

# WEB DEVELOPMENT

WEB DESIGN WITH CSS



ANDY VICKLER

**Web Development**

**Web Design with CSS**

**© Copyright 2021 - All rights reserved.**

The contents of this book may not be reproduced, duplicated or transmitted without direct written permission from the author.

Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

**Legal Notice:**

This book is copyright protected. This is only for personal use. You cannot amend, distribute, sell, use, quote or paraphrase any part or the content within this book without the consent of the author.

**Disclaimer Notice:**

Please note the information contained within this document is for educational and entertainment purposes only. Every attempt has been made to provide accurate, up to date and reliable complete information. No warranties of any kind are expressed or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. The content of this book has been derived from various sources. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, —errors, omissions, or inaccuracies.

# **Table of Contents**

## **Introduction**

## **Chapter One: Web Design with CSS**

*The Tools*

*Notepad & TextEdit*

*Color Coding*

*Visual Aids*

*Completion*

## **Chapter Two: Introducing CSS**

*CSS Syntax*

*CSS Selectors*

*Class Selector*

*CSS Universal Selector*

*The Grouping Selector*

*Adding CSS*

*The External CSS*

*The Internal CSS*

*Inline CSS Styling*

*Multiple Style Sheets*

*Comments*

*Background Colors*

*Text Color*

*Border Colors*

*Alternate Coloring Method*

*RGB Colors*

*RGBA Colors*

*HEX Colors*

*HSL Colors*

## **Chapter Three: CSS Backgrounds**

*CSS Background Properties*

*Images*

*CSS Borders*

*Width*

*Border Colors*

*Border Sides*

*Rounded Borders*

## **Chapter Four: CSS Margins & Padding**

*Margin Collapse Feature*

*CSS Padding*

*CSS Width & Height*

## **Chapter Five: CSS Boxes**

*The Outline*

*Width Property*

*Color Property*

*Outline Offset Property*

*CSS Texts*

*Text Direction*

*Vertical Alignment*

*Decoration*

*Text Transformation*

*Indentation*

*Letter-Spacing*

*Height*

*Word Spacing*

*Text Shadow*

*CSS Fonts*

*CSS Font Style*

*Font-Weight Property*

*Font-Variant Property*

*Font-Size Property*

[CSS Font-Pairings](#)

## **[Chapter Six: Lists, Tables, Displays](#)**

[CSS Lists](#)

[Adding Colors](#)

[CSS Tables](#)

[The Border Property](#)

[The Width Property](#)

[Aligning Table](#)

[CSS Layout](#)

[Positioning Property](#)

[Static](#)

[Relative](#)

[Fixed](#)

[Absolute](#)

[Sticky](#)

[Elements Overlapping](#)

[Layout Overflow](#)

[Visible](#)

[Hidden](#)

[Scroll](#)

[CSS Float](#)

[Clear Property](#)

[CSS Float Examples](#)

[Equal Height Boxes](#)

[Designing Menu](#)

## **[Chapter Seven: CSS Display Features](#)**

[Inline-Block](#)

[Navigation Links](#)

[Alignment](#)

[Center-Aligning Text](#)

[Image Aligning](#)

[Left & Right Aligning](#)

*[Float for Alignment](#)*

## **Chapter Eight: CSS Combinators**

*[Descendant Selector](#)*

*[Child Selector](#)*

*[Adjacent Sibling Selector](#)*

*[General Sibling Selector](#)*

## **Chapter Nine: CSS Bars, Dropdowns & Galleries**

*[Vertical Bars](#)*

*[Active Class](#)*

*[Adding Borders](#)*

*[Sticky Navbar](#)*

*[Horizontal NavBar](#)*

*[Inline](#)*

*[Floats](#)*

*[Glamorizing Horizontal Bars](#)*

*[Active Links](#)*

*[Right Alignment](#)*

*[Fixed Navigation Bar](#)*

*[Sticky Bar](#)*

## **Chapter Ten: CSS Dropdowns & Images**

*[A Basic Dropdown](#)*

*[A Dropdown List](#)*

*[Dropdown Menu Navbar](#)*

*[Image Gallery](#)*

*[Image Sprites](#)*

## **Chapter Eleven: CSS Attributes & Forms**

*[CSS Forms](#)*

*[Padding Property](#)*

*[More Design Space](#)*

*[Colored Inputs](#)*

[Focused Inputs](#)

[Textareas](#)

[Menu Styling](#)

[Responsive Form](#)

[CSS Counters](#)

[Nesting Counters](#)

## **[Chapter Twelve: CSS Web Layout](#)**

[Navbar](#)

[Content](#)

[Unequal Columns](#)

[Footer](#)

**[Conclusion](#)**

**[References](#)**

## Introduction

This book contains proven steps and strategies for designing your website with CSS. I have created all the code from scratch so that you can copy them and paste them on an editor to practice. You can change the values, properties, and styles to bring about different results. CSS allows you to try a variety of color combinations, border styles, border patterns, boxes, forms to design your web pages.

Almost all HTML elements welcome CSS as a styling agent. You can take up single HTML elements and give them unique styles to create a unique look for your website. This book contains lots of code and related output so that you use the code for practicing purposes. In this book, some of the output is missing, and some may not match what you see on a webpage because it was not possible to integrate the web pages in the book. However, when you execute the code in the editor and upload it to a browser, you will see that the output is what I have described and what you have been expecting. Also, please note that different internet browsers produce different results. Some accept a code, and others don't. Therefore, please don't forget to try one code on different browsers like Firefox, Internet Explorer, Safari, and Chrome.

One of the most amazing chapters is the last chapter of the book that contains the codes to create and design your web pages' layout. Once you learn how to design individual elements, the book will take you on the ride to create web layouts. You will learn how to set up the headers, the footers, and the columns to integrate blog posts and other important elements.

I recommend that you keep your laptop running when you start reading the book. I also recommend that you install a professional editor like Adobe Dreamweaver on your system so that you can test all the code samples that I have included in the book. It is also a better option to install multiple internet browsers on your system to test different code pieces so that you know whether your code works best on all browsers. Use the CSS code that works well on all the browsers so that you don't have to lose customers just because your CSS code does not work on their browsers.

If you have a working knowledge of HTML, you will have fun learning sessions. However, you do not have to be an expert in HTML. The coding for

HTML is neatly written and CSS is embedded clearly in the `<style>` and `</style>` tags. After going through a couple of code examples, you will start getting a grasp of which HTML element is used for what purpose. Also, keep in mind that each CSS code has to design one or more HTML elements.

## **Chapter One: Web Design with CSS**

Website creation demands an in-depth understanding of the principles of design and the technology deployed to display the content to the audience. Before you start reading the book, you should make sure that you have a functional computer system and an operating system. You should understand the directory system of your computer so that you may navigate different folders and files. You should know how to save, locate and open different types of files and you should know how to use the mouse to access commands and folders.

When you start working in CSS, you need to install a text editor. There is no specific editor that you must install to kick off the operations. You can use Adobe Dream Weaver and Microsoft Expression Web on the computer.

## ***The Tools***

There are several tools you can use to design your website with CSS language. They range from simple to complex and up to some really sturdy packages. Your tool choice is based on your personal preference, the tools you purchase, and the scope of your project. CSS uses text as its foundation. That's why you can even use the simple text editors like Notepad, TextEdit, etc. They work well for creating and designing web pages. However, these basic editing tools lack some advanced features like checking the code syntax and organizing different site folders. Also, you cannot upload your code to the web servers as well.

When you think you have passed the basic level, you can use advanced code editors like Adobe Dreamweaver and Microsoft Expression Web. Plain text editors like TextEdit and Notepad come with the Mac OS and Windows operations systems, respectively. You don't have to install them on your system as they come pre-installed. When working with plain text editors, you need excellent knowledge of CSS coding because these basic editors don't provide any kind of guidance along the way. They also lack the functionality to preview the pages in a live web browser. That's why you may find it hard to check your web pages to correct the syntax and hunt out the broken links in the pages.

