-size: 3rem;
    margin-bottom: 1rem;
  }

  p {
    margin-bottom: 0.5rem;
  }

  .border-1{
    height:200px;
    width:200px;
    border: 5px solid red;
  }
  .border-2{
    height:200px;
    width:200px;
    border: 5px dotted red;
  }

  .border-3{
    height:200px;
    width:200px;
    border: 5px dashed red;
  }

  .border-4{
    height:200px;
    width:200px;
    border: 5px inset;
  }
  .border-5{
    height:200px;
    width:200px;

    border: 5px outset;
  }
```

```

        .border-6{
            height:200px;
            width:200px;
            border: 5px ridge;
        }
        .border-7{
            height:200px;
            width:200px;
            border: 5px groove;

        }
        .border-8{
            height:200px;
            width:200px;
            border: 5px double;

        }

</style>
</head>
<body>
    <h1><u> Border Attributes</u></h1>

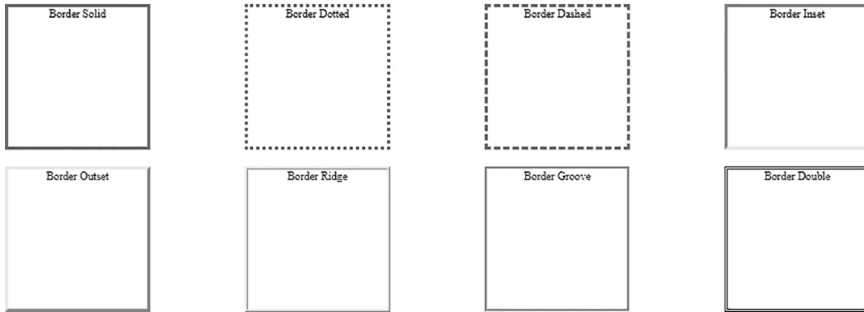
    <div class="container">
        <p class="border-1"> Border Solid
        <p class="border-2"> Border Dotted
        <p class="border-3"> Border Dashed
        <p class="border-4"> Border Inset
    </p></p></p></p>
    </div>

    <div class="container">
        <p class="border-5"> Border Outset
        <p class="border-6"> Border Ridge
        <p class="border-7"> Border Groove
        <p class="border-8"> Border Double

    </p></p></p></p>
    </div>

</body>
</html>

```

Border Attributes

Cheat Sheet - Border Attributes

- border-width

Example:

```
<!DOCTYPE html>
<html>
  <head></head>
    <title>Add CSS In HTML</title>
    <link href="styles.css" rel="stylesheet">

    <style>
      body {
        text-align: center;
        padding: 50px;
        font-size: 28px
      }
      .container{
        display: flex;
        justify-content: space-evenly;
      }
      h1 {
        font-size: 3rem;
        margin-bottom: 1rem;
      }

      p {
        margin-bottom: 0.5rem;
      }

      .border-1{
```

```

        height:auto;
        width:auto;
        border-style: solid;
border-width: thin;
    }

    .border-2{
        height:auto;
        width:auto;
        border-style: solid;
border-width: medium;
    }

    .border-3{
        height:auto;
        width:auto;
        border-style: solid;
border-width: thick;
    }
</style>
</head>
<body>
    <h1><u> CSS border-width Property </u></h1>

    <div class="container">
        <p class="border-1">The border-width property
does not work if it is used alone.
        <br> Use the border-style property to set
the border first.
        </p>
        <p class="border-2">The border-width property
does not work if it is used alone.
        <br> Use the border-style property to set
the border first.
        </p>
        <p class="border-3">The border-width property
does not work if it is used alone.
        <br> Use the border-style property to set
the border first.
        </p>
    </div>
</body>
</html>

```

CSS border-width Property.

The border-width property does not work if it is used alone. Use the border-style property to set the border first.	The border-width property does not work if it is used alone. Use the border-style property to set the border first.	The border-width property does not work if it is used alone. Use the border-style property to set the border first.
--	--	--

Cheat Sheet - CSS Border-width Property

- border-color
- border-left
- border-right
- border-bottom

Here is the example of above-mentioned attributes of Border.

Example:

```
<!DOCTYPE html>
<html>
  <head></head>
    <title>Add CSS In HTML</title>
    <link href="styles.css" rel="stylesheet">

    <style>
      body {
        text-align: center;
        padding: 50px;
        font-size: 28px;
      }
      .container{
        display: flex;
        justify-content: space-evenly;
      }
      h1 {
        font-size: 3rem;
        margin-bottom: 1rem;
      }

      p {
        margin-bottom: 0.5rem;
      }
```

```

.border-1{
    padding: 20px;
    height: auto;
    width: auto;
    border-style: solid;
    border-color: coral;
}

.border-2{
    padding:20px;
    height:auto;
    width:auto;
    border-top:1px solid black;
}

.border-3{
    padding: 20px;
    height: auto;
    width: auto;
    border-bottom: 1px solid black;
}
.border-4{
    padding: 20px;
    height: auto;
    width: auto;
    border-right: 1px solid black;
}

.border-5{
    padding: 20px;
    height: auto;
    width: auto;
    border-left: 1px solid black;
}
</style>
</head>
<body>
    <h1><u> CSS border-width Property </u></h1>

    <div class="container">
        <p class="border-1">
            <u> Border Color </u><br> <br>

```

The border-width property does not work if it is used alone.

 Use the border-style property to set the border first.

```
</p>
</div>
<div class="container">
  <p class="border-2">
    <u> Border Top </u><br> <br>

  </p>
  <p class="border-3">
    <u> Border Bottom </u><br> <br>

  </p>
  <p class="border-4">
    <u> Border Right </u><br> <br>
  </p>
  <p class="border-4">
    <u> Border Left </u><br> <br>
  </p>
</div>
</body>
</html>
```

CSS border-width Property.

Border Color

The border-width property does not work if it is used alone.
Use the border-style property to set the border first.

<u>Border Top</u>	<u>Border Bottom</u>	<u>Border Right</u>	<u>Border Left</u>
-------------------	----------------------	---------------------	--------------------

Cheat Sheet - Border Color, Border (Top, Bottom, Left, Right)

- border-radius
- border-image

Box Model

- float

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

.container{
    max-width: 500px;
    margin: 0 auto;
}
.left{
    float: left;
}
.right{
    float: right;
}
h1{
    text-align: center;
}
</style>
</head>
<body>

<div class="container">
    <h1> Box Model - float</h1>
    <div class="left"> Float - Left </div>
    <div class="right"> Float - Right </div>
</div>
</body>
</html>
```

Box Model - float

Float - Left

Float - Right

- width
- min-width
- max-width

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

.container{
    margin: 0 auto;
}
.left{
    width: 200px;
    border: 1px solid black;
    background-color: aqua;
}
.right{
    min-width: 100px;
    border: 1px solid black;
    background-color: beige;
}
.center{
    max-width: 300px;
    border: 1px solid black;
    background-color: aquamarine;
}
h1{
    text-align: center;
}
</style>
</head>
<body>

<div class="container">
    <h1> Box Model - Width </h1>
    <div class="left"> Left </div>
    <div class="center"> CEnter </div>

```

```

    <div class="right"> Right </div>
</div>
</body>
</html>

```

Box Model - Width



Cheat Sheet Box Model - Width

- max-height
- height
- min-height

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

.container{
    max-width: 300px;
    margin: 0 auto;
}
.left{
    height: 200px;
    border: 1px solid black;
    background-color: aqua;
}
.right{
    min-height:150px;
    border: 1px solid black;
    background-color: beige;
}
.center{
    max-height: 300px;
    border: 1px solid black;
    background-color: aquamarine;
}

```

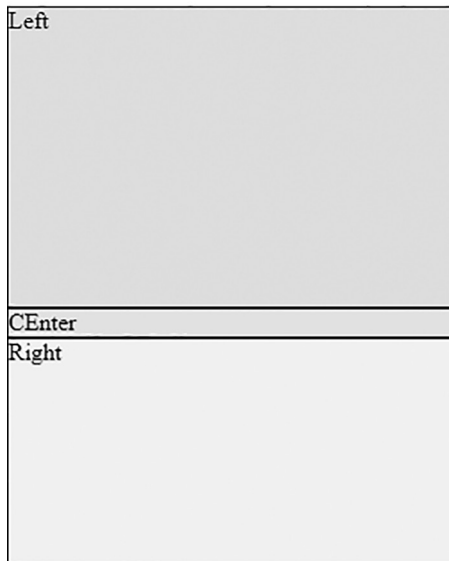
```

h1{
    text-align: center;
}
</style>
</head>
<body>

<div class="container">
    <h1> Box Model - Width </h1>
    <div class="left"> Left </div>
    <div class="center"> CEnter </div>
    <div class="right"> Right </div>
</div>
</body>
</html>

```

Box Model - Width



Cheat Sheet Box Model - Height

Margin

- margin-bottom
- margin-left

- margin-right
- margin-top

Padding

- padding-bottom
- padding-top
- padding-right
- padding-left
- display
- marquee-direction
- marquee-loop
- marquee-speed
- marquee-style
- overflow
- overflow-style
- overflow-x
- rotation
- rotation-point
- visibility
- clear

Template Layout

- box-align
- box-direction
- box-flex

- box-flex-group
- box-lines
- box-orient
- box-pack
- box-sizing
- tab-sid

Table

- border-collapse
- empty-cells
- border-spacing
- table-layout
- caption-side

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

.container{
    max-width: 800px;
    margin: 0 auto;
    text-align: center;
}
.
h1{
    text-align: center;
}
table, td, th {
    border: 1px solid;
}
```

```

table {
    width: 100%;
    border-collapse: collapse;
}
table.table1 {
    empty-cells: show;
}
#table1 {
    border-collapse: separate;
    border-spacing: 15px;
}

table.c {
    table-layout: auto;
    width: 100%;
}

</style>
</head>
<body>

<div class="container">
    <h2>Table</h2>

    <table class="table1">
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
        </tr>
        <tr>
            <td> Alfreds </td>
            <td> Futterkiste </td>
        </tr>
        <tr>
            <td> Island </td>
            <td> </td>
        </tr>
        <tr>
            <td> Magazzini Alimentari Riuniti </td>
            <td> Alimentari </td>
        </tr>
    </table>

```

```

        <td> Alimentari    </td>
        <td>    </td>
    </tr>
</table>

```

<h2> Empty Cells and Border Spacing </h1>

```

<table id="table1">
    <th>Firstname</th>
    <th>Lastname</th>
</tr>
<tr>
    <td> Alfreds </td>
    <td> Futterkiste </td>
</tr>
<tr>
    <td> Island </td>
    <td>    </td>
</tr>
<tr>
    <td> Magazzini Alimentari Riuniti </td>
    <td> Alimentari </td>
</tr>
<tr>
    <td> Alimentari    </td>
    <td>    </td>
</tr>
</table>

```

<h2> Table Layout </h2>

```

<table class="c">
    <tr>
        <th>Company</th>
        <th>Contact</th>
        <th>Country</th>
    </tr>
    <tr>
        <td>Alfreds Futterkiste</td>
        <td>Maria Anders</td>
        <td>Germany</td>
    </tr>
    <tr>
        <td>Island Trading</td>
        <td>Helen Bennett</td>

```

```

        <td>UK</td>
    </tr>
    <tr>
        <td>Magazzini Alimentari Riuniti</td>
        <td>Giovanni Rovelli</td>
        <td>Italy</td>
    </tr>
</table>
<h2> Caption Side </h2>
<table id="example1" border="1">
<caption>Table 1.1 Customers</caption>
<tr>
    <th> Company </th>
    <th> Contact </th>
    <th> Country </th>
</tr>
<tr>
    <td> Futterkiste </td>
    <td> Maria Anders </td>
    <td> Germany </td>
</tr>
<tr>
    <td> Berglunds snabbköp</td>
    <td> Christina </td>
    <td> Sweden </td>
</tr>
<tr>
    <td> comercial Moctezuma </td>
    <td> Francisco Chang </td>
    <td> Mexico </td>
</tr>
<tr>
    <td> Ernst </td>
    <td> Roland Mendel </td>
    <td> Austria </td>
</tr>
</table>

</div>
</body>
</html>

```

Table

Firstname	Lastname
Alfreds	Futterkiste
Island	
Magazzini Alimentari Riuniti	Alimentari
Alimentari	

Empty Cells and Border Spacing

Firstname	Lastname
Alfreds	Futterkiste
Island	
Magazzini Alimentari Riuniti	Alimentari
Alimentari	

Table Layout

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Island Trading	Helen Bennett	UK
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Caption Side

Table 1.1 Customers

Company	Contact	Country
Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina	Sweden
comercial Moctezuma	Francisco Chang	Mexico
Ernst	Roland Mendel	Austria

CSS Cheat Sheet - Table

Colors

- color

Example:

CSS color codes and names.

- Red colors
- Orange colors
- Yellow colors
- Green colors

- Cyan colors
- Blue colors
- Purple colors
- Pink colors
- White colors
- Gray colors
- Brown colors
- opacity

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

body{
    margin: 10%;
    text-align: center;
}
.container{
    max-width: 800px;
    margin: 0 auto;
    text-align: center;
}

.c1{
    background-color: Tomato
}
.c2{
    background-color : Orange
}
.c3{
    background-color: DodgerBlue
}
.c4{
    background-color: MediumSeaGreen
```

```
}  
.c5{  
    background-color: Gray  
}  
.c6{  
    background-color: SlateBlue  
}  
.c7{  
    background-color: Violet  
}  
.c8{  
    background-color: LightGray  
}  
.o1{  
    background-color: Tomato;  
    opacity: 1;  
  
}  
.o2{  
    background-color : Orange;  
    opacity: 0.9;  
  
}  
.o3{  
    background-color: DodgerBlue;  
    opacity: 0.8;  
  
}  
.o4{  
    background-color: MediumSeaGreen;  
    opacity: 0.7;  
  
}  
.o5{  
    background-color: Gray;  
    opacity: 0.6;  
  
}  
.o6{  
    background-color: SlateBlue;  
    opacity: 0.5;  
  
}
```

```

.o7{
  background-color: Violet;
  opacity: 0.4;

}
.o8{
  background-color: LightGray;
  opacity: 0.3;

}
.o9{
  background-color: LightGray;
  opacity: 0.2;

}

.o10{
  background-color: LightGray;
  opacity: 0.1;

}

</style>
</head>
<body>

<h1>CSS Color </h1>
<h2 class="c1"> Tomato </h2>
<h2 class="c2" > Orange </h2>
<h2 class="c3" > DodgerBlue </h2>
<h2 class="c4" > MediumSeaGreen </h2>
<h2 class="c5" > Gray </h2>
<h2 class="c6" > SlateBlue </h2>
<h2 class="c7" > Violet </h2>
<h2 class="c8" > LightGray </h2>
<br>
<h1>CSS Color (Opacity) </h1>
<h2 class="o1"> Tomato (1) </h2>
<h2 class="o2" > Orange (0.9) </h2>
<h2 class="o3" > DodgerBlue (0.8) </h2>
<h2 class="o4" > MediumSeaGreen (0.7) </h2>
<h2 class="o5" > Gray (0.6) </h2>