## ***Notepad & TextEdit***

You can find out Notepad in the Windows accessories panel. See the following steps to create your first HTML file to include your CSS code.

- You have to go to Start, then Programs, then Accessories, and then Notepad. Then choose File and click on New. The file will be saved by default in a .txt format. The web browser will not interpret many of the HTML tags your file contains.
- When you open Notepad, you will see File on the top menu. Now click Save As. Here you can change the type of file extension from .txt to .html in the field designated for the name of the file.
- Hit on All Files in the Save As menu. Now set your encoding value to UTF-8. Now click Save.

If you are working on a Mac OS, you will find TextEdit to create your CSS files.

- On your system, open the Applications folder and watch the TextEdit application. The software TextEdit, by default, is created to save documents in .rtf format. You must change the preferences of the application to use HTML successfully.
- Now you should choose Preferences. You will see a dialog box here.
- In the tab of the New Document, you should choose the Plain text radio button. Now you will only see the plain text without any formatting.
- Click the Open and Save tabs. Deselect the .txt extension and enable the .html extension to change the type of the files you create and save.

## ***Color Coding***

There are many editors you can use to edit your code. Some of the options are advanced as well. Advanced text editors have many benefits that make the editors easy to handle and work with. One of the top benefits is color-coding. Whether you are writing your code for an HTML or CSS page or opening an existing page for editing, you will see code in different colors. This helps in the visual separation of different codes, making it easier to edit and change. The tags packed up with values and attributes have different types of colors, making it easier to locate a certain type of code. It also can help you spot and rectify errors. If you forget to add a quotation mark and closing bracket, it will change the color of the codes. You will know right away that you have made a mistake. If you miss out on a closing bracket or some quotation mark, it will cause the content to be painted in different colors than if you had applied the tags correctly. After you have been accustomed to colored syntax, you will trace the same colors to spot errors in your HTML documents. Most of the modern text editors allow you to customize the syntax color. You can choose the colors of your choice.

## ***Visual Aids***

Advanced editing tools include line numbers in the documents that are opened for editing sessions. These line numbers help orient a person in the code and help in collaboration with developers and designers. The tools make it easy for you to see line breaks that otherwise stay invisible.

## ***Completion***

Most of the text editors automate different tedious tasks like typing brackets and common tags. The shortcuts and methods for different programs vary, but most of the fully-featured text editors keep this capability. Some of the text editors trigger the code's completion by choosing the rest of the code from a list of options. This speeds up coding as you don't have to remember the code in its entirety. Coupled with this, most text editors automate repetitive tasks. They pace up coding by automating clips, macros, snippets, and other codes. These tools allow you to include reusable pieces of code on your pages. This saves you a great deal of time as well.

## Chapter Two: Introducing CSS

CSS is used for styling web pages. It is the short form of Cascading Style Sheets. It is used to describe how you can display HTML elements on your screen or on any other media. It will save you a lot of work when you are designing a website. You can control the web layout of more than one page at the same time. The external stylesheets will be stored in all the CSS files.

CSS defines the style of your web page for different layouts and variations of screen sizes. When it hit the market, HTML was not intended to contain the tags that would format your web pages. It was mainly created for describing how the content of your website will be displayed in the browser. Later on the tags `<font>` and `<color>` were added to HTML, which turned out to be a nightmare for web developers. The development of huge websites where information about the fonts and colors were added to single pages turned out to be expensive and long. It was this problem that CSS solved. It removed the styling feature on HTML pages.

The codes are saved with the `.css` extension. When you have created an external stylesheet file, you will be able to change your website's look just by changing a single file.

## *CSS Syntax*

Each CSS rule contains a declaration block and a selector block. The selectors will point out HTML elements that you need to style. The declaration blocks will have declarations that semicolons should separate. Each declaration must include the name and value of a CSS property, and colons must separate them. Multiple CSS declarations ought to be separated by semicolons and declaration blocks. You must include them inside curly braces.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  color: green;
```

```
  text-align: left;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS Programming!</p>
```

```
<p>I have styled this paragraph with CSS coding.</p>
```

```
</body>
```

```
</html>
```

---

Learning CSS Programming!

I have styled this paragraph with CSS coding.

In the above example, p is the selector that points to HTML elements that you must style. The color is a property of CSS, and green is the value of that property. The text-align again is a CSS property, while the left is the value of that property. This is the basic structure of a CSS code. I will write all the codes in the same structure. The only difference is the length and objective of the codes.

## ***CSS Selectors***

As you have seen in the example, a selector in CSS is used to select a certain HTML element that you need to style. CSS selectors can be divided into the following categories.

- CSS has simple selectors that are used to select elements that are based on the name, the id, and the class.
- You can use combinatory selectors that will select elements by analyzing first the relationship between them.
- Pseudo-class selectors are based on a certain state of HTML elements.
- Attribute selectors focus on the attribute and the value of the attribute.
- Pseudo-elements selectors focus on the style of a certain element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  text-align: left;
```

```
  color: grey;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>I will apply the style to all the paragraphs in this code.</p>
```

```
<p id="para1">Paragraph no 1!</p>
```

```
<p>Paragraph no 2</p>
```

```
</body>
```

</html>

---

I will apply the style to all the paragraphs in this code.

Paragraph no 1!

Paragraph no 2

The id selector will use the HTML element's id attribute for selecting a certain element of the HTML code. The id of HTML elements is unique; that's why the id selector selects a single element. If you want to select a certain element with a unique ID, you can use the # character and id of the element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#paragraph1 {  
  text-align: left;  
  color: green;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p id="paragraph1">Learning CSS!</p>
```

```
<p>I have not applied the style to this paragraph.</p>
```

```
</body>
```

</html>

---

## Learning CSS!

I have not applied the style to this paragraph.

You cannot start the name of an ID with a number.

## ***Class Selector***

The CSS class selector will select all the HTML elements with the help of a certain class attribute. You can write period to and the class's name to select different elements with a specific name.

```
<!DOCTYPE html>
<html>
<head>
<style>
.kaz {
  text-align: left;
  color: red;
}
</style>
</head>
<body>

<p class="kaz"> The CSS style applies to this paragraph. </p>
<p class="kaz"> The CSS style applies to this paragraph </p>

</body>
</html>
```

---

**The CSS style applies to this paragraph.**

**The CSS style applies to this paragraph**

You can have the liberty to specify the HTML elements that you want to be affected by this class.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz {
```

```
  text-align: left;
```

```
  color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4 class="kaz"> The CSS style does not apply to this paragraph. </h4>
```

```
<p class="kaz"> The CSS style applies to this paragraph </p>
```

```
</body>
```

```
</html>
```

---

***The CSS style does not apply to this paragraph.***

**The CSS style applies to this paragraph**

Also, you can add more classes to a single HTML element. This will give a unique look to your web page. It will also make styling easy and fun.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz {
  text-align: left;
  color: blue;
}
```

```
p.large {
  font-size: 200%;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4 class="kaz">I will not change this heading</h4>
```

```
<p class="kaz">I will turn this paragraph red and align it to the left side of
the page.</p>
```

```
<p class="kaz large">I will turn this paragraph red and align it to the left side
of the page. The only difference is that it will be in a larger font size.</p>
```

```
</body>
```

```
</html>
```

---

***I will not change this heading***

I will turn this paragraph red and align it to the left side of the page.

**I will turn this paragraph red and align it to the left side of the page. The only**

difference is that it will be in a larger font size.

## ***CSS Universal Selector***

The CSS universal selector can be identified with the \* sign. It selects each element on a web page. The styling will blanket all the elements and give them a universal appeal.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
* {  
  text-align: left;  
  color: blue;  
}
```

```
* {  
  font-size: 130%;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4 class="kaz">I will change this heading as well</h4>
```

```
<p class="kaz">I will turn this paragraph red and align it to the left side of  
the page. Now it will also be in a larger font size.</p>
```

```
<p class="kaz large">I will turn this paragraph red and align it to the left side  
of the page. It will be in a larger font size.</p>
```

```
</body>
```

</html>

---

*I will change this heading as well*

I will turn this paragraph red and align it to the left side of the page. Now it will also be in a larger font size.