```

```
<h2 class="o6" > SlateBlue (0.5)</h2>
<h2 class="o7" > Violet (0.4) </h2>
<h2 class="o8" > LightGray (0.3) </h2>
<h2 class="o9" > Red (0.2) </h2>
<h2 class="o10" > Green (0.1) </h2>

</body>
</html>
```



CSS Cheat Sheet - CSS Color and Opacity

Grid Positioning

- grid-columns
- grid-rows

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

body{
  margin: 10%;
  text-align: center;
}
```

```

.container{
    max-width: 800px;
    margin: 0 auto;
    text-align: center;
}

.grid-container {
    display: grid;
    grid-template-columns: auto auto auto;
    background-color: #2196F3;
    padding: 10px;
}
.g-item {
    background-color: rgba(255, 255, 255, 0.8);
    border: 1px solid rgba(0, 0, 0, 0.8);
    padding: 20px;
    font-size: 30px;
    text-align: center;
}

</style>
</head>
<body>

    <h1> Grid </h1>
    <div class="grid-container">
        <div class="g-item">1</div>
        <div class="g-item">2</div>
        <div class="g-item">3</div>
        <div class="g-item">4</div>
        <div class="g-item">5</div>
        <div class="g-item">6</div>
        <div class="g-item">7</div>
        <div class="g-item">8</div>
        <div class="g-item">9</div>
    </div>

</body>
</html>

```

Grid

1	2	3
4	5	6
7	8	9

CSS Cheat Sheet - Grid

There are various properties of Grid given below:

- gap
- grid
- grid-auto-columns
- grid-auto-flow
- grid-auto-rows
- grid-column
- grid-column-end
- grid-column-gap
- grid-column-start
- grid-gap
- grid-row-end
- grid-row-gap
- grid-row-start
- grid-template
- grid-template-areas
- grid-template-columns
- grid-template-rows
- row-gap

Columns

- column-count
- column-fill
- column-gap
- column-rule-width
- column-rule-style
- column-rule-color
- column-width

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

body{
    margin:10%;
}
div {
    height:300px;
    background-color:#FFFFFF;
    column-count:3;
    column-gap: 10px;
    column-rule-width: 10px;
    column-rule-style: dotted;
    column-rule-color:blueviolet;
    column-width:10px
}

#myDIV {
    column-fill: balance;
}

</style>
</head>
```

```
<body>
```

```
<h1>The column-count properties</h1>
```

```
<div id="myDIV">
```

Lorem ipsum dolor sit amet, adipiscing elit, sed diam nonummy nibh euismod ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis exerci tation ullamcorper suscipit nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in in vulputate velit esse molestie consequat, vel dolore eu feugiat nulla facilisis at vero eros et acumsan et iusto odio dignissim qui blandit praesent luptatum.

```
</div>
```

```
</body>
```

```
</html>
```

The column-count properties

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation

• ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.
• Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse

• molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

CSS Cheat Sheet - The Column-Count Properties

List and Markers

- list-style-type
- list-style-position
- list-style-image
- marker-offset

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
  body{
    margin:10%;
  }
```

```

ul.a {
    list-style-type: circle;
    outline-offset: 15px;
}
ul.b {
    list-style-type: square;
    list-style-image: ('');
}
ol.c {
    list-style-type: upper-roman;
    list-style-position: outside;
}
ol.d {
    list-style-type: lower-alpha;
    list-style-position: inside;
}
</style>
</head>
<body>

<h1>The list-style-type Property</h1>

<p>Example of unordered lists:</p>

<ul class="a">
    <li> Audi </li>
    <li> BMW </li>
    <li> Volve</li>
</ul>

<ul class="b">
    <li> Audi </li>
    <li> BMW </li>
    <li> Volve</li>
</ul>

<p>Example of ordered lists:</p>

<ol class="c">
    <li> Audi </li>

```

```

    <li> BMW </li>
    <li> Volve</li>
</ol>

<ol class="d">
    <li> Audi </li>
    <li> BMW </li>
    <li> Volve</li>
</ol>

</body>
</html>

```

The list-style-type Property

Example of unordered lists:

- Audi
 - BMW
 - Volve
- Audi
 - BMW
 - Volve

Example of ordered lists:

- I. Audi
 - II. BMW
 - III. Volve
- a. Audi
 - b. BMW
 - c. Volve

CSS Cheat Sheet - List Property

Animations

- animation-name
- animation-duration
- animation-timing-function
- animation-delay
- animation-iteration-count


```

        100% {
            background-color: brown;
        }
    }
</style>
</head>

<body>
    <div id="animation">
        <div id="animation1"> CSS Animations </div>
        <div id="animation2">It is a technique to
change the appearance and behavior of various
elements</div>
    </div>
</body>

</html>

```



Cheat Sheet - CSS Animation

Outline

- outline-color
- outline-style
- outline-width
- outline-offset

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

```

```

p {outline-color:red;}
.container{
    display:flex;
    justify-content:space-evenly;
}
h2{
    text-align:center;
}

p.dotted {
    outline-style: dotted;
    outline-width: medium;

}
p.dashed {
    outline-style: dashed;
}
p.solid {
    outline-style: solid;
}
p.double {
    outline-style: double;
    outline-width: thick;

}
p.groove {
    outline-style: groove
    ;}
p.ridge {
    outline-style: ridge;

    outline-width: thin;
}
p.inset {outline-style: inset;}
p.outset {outline-style: outset;}
</style>
</head>
<body>

<h2>The outline-style Property </h2>
<div class="container">
    <p class="dotted">A dotted outline</p>

```

```

<p class="dashed">A dashed outline</p>
<p class="solid">A solid outline</p>
<p class="double">A double outline</p>
<p class="groove">A groove outline. </p>
<p class="ridge">A ridge outline. </p>
<p class="inset">An inset outline.</p>
<p class="outset">An outset outline. </p>
</div>
</body>
</html>

```

The outline-style Property



CSS Cheat Sheet – Outline Property

Positioning

- bottom/right/top/left
- z-index
- clip
- position

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
  body{
    margin:10%
  }
  div.static {
    position: static;
    border: 3px solid #73AD21;
  }
  div.fixed {
    position: fixed;
    border: 3px solid #73AD21;
  }

```

```
div.relative {
  position: relative;
  border: 3px solid #73AD21;
}
```

```
div.absolute {
  position: absolute;
  border: 3px solid #73AD21;
}
```

```
div.sticky {
  position: sticky;
  border: 3px solid #73AD21;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h2> Positioning</h2>
```

```
<h2> Static </h2>
```

```
  This div element has position: static;
```

```
</div>
```

```
<h2> Fixed </h2>
```

```
<div class="fixed">
```

```
  This div element has position: fixed;
```

```
</div>
```

```
<h2> Relative </h2>
```

```
<div class="relative">
```

```
  This div element has position: relative;
```

```
</div>
```

```
<h2> Sticky </h2>
```

```
<div class="sticky">
```

```
  This div element has position: sticky;
```

```
</div>
```

```
<h2> Absolute </h2>
```

```
<div class="absolute">
```

```
  This div element has position: absolute;
```

```
</div>
```

```
</body>
```

```
</html>
```

Positioning**Static**

This div element has position: static;

Fixed

Relative This div element has position: fixed;

This div element has position: relative;

Sticky

This div element has position: sticky;

Absolute

This div element has position: absolute;

CSS Cheat Sheet – Positioning

Transitions

- transitions-delay
- transitions-duration
- transitions-property
- transition-timing-function

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>

body{
    margin: 10%;
}
.container{
    max-width: 800px;
    margin: 0 auto;
    text-align: center;
}
h2,p{
    text-align: center;
}
div {
```

```

    width: 200px;
    height: 100px;
    background: lightcoral;
    transition-property: width;
    transition-duration: 5s;
    transition-delay: 2s;
}

div:hover {
    width: 300px;
}
</style>
</head>
<body>

<h2> CSS Transitions </h2>
<p> When you bring mouse over the container. You will
see the transition. </p>

<div class="container">
    <h2> Bring Mouse Here </h2>
</div>
</body>
</html>

```

CSS Transitions

When you bring mouse over the container. You will see the transition.



CSS Cheat Sheet - Transitions

3D/2D Transform

- perspective
- perspective-origin

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

#div1 {
    position: relative;
    height: 150px;
    width: 200px;
    border: 1px solid black;
    perspective: 100px;
    perspective-origin: left;
}

#div2, #div4 {
    padding: 50px;
    position: absolute;
    border: 1px solid gainsboro;
    background-color: red;
    background: rgba(230, 202, 202, 0.5);
    transform-style: preserve-3d;
    transform: rotateX(45deg);
}

#div3 {
    position: relative;
    height: 150px;
    width: 150px;
    margin-left: 60px;
    border: 1px solid lightslategray;
    perspective: 100px;
    perspective-origin: bottom right;
}

h1{
    text-align: center;
}

.container{
    display: flex;
    justify-content: center;
}

```

```

</style>
</head>
<body>

    <h1>The perspective-origin Property</h1>
    <div class="container">

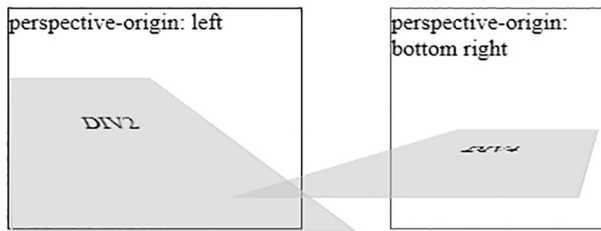
        <div id="div1"> perspective-origin: left

        <div id="div2">DIV2</div>
    </div>

    <div id="div3">perspective-origin: bottom right
        <div id="div4">DIV4</div>
    </div>
</div>
</body>
</html>

```

The perspective-origin Property



CSS Cheat Sheet - The Perspective-Origin Property

- transform

With the CSS transform property, you can use the following 2D transformation methods:

- translate()
- rotate()
- scaleX()
- scaleY()
- scale()

- skewX()
- skewY()
- skew()
- matrix()

Example:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
    <style>

div{
  display: flex;
  background-color:bisque;
  justify-content: space-evenly;

}
h1{
  text-align: center;
}
.translate{
  transform: translate(0px, 0px);
  border: 1px solid lightslategray;
}
.translate:hover{
  transform: translate(10px, 5px);
  border: px solid lightslategray;
  display: flex;
  align-items: center;
  justify-content: center;
}

.rotate{
  transform: rotate(0deg);
  border: 1px solid lightslategray;
  display: flex;
  align-items: center;
  justify-content: center;
}
```

```

.rotate:hover{
    transform: rotate(20deg);
    border: 1px solid lightslategray;
    display: flex;
    align-items: center;
    justify-content: center;
}

.scaleX{
    transform: scaleX(1);
    border: 1px solid lightslategray;
}
.scaleX:hover{
    transform: scaleX(2);
    border: 1px solid lightslategray;
}
.scaleY{
    transform: scaleY(1);
    border: 1px solid lightslategray;
}
.scaleY:hover{
    transform: scaleY(2);
    border: 1px solid lightslategray;
}
.scale{
    transform: scale(1,1);
    border: 1px solid lightslategray;
}

.scale:hover{
    transform: scale(1.5, 1.5);
    border: 1px solid lightslategray;
}

/* Another Container Items */

.skew{
    transform: skew(0deg, 0deg);
    border: 1px solid lightslategray;
}

.skew:hover{

```

```

    transform: skew(20deg, 10deg);
    border: 1px solid lightslategray;
}

.matrix{
    transform: scale(1,1);
    border: 1px solid lightslategray;
}

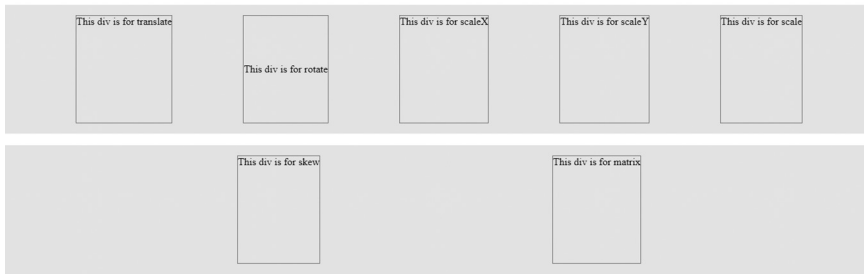
.matrix:hover{
    transform: scale(2,2);
    border: 1px solid lightslategray;
}

.container-1{
    position: relative;
    display: flex;
    height:200px
}

.container-2{
    display: flex;
    height:200px
}
</style>
</head>
<body>
<h1> 2D Transform </h1>
<div class="container-1">
    <p class="translate"> This div is for translate </p>
    <p class="rotate"> This div is for rotate </p>
    <p class="scaleX"> This div is for scaleX </p>
    <p class="scaleY"> This div is for scaleY </p>
    <p class="scale"> This div is for scale </p>
</div>
<br>
<div class="container-2">
    <p class="skew"> This div is for skew </p>
    <p class="matrix"> This div is for matrix </p>
</div>

</body>
</html>

```

2D Transform**CSS Cheat Sheet - 2D Transform**

- transform-origin

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
    <style>
      .origin {
        padding: 10px;
        position: absolute;
        border: 1px solid black;
        background-color: red;
        transform: rotate(50deg);
        transform-origin: 10% 70%;
      }

      .container{
        position: relative;
        height: 350px;
        width: 350px;
        margin: 100px;
        padding: 10px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <h1> Transform Origin </h1>
    </div>
  </body>
</html>

```

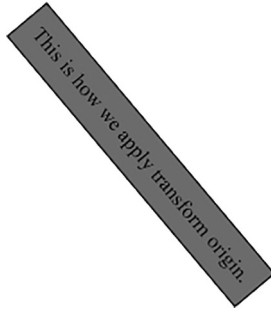
```

    <div class="origin"> This is how we apply
transform origin.</div>
  </div>

  </body>
</html>

```

Transform Origin



CSS Cheat Sheet - Transform Origin

Pseudo-Class

The Pseudo-classes in CSS are used to add special effects to some selectors. You don't need to use JavaScript or any other script to use those effects. A syntax of pseudo-classes is as follows:

```
selector:pseudo-class {property: value}
```

- :active

Example:

```

<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      a:link {
        background-color: burlywood;
      }
    </style>
  </head>
  <body>
    <div class="origin"> This is how we apply
transform origin.</div>
  </div>
</body>
</html>

```

```

    </style>
</head>

<body>
  <h1> Pseudo Classes in CSS </h1>
  <h3> Active </h3>
  <a href = ""> Link</a>
</body>
</html>

```

Pseudo Classes in CSS

Active

Link

CSS Cheat Sheet - Pseudo Classes (Active)

- :focus

Example:

```

<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      a:focus {
        background-color: burlywood;
      }
    </style>
  </head>

  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> Active </h3>
    When you click on this link the background color
    change <a href = ""> Link</a>
  </body>
</html>

```

Pseudo Classes in CSS

Active

When you click on this link the background color change [Link](#)

CSS Cheat Sheet - Pseudo Classes (Focus)

- :visited

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      a:visited {color: red}

    </style>
  </head>

  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> Visited </h3>
    <a href = ""> Link</a>
  </body>
</html>
```