I will turn this paragraph red and align it to the left side of the page. It will be in a larger font size.

## ***The Grouping Selector***

CSS grouping selector applies the style to all the HTML elements that have the same definitions of style. In the following example, I will write a code and implement a CSS grouping selector to all the elements inside the HTML document. It is a better option to form a group of selectors. It helps cut down on the code. I will place a comma to separate each of the selectors in the code.

```
<!DOCTYPE html>
<html>
<head>
<style>
h4, p, p {
  text-align: left;
  color: green;
}
</style>
</head>
<body>

<h4>Learning CSS Group Selectors</h4>
<p>The CSS style applies to this paragraph!</p>
<p>This paragraph will inherit the same CSS style!</p>

</body>
</html>
```

---

***Learning CSS Group Selectors***

The CSS style applies to this paragraph!

This paragraph will inherit the same CSS style!

## ***Adding CSS***

Whenever a browser reads a style sheet, it gets ready to format the HTML document based on the information-packed up in the style sheet. There are three different ways to insert CSS in an HTML document. These are the internal CSS method, external CSS method, and inline CSS method.

## ***The External CSS***

With an external style sheet, you may shift the entire website's look by changing a single file. An HTML page needs to integrate a clear reference to an external style sheet. Please remember that you must put the reference inside the head of the HTML document.

```
<!DOCTYPE html>
<html>
<head>
<link rel="This is a stylesheet" href="kazstyle.css">
</head>
<body>

<h4>I am writing a heading 4</h4>
<p>The style will apply to this paragraph.</p>

</body>
</html>
```

---

### ***I am writing a heading 4***

The style will apply to this paragraph.

You can write the external CSS sheet in any text editor you like. You must save it with the .css extension. This external .css extension file must not contain an HTML tag. There should be no spaces between the unit and the value of the property.

## ***The Internal CSS***

The second type of CSS integration is through internal styling. You can use the internal style sheet if a single HTML page carries a unique style. The internal style sheet may be defined in the <style> element that should be placed inside the HTML document head.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
  background-color: green;
```

```
}
```

```
p {
```

```
  color: blue;
```

```
  margin-left: 60px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4> Learning CSS Internal Styling</h4>
```

```
<p>I am styling the HTML document through internal styling.</p>
```

```
<p>This paragraph carries the same style.</p>
```

```
</body>
```

</html>

---

## ***Learning CSS Internal Styling***

I am styling the HTML document through internal styling.  
This paragraph carries the same style.

## *Inline CSS Styling*

The third method is inline CSS styling. You can apply a different style to a single element. While you are using inline styles, you can add a style attribute to the most relevant element. You can add a CSS property to your style attributes.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="color:red;text-align:left;">The Earth is round.</p>
```

```
<p style="color:blue;">Don't believe in the dull concept of Flat Earth.</p>
```

```
</body>
```

```
</html>
```

---

The Earth is round.

Don't believe in the dull concept of Flat Earth.

## ***Multiple Style Sheets***

If you have defined some properties for the same element in multiple style sheets, the editor will use the last read style sheet's value. If you define the internal style sheet after inserting the external style sheet, the latter will be applied.

```
<!DOCTYPE html>
<html>
<head>
<link rel="This is a css stylesheet" type="text/css" href=kazstyle.css">
<style>
h4 {
  color: blue;
}
</style>
</head>
<body>

<h4>Multiple Styling</h4>
<p>You are seeing a combination of external and internal stylesheets.
Change the color and produce different results</p>

</body>
</html>
```

---

### ***Multiple Styling***

You are seeing a combination of external and internal stylesheets.

## Change the color and produce different results

If you define the internal style before that link to external style sheet, the color will change.

```
<!DOCTYPE html>
<html>
<head>
<style>
h4 {
  color: blue;
}
</style>
<link rel="This is a css stylesheet" type="text/css" href=kazstyle.css">
</head>
<body>

<h4>Multiple Styling</h4>
<p>You are seeing a combination of external and internal stylesheets.
Change the color and produce different results</p>

</body>
</html>
```

---

### ***Multiple Styling***

You see a combination of external and internal stylesheets. Change the color and produce different results

What style do you have to use when you find more than one style for a

specific HTML element? There will be a cascading of a new virtual style sheet, which will have different rules.

- The first rule is inline styling, which will be done inside the HTML element.
- The second rule is internal and external style sheets inside of the head section.
- The third rule is browser default.

The inline style is of top priority. It will override the internal and external styles in a web browser. It will also override the browser defaults.

## Comments

CSS comments cannot be displayed inside the browser; however, they support the documentation of the source code. You can use comments for adding brief explanations to the code or when you edit your source code later. Browsers are used to ignore comments. A comment must be placed inside `<style>` brackets. It must start with `/*`, and it must end with `*/`. In the following example, I will add one comment on the top of the code. See how you can add comments to a CSS code.

```
<!DOCTYPE html>
<html>
<head>
<style>
/* I am learning CSS to design my websites.*/
p {
  color: green;
}
</style>
</head>
<body>

<p>Learning CSS!</p>
<p>I am styling this paragraph with CSS.</p>
<p>CSS does not allow the comments to be shown on the web browser.</p>

</body>
</html>
```

---

## Learning CSS!

I am styling this paragraph with CSS.

CSS does not allow the comments to be shown on the web browser.

You can change the position of the comment at will. When you are writing a complex code, you need comments to explain the code so that when you come back to see the code at a later date, you find it easy to scan it.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
/* I am learning CSS to design my websites.*/
```

```
p {
```

```
    color: green; /* I have set the color to green. Please don't repeat the same  
    color later in the code. Choose red or blue instead*/
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS!</p>
```

```
<p>I am styling this paragraph with CSS.</p>
```

```
<p>CSS does not allow the comments to be shown on the web browser.</p>
```

```
</body>
```

```
</html>
```

---

Learning CSS!

I am styling this paragraph with CSS.

CSS does not allow the comments to be shown on the web browser.

## ***Background Colors***

It is nice to change the color of the background when you are designing a website. Different pages need different colors as per the structure and content on them. With CSS, you can add different colors to your web pages. This makes it easy to identify your web pages.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h4 style="background-color:SkyBlue;">Learning CSS</h4>
```

```
<p style="background-color:pink;">
```

In this code, I will change the background of the two HTML elements. Changing the color of the background does not change the color of the text of the elements.

```
</p>
```

```
</body>
```

```
</html>
```

---

## ***Learning CSS***

In this code, I will change the background of the two HTML elements. Changing the color of the background does not change the color of the text of the elements.