Pseudo Classes in CSS

Visited

[Link](#)

CSS Cheat Sheet - Pseudo Classes (Visited)

- :enabled
- :disabled

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <title>Title of the document</title>
    <style>
      input {
        padding: 2px 5px;
        margin-bottom: 10px;
        border: 1px solid #ccc;
      }
      input[type=text]:enabled {
        background: lightgreen;
      }
      input[type=text]:disabled {
        background: salmon;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> Disabled and Enabled Class</h3>
    <form action="">
      <label for="name">First name:</label>
      <input type="text" enabled="enabled" value="Sam"
id="name">
      <br>
      <label for="country">Country:</label>
      <input type="text" disabled="disabled"
value="India" id="country">
    </form>
  </body>
</html>

```

Pseudo Classes in CSS

Disabled and Enabled Class

First name:

Country:

CSS Cheat Sheet - Pseudo Classes (Disabled - Enabled)

- :hover

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      a:hover {color: #FFCC00;
        font-size:36px}
    </style>
  </head>

  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> Visited </h3>
    The color of the link will change when you drag
    your mouse over the link - <a href = ""> Link</a>
  </body>
</html>
```

Pseudo Classes in CSS

Visited

The color of the link will change when you drag your mouse over the link - [Link](#)

CSS Cheat Sheet - Pseudo Classes (Hover)

- :checked

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      input:checked {
        height: 30px;
        width: 20px;
      }
    </style>
  </head>

  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> checked </h3>
    <form action="">
      <p>You can tick the box to check the working
of the checked pseudo class</p>
      <input type="checkbox" checked="checked"
value="Bike"> I have a bike<br>
      <input type="checkbox" value="Car"> I have a
car
    </form>
  </html>
```

Pseudo Classes in CSS

checked

You can tick the box to check the working of the checked pseudo class

- ☐ I have a bike
- ☐ I have a car

CSS Cheat Sheet - Pseudo Classes (Checked)

- :selection

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      ::selection {
        color: red;
        background: yellow;
      }
    </style>
  </head>

  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> selection </h3>
    <p> Here we have some text when you select
    any of the text its color get changes.</p>
  </html>
```

Pseudo Classes in CSS

selection

Here we have some text when you select any of the text its color get changes.



CSS Cheat Sheet Pseudo Classes (Selection)

- :nth-child(n)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:nth-child(3) {
        background: lightgreen;
      }
    </style>
  </head>

  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> nth-child(n) </h3>

    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
      <li>Fourth list item</li>
      <li>Fifth list item</li>
    </ul>

  </body>
</html>
```

Pseudo Classes in CSS

nth-child(n)

- First list item
- Second list item
- **Third list item**
- Fourth list item
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (nth-child(n))

- :nth-last-child(n)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:nth-last-child(3) {
        background: lightgreen;
      }
    </style>
  </head>

  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> nth-last-child(n) </h3>

    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
      <li>Fourth list item</li>
      <li>Fifth list item</li>
    </ul>

  </body>
</html>
```

Pseudo Classes in CSS

nth-last-child(n)

- First list item
- Second list item
- **Third list item**
- Fourth list item
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (nth-last-child(n))

- :first-child (use this symbol >)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      div > p:first-child {
        background-color: #FFCC00;
      }
    </style>
  </head>

  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> First - Child </h3>
    <div>
      <p>First paragraph in div.</p>
      <p>Second paragraph in div.</p>
    </div>
    <p>This paragraph is not in any div </p>
    <div>
      <p>This paragraph will also be the first paragraph
in the other div tags.</p>
      <p>This is a second paragraph inside the div.
This paragraph will not be effected.</p>
    </div>
  </html>
```

Pseudo Classes in CSS

First - Child

First paragraph in div.

Second paragraph in div.d

This paragraph is not in any div

This paragraph will also be the first paragraph in the other div tags.

This is a second paragraph inside the div. This paragraph will not be effected.

CSS Cheat Sheet - Pseudo Classes (first-child)

- :last-child (use this symbol >)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      div > p:last-child {
        background-color: #FFCC00;
      }
    </style>
  </head>

  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> Last - Child </h3>

    <div>
      <p>First paragraph in div.</p>
      <p>Second paragraph in div.d</p>
    </div>

    <p>This paragraph is not in any div </p>

    <div>
```

<p>This paragraph will also be the first paragraph in the other div tags.</p>

<p>This is a second paragraph inside the div. This paragraph will not be effected.</p>
</div>

</html>

Pseudo Classes in CSS

Last - Child

First paragraph in div.

Second paragraph in div.d

This paragraph is not in any div

This paragraph will also be the first paragraph in the other div tags.

This is a second paragraph inside the div. This paragraph will not be effected.

CSS Cheat Sheet - Pseudo Classes (last-child)

- :only-child

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
        text-align: center;
      }
      p:only-child {
        background: red;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> Only - Child </h3>
    <p>This is a paragraph. </p>
    <div><p> This is a paragraph and only child of
div. </p></div>
    <div><p> This is a another paragraph and only
child of div. </p></div>
  </body>
</html>
```

Pseudo Classes in CSS

Only - Child

This is a paragraph.

This is a paragraph and only child of div.

This is a another paragraph and only child of div.

CSS Cheat Sheet - Pseudo Classes (only-child)

- :nth-of-type(n)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:nth-of-type(2) {
        background: lightgreen;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> nth-of-type(n) </h3>

    <ul>
      <li> First list item </li>
      <li> Second list item </li>
      <li> Third list item </li>
      <li> Fourth list item </li>
      <li> Fifth list item </li>
    </ul>
  </body>
</html>
```

Pseudo Classes in CSS

nth-of-type(n)

- First list item
- **Second list item**
- Third list item
- Fourth list item
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (nth-of-type(n))

- :nth-last-of-type(n)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:nth-last-of-type(2) {
        background: lightgreen;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> nth-last-of-type(n) </h3>

    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
      <li>Fourth list item</li>
      <li>Fifth list item</li>
    </ul>

  </body>
</html>
```

Pseudo Classes in CSS

nth-last-of-type(n)

- First list item
- Second list item
- Third list item
- **Fourth list item**
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (nth-last-of-type(n))

- :first-of-type
- :last-of-type

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:first-of-type {
        background: red;
      }
      li:last-of-type {
        background: yellowgreen;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> first-of-type and last-of-type </h3>

    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
      <li>Fourth list item</li>
      <li>Fifth list item</li>
    </ul>

  </body>
</html>
```

Pseudo Classes in CSS

first-of-type and last-of-type

- First list item
- Second list item
- Third list item
- Fourth list item
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (first-of-type and last-of-type)

- :only-of-type

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      li:only-of-type {
background: rgb(192, 118, 192);
}
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> only-of-type </h3>

    <li>First list item</li>
    <ul>
      <li>First list item</li>
      <li>Second list item</li>
      <li>Third list item</li>
      <li>Fourth list item</li>
      <li>Fifth list item</li>
    </ul>

  </body>
</html>
```

Pseudo Classes in CSS

only-of-type

• First list item

- First list item
- Second list item
- Third list item
- Fourth list item
- Fifth list item

CSS Cheat Sheet - Pseudo Classes (only-of-type)

- :empty

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      p:empty {
        width: 100px;
        height: 20px;
        background: lightblue;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> empty </h3>
    <p></p>
    <p>Second Paragraph</p>
    <p>Third Paragraph</p>
    <p>Fourth Paragraph</p>
  </body>
</html>
```

Pseudo Classes in CSS

empty



Second Paragraph

Third Paragraph

Fourth Paragraph

CSS Cheat Sheet - Pseudo Classes (empty)