## ***Text Color***

You also can change the color of the text of HTML elements. For that purpose, I will have to change the position of the code. The following example will explain how you can change the color of the code.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h4 style="color:SkyBlue;">CSS Text Colors</h4>
```

```
<p style="color:Pink;">You can see that the position of the code has changed. Now it is integrated into the HTML concerned.</p>
```

```
<p style="color:SeaGreen;">This is the easiest and the most efficient way to add color to your texts throughout a web page. You can give a unique color to each element on your page in case you need it.</p>
```

```
</body>
```

```
</html>
```

---

## ***CSS Text Colors***

You can see that the position of the code has changed. Now it is integrated into the HTML concerned.

This is the easiest and the most efficient way to add color to your texts throughout a web page. You can give a unique color to each element on your page in case you need it.

## ***Border Colors***

Although border colors are not common for web pages, CSS provides you the most effective way to add borders to HTML elements and paint them in your desired colors. When you write the code, you will have the opportunity to set the width of the borders. You can give alternate thickness to your borders to distinguish them from one another. This comes in handy when you have to emphasize a certain element on the web page. For example, when you design a landing page, you can use CSS's border feature to enclose and color your Call To Action to make it prominent and eye-catching. Just like we inserted the code inside individual elements to color the texts, we can use the same technique to create and add color to CSS borders. See the following example to know how to do that.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="border: 6px solid pink;">CSS Border Color</p>
```

```
<p style="border: 4px solid SkyBlue;">CSS Border Color</p>
```

```
<p style="border: 2px solid Violet;">CSS Border Color</p>
```

```
</body>
```

```
</html>
```

---

CSS Border Color

CSS Border Color

CSS Border Color

## ***Alternate Coloring Method***

There is another way to add colors in CSS. You can use HEX values, RGB values, RGBA values, HSL values, and HSLA values.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>The following will display the same color as name "Tomato":</p>
```

```
<p style="background-color:rgb(255, 99, 71);">This color is the rgb(255, 99, 71) This decreases the chances of mistake when you are working on a professional website.</p>
```

```
<p style="background-color:#ff6347;">This is the second method with a code #ff6347</p>
```

```
<p style="background-color:hsl(9, 100%, 64%);">This is the hsl method (9, 100%, 64%)</p>
```

```
<p>The following will display the same color as "Tomato", but it will be 50% transparent:</p>
```

```
<p style="background-color:rgba(255, 99, 71, 0.5);">This is the exact number: rgba(255, 99, 71, 0.5)</p>
```

```
<p style="background-color:hsla(9, 100%, 64%, 0.5);">This is the second method: hsla(9, 100%, 64%, 0.5)</p>
```

```
<p>You can have predefined color names, and you also can specify the colors by using RGB, HSL, HEX, or transparent colors with the help of RGBA or HSLA color values.</p>
```

```
</body>
```

```
</html>
```

---

The following will display the same color as the name "Tomato":

This color is the `rgb(255, 99, 71)` This decreases the chances of mistakes when you are working on a professional website.

This is the second method with code `#ff6347`

This is the hsl method `(9, 100%, 64%)`

The following will display the same color as "Tomato," but it will be 50% transparent:

This is the exact number: `rgba(255, 99, 71, 0.5)`

This is the second method: `hsla(9, 100%, 64%, 0.5)`

You can have predefined color names, and you also can specify the colors by using RGB, HSL, HEX, or transparent colors with the help of RGBA or HSLA color values.

## RGB Colors

RGB is the short form of the three major colors Red, Green, and Blue. These three colors, in science, are considered as the source of light. I have used RGB pattern of colors in the past examples. Each parameter of this color formula defines the intensity of each color in the combination between the 0 to 255 value. If you set the first unit at 255 and the other two at zero, the HTML element will give off intense red color.

If you have to display black color, you have to set the colors' parameters to 0 like `rgb(0, 0, 0)`. If you want to display white, you have to set the parameters to 255 like `rgb(255, 255, 255)`. To further explain the concept, I will alter the values of `rgb` in different ways, and you will see a variety of colors on the web screen. This appears to be a bit technical and highly professional to add

colors to your web page.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:rgb(255, 0, 0);">You will see extreme red color using the code: rgb(255, 0, 0)</p>
```

```
<p style="background-color:rgb(0, 0, 255);">This code will produce intense blue color: rgb(0, 0, 255)</p>
```

```
<p style="background-color:rgb(60, 185, 113);">Now that we have mixed up the value, you will see sea green color: rgb(60, 179, 113)</p>
```

```
<p style="background-color:rgb(238, 130, 238);">There is another mix of value and it is likely to produce purple color: rgb(200, 100, 250)</p>
```

```
<p style="background-color:rgb(255, 165, 0);">This mix will produce orange color: rgb(255, 110, 0)</p>
```

```
<p style="background-color:rgb(106, 90, 205);">This mix produces violet: rgb(110, 90, 200)</p>
```

```
<p>When you are working in HTML, you may specify the colors by using RGB values.</p>
```

```
</body>
```

```
</html>
```

---

You will see extreme red color using the code: rgb(255, 0, 0)

This code will produce intense blue color: rgb(0, 0, 255)

Now that we have mixed up the value, you will see sea green color:

rgb(60, 179, 113)

There is another mix of value, and it is likely to produce purple color: rgb(200, 100, 250)

This mix will produce orange color: rgb(255, 110, 0)

This mix produces violet: rgb(110, 90, 200)

When you are working in HTML, you may specify the colors by using RGB values.

If you love grey color, you can use RGB coding to produce different shades of grey for your web page. See the following:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:rgb(0, 0, 0);">You will be able to see the text because it will paint the background pitch black: rgb(0, 0, 0)</p>
```

```
<p style="background-color:rgb(60, 60, 60);">This is a bit lighter than the first but you will not see any text due to black shade: rgb(60, 60, 60)</p>
```

```
<p style="background-color:rgb(120, 120, 120);">As the values are increasing, you will see a lighter shade: rgb(120, 120, 120)</p>
```

```
<p style="background-color:rgb(180, 180, 180);">This is lighter than the previous: rgb(180, 180, 180)</p>
```

```
<p style="background-color:rgb(240, 240, 240);">This is much better: rgb(240, 240, 240)</p>
```

```
<p style="background-color:rgb(255, 255, 255);">As values are at the maximum, the background is completely white: rgb(255, 255, 255)</p>
```

```
<p>When you are using equal values for red, green, and blue, you will produce different shades of gray. Inequality in values will not give you grey
```

color</p>

<p style="background-color:rgb(255, 100, 255);">Unequal values can ruin the grey shade and a new color will pop up as a result: rgb(255, 100, 255)</p>

</body>

</html>



This is a bit lighter than the first, but you will not see any text due to the black shade: rgb(60, 60, 60)

As the values are increasing, you will see a lighter shade: rgb(120, 120, 120)

This is lighter than the previous: rgb(180, 180, 180)

This is much better: rgb(240, 240, 240)

As values are at the maximum, the background is completely white: rgb(255, 255, 255)

When you are using equal values for red, green, and blue, you will produce different shades of gray. Inequality in values will not give you grey color

Unequal values can ruin the grey shade, and a new color will pop up as a result: rgb(255, 100, 255)

## ***RGBA Colors***

RGBA color values appear to be an extension of RGB values with the alpha channel. It specifies color opacity. RGBA is the short form of (red, green, blue, alpha). An alpha parameter is a number in between 0.0 fully transparent and 1.0, which is not transparent. I will keep the color values the same and change only the value of alpha. The results are delightful to watch.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:rgba(255, 70, 190, 0);">When the alpha is zero, the background is totally transparent: rgba(255, 70, 190, 0)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 0.2);">The opacity slightly decreases when you increase the value of alpha: rgba(255, 70, 190, 0.2)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 0.4);">This is more opaque but still lighter: rgba(255, 70, 190, 0.4)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 0.6);">Keep increasing the value and you will see darker background without changing the colors: rgba(255, 70, 190, 0.6)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 0.8);">Getting dense: rgba(255, 70, 190, 0.8)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 1);">This is the densest of all: rgba(255, 70, 190, 1)</p>
```

```
<p style="background-color:rgba(255, 70, 190, 2);">Value 1 for alpha is the maximum value. You cannot increase it any further. Even if you do, it has no effect on the transparency of the background: rgba(255, 70, 190, 2)</p>
```

```
<p>This is the most professional method to manipulate colors without changing the original values.</p>
```

```
</body>
```

```
</html>
```

---

When the alpha is zero, the background is transparent: `rgba(255, 70, 190, 0)`

The opacity slightly decreases when you increase the value of alpha: `rgba(255, 70, 190, 0.2)`

This is more opaque but still lighter: `rgba(255, 70, 190, 0.4)`

Keep increasing the value, and you will see darker background without changing the colors: `rgba(255, 70, 190, 0.6)`

Getting dense: `rgba(255, 70, 190, 0.8)`

This is the densest of all: `rgba(255, 70, 190, 1)`

Value 1 for alpha is the maximum value. You cannot increase it any further. Even if you do, it has no effect on the transparency of the background: `rgba(255, 70, 190, 2)`

This is the most professional method to manipulate colors without changing the original values.