- :root

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      :root {
        background: lightcyan;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> root </h3>
    <p>Second Paragraph</p>
    <p>Third Paragraph</p>
    <p>Fourth Paragraph</p>
  </body>
</html>
```

Pseudo Classes in CSS

root

Second Paragraph

Third Paragraph

Fourth Paragraph

CSS Cheat Sheet - Pseudo Classes (root)

- :target

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      :target {
        border: 2px solid #e0b8b8;
        background-color: #cbd9a6;
      }
    </style>
  </head>
  <body>

    <h1> Pseudo Classes in CSS </h1>
    <h3> target </h3>
    <p><a href="#news1"> New content 1 </a></p>
    <p><a href="#news2"> New content 2 </a></p>

    <p>Click on the links above and the :target selector
    highlight the current active HTML anchor.</p>

    <p id="news1"><b> Target Content 1 </b></p>
    <p id="news2"><b> Target Content 2</b></p>

  </body>
</html>
```

Pseudo Classes in CSS

target

[New content 1](#)

[New content 2](#)

Click on the links above and the :target selector highlight the current active HTML anchor.

Target Content 1

Target Content 2

CSS Cheat Sheet - Pseudo Classes (target)

- :not(x)

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }

      p {
        color: blue;
      }

      :not(p) {
        color: green;
      }
    </style>
  </head>
  <body>
    <h1> Pseudo Classes in CSS </h1>
    <h3> not(p) </h3>
    <p> Paragraph 1</p>
    <p> Paragraph 2 </p>
    <div>This is some text in a div element.</div>
  </body>
</html>
```

Pseudo Classes in CSS

not(p)

Paragraph 1

Paragraph 2

This is some text in a div element.

CSS Cheat Sheet - Pseudo Classes (not(p))

Pseudo-Element

Pseudo-elements in CSS are used to add special effects to selectors. You don't need to use JavaScript or any other script to use those effects. Syntax of pseudo-element is as follows:

```
selector: pseudo-element {property: value}
```

- ::first-letter

Example:

```
<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      p: first-letter { text-decoration: underline;
        font-size: 30px;
      }
      p.first-para: first-letter { text-decoration:
none; }
      h1, h3{
        text-align: center;
      }
    </style>
  </head>

  <body>
    <h1> Pseudo Elements in CSS </h1>
```

```

    <h3> First Letter </h3>
    <p class="first-para">
        This line will not have underline because
this belongs to inline class.
    </p>

    <p>
        The first line of this paragraph will be
underlined using the Pseudo element in CSS.
        Rest of the lines in this paragraph will
show normal.
        This example shows how we can use :first-line
pseduo element to give effect
        to the first line of HTML element.
    </p>
</body>
</html>

```

Pseudo Elements in CSS

First Letter

This line will not have underline because this belongs to inline class.

The first line of this paragraph will be underlined using the Pseudo element in CSS. Rest of the lines in this paragraph will show normal. This example shows how we can use :first-line pseduo element to give effect to the first line of HTML element.

CSS Cheat Sheet - Pseudo Element (first-letter)

- ::first-line

Example:

```

<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      p:first-line { text-decoration: underline; }
      p.first-para:first-line { text-decoration:
none; }
      h1{
        text-align: center;
      }
    </style>
  </head>

```

```

<body>
  <h1> Pseudo Elements in CSS </h1>
  <p class="first-para">
    This line will not have underline because
this belongs to inline class.
  </p>

  <p>
    The first line of this paragraph will be
underlined using the Pseudo element in CSS.
    Rest of the lines in this paragraph will
show normal.
    This example shows how we can use :first-line
pseduo element to give effect
    to the first line of HTML element.
  </p>
</body>
</html>

```

Pseudo Elements in CSS

First Line

This line will not have underline because this belongs to inline class.

The first line of this paragraph will be underlined using the Pseudo element in CSS. Rest of the lines in this paragraph will show normal. This example shows how we can use first-line pseduo element to give effect to the first line of HTML element.

CSS Cheat Sheet – Pseudo Element (first-line)

- ::before

Example:

```

<html>
  <head>
    <style type = "text/css">
      body{
        margin:5%;
      }
      p:before {
        content: 'This content before paragraph';
        width:20px;
        height: 20px;
        background-color: bisque;
      }
    </style>
  </head>
  <body>
    <p>
      This content before paragraph
    </p>
  </body>
</html>

```

```

        p.first-para: first-line { text-decoration:
none; }
        h1, h3{
            text-align: center;
        }
    </style>
</head>

<body>
    <h1> Pseudo Elements in CSS </h1>
    <h3> First Line </h3>
    <p class="first-para">
        This line will not have underline because
this belongs to inline class.
    </p>

    <p>
        The first line of this paragraph will be
underlined using the Pseudo element in CSS.
        Rest of the lines in this paragraph will
show normal.
        This example shows how we can use :first-line
pseduo element to give effect
        to the first line of HTML element.
    </p>
</body>
</html>

```

Pseudo Elements in CSS

First Line

This content before paragraph This line will not have underline because this belongs to inline class.

This content before paragraph The first line of this paragraph will be underlined using the Pseudo element in CSS. Rest of the lines in this paragraph will show normal. This example shows how we can use :first-line pseduo element to give effect to the first line of HTML element.

CSS Cheat Sheet – Pseudo Element (before)

- ::after

Example:

```

<html>
    <head>
        <style type = "text/css">
            body{

```

```

        margin:5%;
    }
    p:after {
        content: 'This content after paragraph';
        width:20px;
        height: 20px;
        background-color: bisque;
    }
    p.first-para:first-line { text-decoration:
none; }
    h1, h3{
        text-align: center;
    }
</style>
</head>

<body>
    <h1> Pseudo Elements in CSS </h1>
    <h3> First Line </h3>
    <p class="first-para">
        This line will not have underline because
this belongs to inline class.
    </p>