## HEX Colors

Hex denotes hexadecimal colors. A hexadecimal color is specified with a hexadecimal number. The six-digit number can be divided into pairs of two reds, two greens, and two blues. The hexadecimal integers specify the color components.

The values can be set in the form of 00 and ff. The ff value is the maximum value of 255. The HEX colors are a mix of numbers and alphabets. See the following code to understand how you can exploit HEX and create a

delightful mix of colors.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:#ff0000;">As the first two spots are designated red, ff means 255 so you will see dense red: #ff0000</p>
```

```
<p style="background-color:#0000ff;">The last two spots are designated blue. With double f, you will see dense blue: #0000ff</p>
```

```
<p style="background-color:#3cb371;">This is a mix and it will produce sea green: #3cb371</p>
```

```
<p style="background-color:#gg10ff;">You cannot use any alphabet after f. If you do, you will see no color on the screen. If no color perturbs you, look out for the g, z or q. #gg10ff</p>
```

```
<p style="background-color:#ffa500;">Get ready for the orange: #ffa500</p>
```

```
<p style="background-color:#6a5acd;">Violet is here. This might perplex you as to how you will know which color combination produces what kind of results, however, the more practice, the better combinations you will be able to produce: #6a5acd</p>
```

```
</body>
```

```
</html>
```

---

As the first two spots are designated red, ff means 255, so you will see dense red: #ff0000

The last two spots are designated blue. With double f, you will see dense blue: #0000ff

This is a mix, and it will produce sea green: #3cb371

You cannot use any alphabet after f. If you do, you will see no color on the screen. If no color perturbs you, look out for the g, z, or q. #gg10ff

Get ready for the orange: #ffa500

Violet is here. This might perplex you as to how you will know which color combination produces what kind of results; however, the more practice, the better combinations you will be able to produce: #6a5acd

You may see in the CSS source a three-digit hex code as well. The shorter code is a shorthand for the six-digit codes. The single digits represent r, g, and b. In the shorter code, one value is used for double values. The code #ff00cc is the same as #f0c.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
  background-color: #fd2; /* This is the same as #ffdd22 */
```

```
}
```

```
h4 {
```

```
  color: #f0a; /* This is the same as #f0a */
```

```
}
```

```
p {
```

```
  color: #c54; /* This is the same as #cc5544 */
```

```
}  
</style>  
</head>  
<body>  
  
<h4>Learning CSS three-digit Hexadecimal Code</h4>  
<p>Let us see how the three-digit code applies to this paragraph.</p>  
  
</body>  
</html>
```

---

### ***Learning CSS 3-digit Hexadecimal Code***

Let us see how the three-digit code applies to this paragraph.

## ***HSL Colors***

The CSS HSL is the short form of hue, saturation, and lightness. You can specify a color by using the hue, saturation, and lightness in the HSL order. Hue ranges from 0 to 360, where 0 denotes red, 120 denotes green, and 240 denotes blue. The value of saturation is in percentage form. 0% saturation means a gray shade, and 100% means full color. Lightness also comes in the form of percentage where 0% is black, 50% is somewhere in the middle of dark and light, and 100% is completely white. You can change the values and experiment with the colors.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:hsl(0, 100%, 50%);">This HSL code produces  
dense red color: hsl(0, 100%, 50%)</p>
```

```
<p style="background-color:hsl(240, 100%, 50%);">As I said 240 will bring  
blue, you will see dense blue: hsl(240, 100%, 50%)</p>
```

```
<p style="background-color:hsl(147, 50%, 47%);">Time to turn the  
background sea green. You can change the values to change the density of  
sea green color: hsl(147, 50%, 47%)</p>
```

```
<p style="background-color:hsl(300, 76%, 72%);">Purple color: hsl(300,  
76%, 72%)</p>
```

```
<p style="background-color:hsl(39, 100%, 50%);">Time to bring about  
orange color: hsl(39, 100%, 50%)</p>
```

```
<p style="background-color:hsl(248, 53%, 58%);">Violet, we have here:  
hsl(248, 53%, 58%)</p>
```

```
<p style="background-color:hsl(248, 100%, 100%);">There is no violet  
when I changed the saturation and lighting: hsl(248, 53%, 58%)</p>
```

```
<p style="background-color:hsl(248, 100%, 50%);">Violet turns blue just
```

because of saturation: hsl(248, 53%, 58%)</p>

</body>

</html>

---

This HSL code produces dense red color: hsl(0, 100%, 50%)

As I said, 240 will bring blue, and you will see dense blue: hsl(240, 100%, 50%)

Time to turn the background sea green. You can change the values to change the density of sea green color: hsl(147, 50%, 47%)

Purple color: hsl(300, 76%, 72%)

Time to bring about orange color: hsl(39, 100%, 50%)

Violet, we have here: hsl(248, 53%, 58%)

There is no violet when I changed the saturation and lighting: hsl(248, 53%, 58%)

Violet turns blue just because of saturation: hsl(248, 53%, 58%)

Saturation does wonders. It changes the color even if the hue is the same. You can describe saturation as the measure of gauging the intensity of the color. At 100%, it is in the purest color. There are no gray shades. 50% means it gives room to 50% gray shade; however, you still can see the original color. At 0%, saturation is fully gray. You are no longer able to see the original color.

<!DOCTYPE html>

<html>

<body>

<p style="background-color:hsl(0, 100%, 45%);">The Red Hot Shade: hsl(0, 100%, 45%)</p>

<p style="background-color:hsl(0, 80%, 50%);">A bit Grayish Red: hsl(0, 80%, 45%)</p>

<p style="background-color:hsl(0, 60%, 50%);">A blend of red and gray: hsl(0, 60%, 45%)</p>

<p style="background-color:hsl(0, 40%, 50%);">Grey starts prevailing over red: hsl(0, 40%, 45%)</p>

<p style="background-color:hsl(0, 20%, 50%);">Red has almost vanished giving room to Gray: hsl(0, 20%, 45%)</p>

<p style="background-color:hsl(0, 0%, 50%);">There is no red and only gray on the screen: hsl(0, 0%, 45%)</p>

<p>With HSL colors, you can set the tone of the color and create color blends by increasing and decreasing the saturation. 0% saturation is full gray.</p>

</body>

</html>

---

The Red Hot Shade: hsl(0, 100%, 45%)

A bit Grayish Red: hsl(0, 80%, 45%)

A blend of red and gray: hsl(0, 60%, 45%)

Grey starts prevailing over red: hsl(0, 40%, 45%)

Red has almost vanished, giving room to Gray: hsl(0, 20%, 45%)

There is no red and only gray on the screen: hsl(0, 0%, 45%)

With HSL colors, you can set the tone of the color and create color

blends by increasing and decreasing the saturation. 0% saturation is full gray.

HSL is the favorite of programmers because they allow them to experiment without changing the original colors. Just like saturation, you can change the lightness of the color. At 0%, it means no light. 50% means half-light, and 100% means full light. You will see white color on the screen.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:hsl(150, 100%, 0%);">At 0%, your website appears pitch black. Not recommended! hsl(0, 100%, 0%)</p>
```

```
<p style="background-color:hsl(150, 100%, 25%);">Dense green pops up: hsl(0, 100%, 25%)</p>
```

```
<p style="background-color:hsl(150, 100%, 50%);">Increasing the lightness brings parrot color out of dense green: hsl(0, 100%, 50%)</p>
```

```
<p style="background-color:hsl(150, 100%, 75%);">Sea green appears on further elevation of lightness: hsl(0, 100%, 75%)</p>
```

```
<p style="background-color:hsl(150, 100%, 90%);">skyblue emerges when you elevate it to 90%: hsl(0, 100%, 90%)</p>
```

```
<p style="background-color:hsl(150, 100%, 100%);">At 100%, there is no color on the screen: hsl(0, 100%, 100%)</p>
```

```
</body>
```

```
</html>
```



Dense green pops up: hsl(0, 100%, 25%)

Increasing the lightness brings parrot color out of dense green: hsl(0, 100%, 50%)

Sea green appears on further elevation of lightness: hsl(0, 100%, 75%)

sky blue emerges when you elevate it to 90%: hsl(0, 100%, 90%)

At 100%, there is no color on the screen: hsl(0, 100%, 100%)

You also can add alpha to HSL colors. It makes the coloring process more interesting and amazing. The HSL code turns into HSLA when it includes the alpha channel. The alpha channel will tune the opacity of the color. The value 0 for alpha denotes a transparent background, while the value 1 for alpha denotes opacity.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p style="background-color:hsla(150, 100%, 50%, 0);">Full Transparent display</p>
```

```
<p style="background-color:hsla(150, 100%, 50%, 0.2);">A light skyblue color when you increase the transparency</p>
```

```
<p style="background-color:hsla(150, 100%, 50%, 0.4);">Skyblue turning sea green</p>
```

```
<p style="background-color:hsla(150, 100%, 50%, 0.6);">Darker sea green</p>
```

```
<p style="background-color:hsla(150, 100%, 50%, 0.8);">Dense sea green</p>
```

```
<p style="background-color:hsla(150, 100%, 50%, 1);">Denser sea green when the opacity is fully restored.</p>
```

</body>

</html>

---

Full Transparent display

A light sky blue color when you increase the transparency

Skyblue turning sea green

Darker sea green

Dense sea green

Denser sea green when the opacity is fully restored.

## Chapter Three: CSS Backgrounds

With CSS, you can experiment with different background effects for certain elements. We have already worked on how to add and change the color of the background in CSS. Coupled with that fact, you can add a property to set the opacity of the background. The opacity property takes any value between 0 and 1. The lower the value is, the more transparent the background is. It works just like alpha value, but it has the status of a property. This means it can survive independently in the CSS code.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: skyblue;
}

div.fst {
    opacity: 0.1;
}

div.Snd {
    opacity: 0.3;
}

div.Trd {
```

```
    opacity: 0.6;
}
</style>
</head>
<body>
```

```
<p>CSS Background Properties</p>
```

```
<p>When you are using the CSS opacity property for adding transparency to the background of a certain HTML element, all the child elements it contains become transparent. It uniformly applies to each of them, making the text inside a transparent element. If you get it wrong, your users will find it hard to read:</p>
```

```
<div class="fst">
  <h4>Setting opacity at 0.1</h4>
</div>
```

```
<div class="Snd">
  <h4>Setting opacity at 0.3</h4>
</div>
```

```
<div class="Trd">
  <h4>Setting opacity at 0.6</h4>
</div>
```

```
<div>
  <h4>Setting opacity at 1. This also is the default value</h4>
```

</div>

</body>

</html>

---

## ***CSS Background Properties***

When you are using the CSS opacity property for adding transparency to the background of a certain HTML element, all the child elements it contains become transparent. It uniformly applies to each of them, making the text inside a transparent element. If you get it wrong, your users will find it hard to read:

***Setting opacity at 0.1***

***Setting opacity at 0.3***

***Setting opacity at 0.6***

***Setting opacity at 1. This also is the default value***

You may find some differences in the display in the book. However, when you use the code in an editor, you will see the exact results. Use the code and upload it to a browser to get the results.

## ***Images***

Another CSS property for backgrounds is the image property. You can specify how you can use images as the background of your web page. The image, by default, is always repeated so that it can fully cover the entire element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
  background-image: url("skyfall.gif");
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>CSS Background Images!</h4>
```

```
<p>This code will set an image as the background for your web page!</p>
```

```
</body>
```

```
</html>
```

---

### ***CSS Background Images!***

This code will set an image as the background for your web page!

The most important thing to keep in mind when using images for the

background is that you should not use an image that meddles with the text and makes it unreadable. You can set the background image for certain elements without displaying it on the entire web page. The background-image property can repeat the property in vertical and horizontal styles. Some of the images ought to be repeated at a certain angle, or they look strange. You can choose not to repeat the background image by using the no-repeat image property. You also can position the background image to the top or bottom corners by using specific properties.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    background-image: url("skyfall.gif");
```

```
    background-repeat: no-repeat;
```

```
    background-position: right bottom;
```

```
    margin-right: 400px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>CSS Background Images!</h4>
```

```
<p>This code will set an image as the background for your web page!</p>
```

```
</body>
```

</html>

---

## ***CSS Background Images!***

This code will set an image as the background for your web page!

Background attachment property tends to specify if the image must scroll or stay fixed, which means it will not scroll down or up when a user scrolls the other part of the page.

## ***CSS Borders***

The CSS border properties are interesting as they allow you to fix the width, style, and color of an HTML element's border.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.dot {border-style: dotted;}
```

```
p.dash {border-style: dashed;}
```

```
p.sol {border-style: solid;}
```

```
p.doub {border-style: double;}
```

```
p.groov {border-style: groove;}
```

```
p.ridg {border-style: ridge;}
```

```
p.ins {border-style: inset;}
```

```
p.out {border-style: outset;}
```

```
p.non {border-style: none;}
```

```
p.hid {border-style: hidden;}
```

```
p.mixed {border-style: dashed dotted solid double;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>You are learning border-style Properties</h4>
```

```
<p>You can design your website with a variety of borders:</p>
```

```
<p class="dot">This paragraph has a dotted border.</p>
```

```
<p class="dash">This paragraph has a dashed border.</p>
<p class="sol">This paragraph has a solid border.</p>
<p class="doub">This paragraph has a double border.</p>
<p class="groov">This paragraph has a groove border.</p>
<p class="ridg">This paragraph has a ridged border.</p>
<p class="ins">This paragraph has an inset border.</p>
<p class="out">This paragraph has an outset border.</p>
<p class="non">This paragraph lacks a border.</p>
<p class="hid">This paragraph has a hidden border.</p>
<p class="mixed">This paragraph has a blended border.</p>

</body>
</html>
```

---

## ***You are learning border-style Properties***

You can design your website with a variety of borders:

This paragraph has a dotted border.
This paragraph has a dashed border.
This paragraph has a solid border.
This paragraph has a double border.
This paragraph has a groove border.
This paragraph has a ridged border.
This paragraph has an inset border.

This paragraph has an outset border.

This paragraph lacks a border.

This paragraph has a hidden border.

This paragraph has a blended border.

## Width

You can set and change the borders' width by specifying the size in em, px, cm, and pt. These measures will make the borders thin, thick, or make them medium.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.dot {border-style: dotted; border-width: 10px;}
```

```
p.dash {border-style: dashed; border-width: medium;}
```

```
p.sol {border-style: solid; border-width: 4px;}
```

```
p.doub {border-style: double; border-width: thick;}
```

```
p.groov {border-style: groove; border-width: thin;}
```

```
p.ridg {border-style: solid; border-width: 15px;}
```

```
p.ins {border-style: solid; border-width: 5px 15px;} /*5px top right, 15px  
bottom left */
```

```
p.out {border-style: solid; border-width: 5px 20px;}
```

```
p.non {border-style: solid; border-width: 5px 15px 20px 5px;} /* 5px top,  
15px right, 20px bottom, 5px left */
```

```
p.hid {border-style: hidden;}
```

```
p.mixed {border-style: dashed dotted solid double;}
```

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

<h4>Testing different border widths types. </h4>
<p>I will also use multiple border widths:</p>