    <p>
        The first line of this paragraph will be
underlined using the Pseudo element in CSS.
        Rest of the lines in this paragraph will
show normal.
        This example shows how we can use :first-line
pseduo element to give effect
        to the first line of HTML element.
    </p>
</body>
</html>

```

Pseudo Elements in CSS

First Line

This line will not have underline because this belongs to inline class. This content after paragraph

The first line of this paragraph will be underlined using the Pseudo element in CSS. Rest of the lines in this paragraph will show normal. This example shows how we can use :first-line pseduo element to give effect to the first line of HTML element. This content after paragraph

CHAPTER SUMMARY

In this chapter, we get to know how we can use the CSS attributes in coding also we have covered few examples of attributes of Margin. Padding, Width, Height, Box Model, List, etc.

Appraisal

CSS IS THE LANGUAGE of designing your website, which you can use to add backgrounds, colors, and even transitions or other interactive features. It will also help you to create a lightweight and responsive SEO-friendly website. CSS is a style sheet language that describes how the features of your website should look. You can control the structure, layout, font, and color of your website. Why Use CSS? CSS is used to define styles for your web pages, including the design, layout, and variations in display for different devices and screen sizes. A CSS rule-set consists of a selector and a declaration block.

CSS Features

- If anyone wants to start a professional web design project, it is important to have knowledge of CSS and HTML.
- By using CSS, we can control a variety of styles, such as text color, font style, segmentation, column and layout size, background color and images, layout, layout and contrast for different screens, and device size, and many other effects.
- CSS has the ability to control HTML documents, so it's easy to read. Integrated with HTML and XHTML tagging languages.
- Other Languages: Once we have mastered some of the basics of CSS and HTML, other related technologies such as Angular, PHP, and JavaScript become clearer for us to understand.

There are various CSS framework in the market such as Bootstrap, Bulma, Tailwind, Foundation, Skeleton, Milligram, Pure, and Materialize.

First, Bootstrap is a free and open-source CSS-based framework. It is used to create mobile-first responsive websites. It is one of the most popular front-end frameworks in the world. As of August 2021, it is the tenth most starred project on GitHub. Bootstrap essentially contains components that you can use to create design templates. These components use CSS and sometimes JavaScript as well. For example, forms, buttons, navbar, etc. Mark Otto and Jacob Thornton initially created Bootstrap. It was released for the first time more than 10 years ago, in August 2011. Since then, many changes have been made, and improved versions have been released. The latest stable version was released in October 2021, and it is called Bootstrap 5. It was initially called Twitter Blueprint as both Mark and Jacob used to work at Twitter and wanted to create a framework that would help them maintain consistency across the platform. However, utilizing multiple different libraries to make the user interface resulted in inconsistencies and differences across the platform. It was also proving incredibly difficult to maintain the code. This duo of a developer and a designer then came up with Bootstrap. Bootstrap has a prevalent 12-column grid system that helps quickly build layouts. The grid system is fully responsive and has six breakpoints based on screen resolution. Bootstrap also has many pre-built components that the users can modify according to their needs by using the additional classes provided.

PREREQUISITES OF CSS FRAMEWORKS

- **HTML:** It stands for Hypertext Markup Language. HTML is a standard markup language that is used for creating web pages to display in a web browser. Web browsers usually receive HTML documents from the server and then render them in the correct format from the client. Knowledge of HTML is a must for learning Bootstrap.
- **CSS:** It stands for Cascading Style Sheets. CSS is a style sheet language used to describe the presentation and styling of an HTML document. It was developed by World Wide Web Consortium (W3C) in 1996. The knowledge of CSS is essential for learning Bootstrap because to use any component of Bootstrap, you have to use the corresponding utility classes.

- JavaScript: It is considered the programming language of the web. Combined with HTML and CSS, these three languages are considered the first stepping stones for web development. Even though knowledge of JavaScript is not essential, it would prove to be useful as there are lots of JavaScript plugins in Bootstrap. You can also add your custom functionality by using Bootstrap.

ALTERNATIVE VERSIONS OF CSS FRAMEWORKS

- Foundation: It is one of the most advanced and comprehensive frameworks available in the market. It is a family of front-end frameworks with unique features and unmatched flexibility. It is a free front-facing framework with open source, which provides a responsive grid and components for HTML and CSS UI, templates, and captions, including typing, forms, buttons, navigation and other interactive features, as well as the dedicated functionality with JavaScript extensions. The Foundation is an open-source project, and was maintained by ZURB. Since 2019, the Foundation has been cared for by volunteers.
- Bulma CSS: It is a pure CSS framework which means it has no JavaScript plugins. The flexbox model heavily inspires it. Bulma is built on Flexbox, a CSS layout model that automatically adjusts the page aspect width based on its container width. Flexbox requires developers to write both HTML and CSS to create features that stick to this model. With Bulma, Flexbox behavior and many other things can be handled with HTML classes. Bulma CSS is also an open source – you can view its source code in the Bulma GitHub – which allows developers to create new jobs within the framework.
- Tailwind CSS: It is a low-level CSS framework that is also highly customizable. It has a vast range of utility classes that you can mix and match according to your needs. Tailwind CSS offers many features, some notable features, intuitive style, flexibility, and freedom for its users when designing a web application. Therefore, we take full responsibility for making you understand what Tailwind is and how it differs from other frameworks such as bootstrap.

Tailwind CSS has recently grown into the first widely used CSS framework so we need to come up with a blog that outlines the framework and simplifies its installation process for our readers.

- **Materialize CSS:** It is a modern responsive CSS framework based on Material Design by Google. The framework has components and animations that help provide more feedback to the users. Even though the UI components are similar to Bootstrap, they look and feel much lighter, which gives it a better look. Materialize is a partial UI library built with CSS, JavaScript, and HTML. Reusable UI components help create attractive, consistent, functional, and web-based web pages while adhering to modern web design principles such as browser portability, device autonomy, and downtime.

Materialize has a responsive design built in so that a website created using Materialize will redesign itself as a device size. Material classes are designed in such a way that the website can fit any screen size.

- **Pure CSS:** Pure CSS is a CSS framework. A collection of free and open tools for creating responsive websites and web applications. Pure CSS is developed by Yahoo and is used to create fast, beautiful, and responsive websites. It can be used as an alternative to Bootstrap.

Index

Note: *Italic* page numbers refer to figures.

- active/disabled buttons 100
- adjacent sibling selectors 11
- alert links 96, 97
- alerts 95
- align items 91–92
- align-self flexbox 90–91
- animations CSS cheat sheet 435–437, 437
- @import rule 32
- @media rule 31–32
- Atom 51
- attribute selector 34–35

- badges 97–98
- block level buttons 100
- block-level elements 23–24
- blockquote
 - milligram 306–307, 307
- Bootstrap 46, 476
 - badges 97–98
 - breakpoints 72–73
 - buttons 99–103
 - columns 73
 - components 95–97
 - containers 60–63, 73
 - create home page with 59
 - download using package manager 51–54
 - dropdowns 102–103
 - features 58
 - flexbox 77–93
 - grid system 63–64
 - gutters 73
 - history 46, 56–58
 - jumbotron 98–99