<p class="dot">This paragraph has a dotted border.</p>
<p class="dash">This paragraph has a dashed border.</p>
<p class="sol">This paragraph has a solid border.</p>
<p class="doub">This paragraph has a double border.</p>
<p class="groov">This paragraph has a groove border.</p>
<p class="ridg">This paragraph has a ridged border.</p>
<p class="ins">This paragraph has an inset border.</p>
<p class="out">This paragraph has an outset border.</p>
<p class="non">This paragraph lacks a border.</p>
<p class="hid">This paragraph has a hidden border.</p>
<p class="mixed">This paragraph has a blended border.</p>

</body>
</html>
```

---

### ***Testing different border widths types.***

I will also use multiple border widths:

This paragraph has a dotted border.

This paragraph has a dashed border.

This paragraph has a solid border.

This paragraph has a double border.

This paragraph has a groove border.

This paragraph has a ridged border.

This paragraph has an inset border.

This paragraph has an outset border.

This paragraph lacks a border.

This paragraph has a hidden border.

This paragraph has a blended border.

This is how you can change the width of the border in a single line of code. I have tested multiple width sizes for each side of the border through different methods. You can choose any method as suits you when you are designing your website.

## Border Colors

You can use the CSS border color property to set the color of the borders. You can use HEX, HSL, or RGB to specify the color. You also can use the name of the color. If you do not set the color, it will inherit the color of the HTML element concerned. I will use all types of coloring patterns to color the borders in my example to find it easy to use all the methods when you are designing a web page. One interesting thing about color property is that you can give different colors to each side of the border.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
<style>
p.dot {border-style: dotted; border-width: 10px; border-color: green;}
p.dash {border-style: dashed; border-width: medium; border-color: skyblue;}
p.sol {border-style: solid; border-width: 4px; border-color: skyblue green;}
p.doub {border-style: double; border-width: thick; border-color: skyblue red
yellow green;}
p.groov {border-style: groove; border-width: thick; border-color: orange blue
green;}
p.ridg {border-style: ridge; border-width: 15px; border-color: #aa5588;}
p.ins {border-style: inset; border-width: 5px 15px; border-color: rgb(100,
255, 50);} /*5px top right, 15px bottom left */
p.out {border-style: outset; border-width: 5px 20px; border-color: hsl(0,
100%, 50%);}
p.non {border-style: none; border-width: 5px 15px 20px 5px; border-color:
transparent;} /* 5px top, 15px right, 20px bottom, 5px left */
</style>
</head>
<body>
```

```
<h4>Testing different border widths types. </h4>
```

```
<p>I will also use multiple border widths:</p>
```

```
<p class="dot">Dotted border and color property.</p>
```

```
<p class="dash">Dashed border in skyblue color.</p>
```

```
<p class="sol">This solid border has two colors.</p>
```

```
<p class="doub">Double border with four different colors.</p>
```

```
<p class="groov">This paragraph has a solid border painted in three different colors.</p>
```

```
<p class="ridg">Ridged border in purple.</p>
```

```
<p class="ins">Inset border in rgb colors.</p>
```

```
<p class="out">Outset border in HSL colors.</p>
```

```
<p class="non">This paragraph lacks border color.</p>
```

```
</body>
```

```
</html>
```

---

### *Testing different border widths types.*

I will also use multiple border widths:

Dotted border and color property.

Dashed border in sky blue color.

This solid border has two colors.

Double border with four different colors.

This paragraph has a solid border painted in three different colors.

Ridged border in purple.

Inset border in rgb colors.

Outset border in HSL colors.

This paragraph lacks border color.

## Border Sides

You can design a border in different patterns by adding different styles to the same. Just like we added multiple colors to borders, we can design them in multiple ways. You can give different designs to the top, left, bottom, and right sides of the border.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.kaz {
  border-top-style: solid; border-color: skyblue;
  border-right-style: dotted; border-color: green;
  border-bottom-style: double; border-color: skyblue;
  border-left-style: ridge; border-color: green;
}
</style>
</head>
<body>

<h4>I am creating Individual Border Sides</h4>
<p class="kaz">The paragraph will have four different border styles.</p>

</body>
</html>
```

---

***I am creating Individual Border Sides***

The paragraph will have four different border styles.

You must keep in mind that there are some rules for designing borders.

Rule 1: The first rule is that if you add four values and order them as ridge dotted solid dashed, the border at the top will be ridged, the border to the right will be dotted, the border at the bottom end will be solid, and the border to the left will be dashed.

Rule 2: If you add three values and order them as solid dotted dashed, the top one will be solid, the right and left ones will be dotted, and the bottom one will be dashed.

Rule 3: If you add only solid and dashed, the top and bottom will be solid, and the right and left will be dashed.

Rule4: If you add just solid, all sides of the border will be solid.

## **Rounded Borders**

A special property of borders is for rounded borders. You can turn a square border into a round border with a special CSS property. The property is named as border-radius.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz {
```

```
border-top-style: solid; border-color: skyblue; border-radius: 15px;
```

```
border-right-style: dotted; border-color: green border-radius: 15px;
```

```
border-bottom-style: double; border-color: skyblue border-radius: 15px;
```

```
border-left-style: ridge; border-color: green border-radius: 15px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>I am creating Individual Border Sides</h4>
```

```
<p class="kaz">The paragraph will have four different border styles and rounded corners.</p>
```

```
</body>
```

```
</html>
```

## Chapter Four: CSS Margins & Padding

You can use CSS margins to create some space around different elements out of the borders you have already defined.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  margin: 100px;
  border: 4px solid #5DAD60;
}
</style>
</head>
<body>

<h4>Learning CSS Margins</h4>
<div>This HTML element will have 100px margins.</div>

</body>
</html>
```

---

### *Learning CSS Margins*

This HTML element will have 100px margins.

CSS margins have multiple properties. You can use them to build space around certain elements out of the borders. You can control the margins as you like. You can set up the margins for different sides of the top, bottom,

left, and right elements. You can specify the margins for each side in an individual line of code. The browser will calculate the margins, specify the lengths and widths of the element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 5px solid skyblue;
  margin-top: 150px;
  margin-bottom: 150px;
  margin-right: 150px;
  margin-left: 80px;
  background-color: lightblue;
}
</style>
</head>
<body>

<p>I am applying individual CSS margin properties</p>

<div>This specific div element will have the top margin of 150px, the right
margin of 150px, the bottom margin of 150px, and the left margin of 80px.
</div>

</body>
</html>
```

---

I am applying individual CSS margin properties

This specific div element will have the top margin of 150px, the right margin of 150px, the bottom margin of 150px, and the left margin of 80px.

The following example contains the CSS inherit feature for margins. See how it works.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 5px solid skyblue;
  margin-left: 50px;
}

p.kaz {
  margin-left: inherit;
}
</style>
</head>
<body>
```

```
<h4>I am experimenting on the inherit value of margins</h4>
```

```
<p>I will let the left margin to inherit by the parent HTML element:</p>
```

```
<div>
```

```
<p class="kaz">This CSS paragraph keeps an inherited left margin right from  
the div element.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

### *I am experimenting on the inherit value of margins*

I will let the left margin inherit by the parent HTML element:

This CSS paragraph keeps an inherited left margin right from the div element.

### **Margin Collapse Feature**

The top and bottom margins of HTML elements are bound to be collapsed in the form of margins equal to the widest of the two margins. This usually does not happen on the right and left margins but does happen to the bottom and top margins.

## ***CSS Padding***

CSS padding adds an air of spaciousness to the content of the HTML element within the confines of the borders.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
padding: 50px;
border: 5px solid #4DAB90;
}
</style>
</head>
<body>

<h4>I am learning CSS Padding</h4>
<div>This HTML element contains a 100px padding.</div>

</body>
</html>
```

You get full control of the properties of CSS padding. You can use it to generate space about the content of the element within the defined borders. In CSS programming, you can use properties for setting up padding for different elements to the top, bottom, right, and left sides. You can specify padding for different properties. The CSS padding properties have the length property used to specify the padding in cm, px, and pt. The width is specified in percentage. Like margins, this too has the inheritance feature.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
border: 4px solid skyblue;
background-color: lightblue;
padding-top: 40px;
padding-right: 50px;
padding-bottom: 30px;
padding-left: 60px;
}
</style>
</head>
<body>
```

```
<p>I will use individual properties of CSS padding</p>
```

```
<div>This specific div element contains the top padding of 40px, the right
padding of 50px, the bottom padding of 30px, and the left padding of 60px.
</div>
```

```
</body>
</html>
```

---

I will use individual properties of CSS padding

This specific div element contains the top padding of 40px, the right padding of 50px, the bottom padding of 30px, and the left padding of 60px.