 - meta tag 56
 - Navbars 103–105
 - pages 59–60
 - responsive breakpoints 93–94
 - rows 73
 - Sass variables 73
 - start with 51
 - structure 64–72
 - template 54–56
 - versions 47–49
- Bootstrap 5, 50
- border color 206–208, 208
- border-color property 20
- border-style property 19
- border opacity 208–210, 211
- border-radius classes 197–202, 199, 203
- borders CSS cheat sheet 408–415, 411, 413, 415
- border style 211–212, 212
- border width 19, 203–205, 206
- box border and content 148–149, 150
- box model 22–23, 416, 416–419, 418, 419
- Bulma 477
 - browser support 109–110
 - for CDN 109
 - column system 111
 - control placeholder 130
 - CSS library 108
 - extends 132
 - features and benefits 108–109
 - findColorInvert () function 131–132
 - form control Mixins 129
 - functions 131
 - libraries 115–118

Bulma (*cont.*)

- with node-Sass installation 113
- package 108
- responsive mixins 126–129
- Sass 113–115
- sizes 130
- starter template 110–111
- utilities 119–126
- variables 111–112

buttons

- Bootstrap 99–103
- pure CSS 343–351, 347, 351
- skeleton 286–290, 288, 290

button tags 101

callout 240–245, 242, 244, 245

card divider 248, 250–252, 252

cards 383–385

Cascading Style Sheets (CSS) 391–392

- attribute selector 34–35
- background color feature 17–18
- background properties 16
- benefits 38–40
- block-level elements 23–24
- border property 19–21
- box model 22–23
- box size feature 30
- classes 7–8
- colors 13–16
- comments 13
- definition 1–2
- drops 40–41
- element type selectors 9–12
- features 475–476
- files 5–6
- flexbox 29–30
- float property 25
- fragmentation 36–38
- functions 26–29
- Grid Layout 30
- history 2
- HTML 5, 6, 392–395, 393
- length 22
- location 8–9
- margins 20–21
- max-width 22, 24
- media questions and responsive web design 33

media types 31–33

modules 5

overflow 25

padding 21

rules and values 6–7

selectors 9

specification 4–5

syntax 2–3

tables 23

types 4

universal selector 9

version 4

width 22, 24

z-index layout 24

case insensitivity 38

cheat sheet

- animations 435–437, 437
- backgrounds 402–408, 405, 408
- borders 408–415, 411, 413, 415
- box model 416, 416–419, 418, 419
- colors 424–429, 429
- columns 432–433, 433
- font 395–397, 397
- grid positioning 429–431, 431
- list and markers 433–435, 435
- margin 419–420
- outline 437–439, 439
- padding 420
- positioning 439–440, 441
- Pseudo-class 449–473, 450, 451, 453–473
- table 421–424, 425
- template layout 420–421
- text 397–402, 402
- 3D/2D transform 442–449, 444, 448, 449
- transitions 441–442, 442

child selectors 11

class selectors 10

clear class 158–163, 160, 163

closable 246–247, 247

code

milligram 318–319, 319

skeleton 295–296, 296

control Mixins 129

colored spinners 100–101

colors 13–16

- Content Delivery Network & Content Distribution Network (CDN) 53
- CSS FlexBox 49
- CSS1, 4
- CSS2, 4
- CSS2.1, 4
- CSS3, 2, 4, 14, 25–27, 40, 304, 391
- CSS4, 4
- descendant selectors 10–11
- flex 174–176, 176
- flexbox 29–30, 77–93
- flex direction 171–173, 173
- flex-fill 93
- flex-grow 92–93
- flex-shrink 92–93
- flex-wrap
 - in Bootstrap 83–84
 - in CSS 82
- findColorInvert () function 131–132
- float property 25
- fragmentation 36–38
- full-width jumbotron 98–99
- font 139–140, 140
- form control Mixins 129
- Foundation 225–226, 477
 - accordion and URLs 238–240, 240
 - advanced options 236–238, 238
 - callout coloring 240–244, 242, 244
 - callout sizing 244–245, 245
 - card 248–249, 249
 - card divider 250–252, 252
 - closable 246–247, 247
 - compatibility 229–230
 - dropdown 252–254, 254
 - features 273–274
 - file structure 228
 - foundation integration levels 274
 - installation 226–228
 - jQuery 229
 - kitchen sink 231–236
 - mobile devices 228
 - off-canvas 264–268, 266, 268
 - position 254–259, 256, 259
 - principles 230
 - reveal (model) 268–273, 270, 273
 - section alignment 261–263
 - functions 26–29
 - Bulma 131
 - grid system
 - Bootstrap 63–64
 - materialize 367
 - milligram 316–317, 318
 - skeleton 277–280, 280
 - grid template columns 178–180, 180
 - grouping selectors 12
 - gutters 73
- HEX 20
- hexadecimal notation 14
- Horizontal-to-Scrollable menu 331–338, 338
- Hypertext Markup Language (HTML) 5, 6, 45, 392–395, 393, 476
- id selector 9–10
- indented syntax 48
- invisible element 17
- JavaScript 45, 53, 477
- Jquery Library 53
- jsDelivr 53, 54
- JIT. *see* Just-In-Time (JIT)
- jQuery 229
- jumbotron 98–99
- justify content 181–183, 183
 - in Bootstrap 89–90
 - in CSS 84–89
- justify items 183–186, 186
- Just-In-Time (JIT) 147
 - mode features 148
- kitchen sink 229, 231–236
- margin 144–145, 146, 190–193, 194
- materialize 359, 478
 - buttons 381–383
 - cards 383–385
 - color 362–366, 364, 366
 - columns for large screen devices 371–373, 373
 - columns for medium screen devices 369–371, 371
 - columns for small screen devices 367–369, 369

- materialize (*cont.*)
 - features 389
 - form 386–388, 389
 - grids 367
 - install 360–361, 362
 - media 375–377
 - table 377–381
 - utility classes 373–375, 375
- meta tag 56
- media 31–33
 - materialize 375–377
 - queries 33–34
- milligram
 - absolute lengths 301–303, 304
 - blockquote 306–307, 307
 - button 308–311, 311
 - code 318–319, 319
 - column widths 321–323
 - forms 313–314, 314
 - grid 316–317, 318
 - installation 299–300
 - lists 311–312, 312
 - not to use 325
 - reasons to use 325
 - relative length units 304
 - responsiveness 323–325
 - tables 314–316, 316
 - typography 304–306, 306
 - utilities 319–320, 321
- mobile-first development 46
- Navbars 103–105
- Node.js 52
- Normalize.css 49
- off-canvas 264–268, 266, 268
- outline buttons 101–102
- overflow classes 163–170, 166, 168, 171
- padding classes 141–143, 143, 186–189, 190
- Popper.js 53, 56, 102
- pure CSS 327–328, 478
 - buttons 343–351, 347, 351
 - foundation 339, 340
 - Horizontal-to-Scrollable menu
 - 331–338, 338
 - images 340–343, 343
 - install 328–330, 330
 - responsive design 331
 - tables 352–358, 357
- relative length units 304
- Responsive Mixins 126–129
- reveal (model) 268–273, 270, 273
- RGB functional notation 15
- ring opacity 213
- ring width 213–214, 215
- rows 73
- Sass 113–115
- Sass Script 48, 49
- Sass variables 73
- shadow effect properties 215–216, 217
- space between 194–196, 196
- skeleton 275–276
 - benefits 297
 - buttons 286–290, 288, 290
 - code 295–296, 296
 - disadvantages 297
 - file structure 276
 - forms 291–293, 293
 - grid system 277–280, 280
 - length property 286
 - lists 293–295, 295
 - supported browser 276–277
 - typography 280–285, 282, 285
- tables collapse and layout 217–219, 220
- Tailwind CSS 135–136, 477
 - alignment 181–183, 183
 - benefits 220–221
 - border color 206–208, 208
 - border opacity 208–210, 211
 - border-radius classes 197–202, 199, 203
 - border style 211–212, 212
 - border width 203–205, 206
 - box sizing 148–149, 150
 - CDN 137–139, 139
 - clear class 158–163, 160, 163
 - composition 222
 - consistency 222
 - container 141
 - customization 222
 - disadvantages 223–224
 - display 150–156, 152, 155
 - flex 174–176, 176

- flex direction 171–173, 173
- float classes 156–157, 157
- grid template columns 178–180, 180
- install 136–137
- justify items 183–186, 186
- layout 141–146, 143, 146
- less cognitive load 221–222
- margin 190–193, 194
- order 176–178, 178
- overflow classes 163–170, 166, 168, 171
- Play CDN 223
- ring opacity 213
- ring width 213–214, 215
- shadow effect properties 215–216, 217
- small styles 221
- space between 194–196, 196
- spacing 186–189, 190
- start with 136–137
- syntax 221
- tables collapse and layout 217–219, 220
- typography 139–140
- tooltip 226
- three equal column 73–74
- Twitter Blueprint 46
- two equal column 74–77
- typography
 - milligram 304–306, 306
 - skeleton 280–285, 282, 285
 - Tailwind CSS 139–140
- VS Code 51
- web-level technologies 4
- z-Index
 - Bootstrap 94
 - layout 24
- ZURB 225, 226, 231, 249, 273, 477



Taylor & Francis Group
an informa business



Taylor & Francis eBooks

www.taylorfrancis.com

A single destination for eBooks from Taylor & Francis with increased functionality and an improved user experience to meet the needs of our customers.

90,000+ eBooks of award-winning academic content in Humanities, Social Science, Science, Technology, Engineering, and Medical written by a global network of editors and authors.

TAYLOR & FRANCIS EBOOKS OFFERS:

A streamlined experience for our library customers

A single point of discovery for all of our eBook content

Improved search and discovery of content at both book and chapter level

REQUEST A FREE TRIAL
support@taylorfrancis.com

 **Routledge**
Taylor & Francis Group

 **CRC Press**
Taylor & Francis Group