Another important feature is the width feature. Adding the width property measures and defines how much width the HTML element will have. When you define the width, the padding is added to the width property, which produces unexpected results.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div.kaz {
```

```
  width: 400px;
```

```
  background-color: skyblue;
```

```
}
```

```
div.kaz1 {
```

```
  width: 350px;
```

```
  padding: 30px;
```

```
  background-color: yellow;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS padding and width features</p>
```

```
<div class="kaz">This specific div will be 400px wide.</div>
```

```
<br>
```

```
<div class="kaz1">The div will have 400px, even though it has 350px in the CSS.</div>
```

```
</body>
```

```
</html>
```

---

Learning CSS padding and width features

This specific div will be 400px wide.

The div will have 400px, even though it has 350px in the CSS.

If you want to keep the width at 350px, you will have to include the box-sizing property. This will maintain the width of the element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div.kaz {
```

```
  width: 400px;
```

```
  background-color: skyblue;
```

```
}
```

```
div.kaz1 {  
  width: 300px;  
  padding: 30px;  
  box-sizing: border-box;  
  background-color: yellow;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS padding and width features</p>
```

```
<div class="kaz">This specific div will be 400px wide.</div>
```

```
<br>
```

```
<div class="kaz1">The div will have 400px, even though it has 300px in the  
CSS.</div>
```

```
</body>
```

```
</html>
```

---

Learning CSS padding and width features

This specific div will be 400px wide.

The div will have 400px, even though it has 300px in the CSS.



## ***CSS Width & Height***

You can add further style to your website by using the width and length properties of CSS. Let us have a demonstration of the properties and see how they work.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
    height: 100px;
```

```
    width: 150%;
```

```
    border: 1px solid #6DCA90;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS width & height properties</p>
```

```
<div>The HTML element under demonstration has a height of 100 pixels and  
a width of 150%.</div>
```

```
</body>
```

```
</html>
```

---

This is CSS width & height properties

The HTML element under demonstration has a height of 100 pixels and a width of 150%.

These properties do not include borders, padding, and margins. It will set the width and height of an area within the border, padding, and element margins. You can set the value to auto to set up the default settings. The browser will calculate the width and height and adjust itself to give the most responsive display. The length value will define the width and height of the element in cm and px. The initial value is used to set the width or height to the default values. The inherit value will trigger the inherit feature of the property and allot the element the parent value.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
    height: 100px;
```

```
    width: 150%;
```

```
    background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS width & height properties</p>
```

```
<div>The HTML element under demonstration has a height of 100 pixels and a width of 150%.</div>
```

```
</body>
```

```
</html>
```

---

This is CSS width & height properties

The HTML element under demonstration has a height of 100 pixels and a width of 150%.

I am using px feature instead of percentage to make it more precise and efficient.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  height: 100px;
```

```
  width: 150px;
```

```
  background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS width & height properties</p>
```

```
<div>The HTML element under demonstration has a height of 100 pixels and
```

```
a width of 150%.</div>
```

```
</body>
```

```
</html>
```

---

This is CSS width & height properties

The HTML element under demonstration has a height of 100 pixels and a width of 150%.

You can set up the width at maximum by using the max-width property. You can specify this property in the length values such as cm and px or % of the block. If the browser is smaller than your HTML element's width, there will be a horizontal scrollbar on your page. The max-width feature will handle this situation well. It will improve the responsiveness of the website. The max-width feature overrides the width feature.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
    max-width: 500px;
```

```
    height: 100px;
```

```
    background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS max-width property</p>
```

```
<div>The HTML element under demonstration has a height of 100 pixels and  
a max-width of 500px.</div>
```

```
</body>
```

```
</html>
```

---

This is CSS max-width property

The HTML element under demonstration has a height of 100 pixels  
and a max-width of 500px.

## Chapter Five: CSS Boxes

The term box model in CSS is an integral part of the website design and layout. The box model in CSS is a box that encloses an HTML element. It encompasses borders, margins, padding, and the actual content. A CSS box model has several parts. One of the most important parts is content, where you can insert the images you have to display on the website and the text you have to put in the box. The second component is padding that clears all the areas of the content. Padding is always transparent.

The third part of the box model is borders. The border traces around the content of the model and the padding as well. It encloses them completely. The fourth section of the box model is named margin. It clears up areas around the border and creates space to make the box look tidy and formatted. The margin is always transparent. The box model tends to add a border around this element and define the designated space between all the elements. In the following example, I will create a box model.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: grey;
  width: 200px;
  border: 20px solid skyblue;
  padding: 40px;
  margin: 30px;
}
</style>
</head>
```

```
<body>
```

```
<h4>Learning the Box Model</h4>
```

```
<p>If we dive deeper into the CSS box model, it turns out to be a box that engulfs each HTML element you want to emphasize. </p>
```

```
<div>This is the place where you have to fill in the text. It has borders, appropriate padding, perfect margins, and some text you want your users to read. It can be a product description.</div>
```

```
</body>
```

```
</html>
```

---

## ***Learning the Box Model***

If we dive deeper into the CSS box model, it turns out to be a box that engulfs each HTML element you want to emphasize.

This is the place where you have to fill in the text. It has borders, appropriate padding, perfect margins, and some text you want your users to read. It can be a product description.

If you want to learn how the box model works, you should consider how to correctly set up the height and width of the element. When you set up the height and width of an HTML element, you set up the area's height and width. If you want to calculate the size of the element, you need to add padding, margins, and borders. You should calculate the total width of a certain element by the following method.

The total width is equal to right padding, left padding, width, right border, left border, right margin, and left margin. The total height is calculated by measuring the height, bottom padding, top padding, bottom margin, top

margin, bottom border, and top border.

## ***The Outline***

A CSS outline is drawn out of the border of an element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  border: 4px solid grey;
```

```
  outline: #9FFA99 solid 20px;
```

```
  margin: auto;
```

```
  padding: 40px;
```

```
  text-align: left;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS Outline for boxes</h4>
```

```
<p>This text will appear in the box. It has a 4px grey border and a light green  
outline with 40px width.</p>
```

```
</body>
```

```
</html>
```

---

***Learning CSS Outline for Boxes***

This text will appear in the box. It has a 4px grey border and a light green outline with 40px width.

An outline is drawn around an HTML element. It is traced outside of the borders so that the element stands out of the rest of the page. It is used to emphasize a certain piece of text, an image, or a random piece of information. Outline should not be confused with borders because, unlike CSS borders, it is traced out of the border of an element. It may overlap with the content as well. It is not part of the dimensions of an element. The width and height of an element should not be affected by an outline's width. CSS outline feature has different properties, which gives programmers the freedom to experiment. Just like we changed the look and design of CSS borders, we can change CSS outlines' look.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {outline-color:skyblue;}
```

```
p.dot {outline-style: dotted;}
```

```
p.dash {outline-style: dashed;}
```

```
p.sol {outline-style: solid;}
```

```
p.doub {outline-style: double;}
```

```
p.groov {outline-style: groove;}
```

```
p.rid {outline-style: ridge;}
```

```
p.ins {outline-style: inset;}
```

```
p.out {outline-style: outset;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS outline-style Property</p>
```

```
<p class="dot"> The dotted outline </p>
```

```
<p class="dash"> The dashed outline </p>
```

```
<p class="sol"> The solid outline </p>
```

```
<p class="doub"> The double outline </p>
```

```
<p class="groov"> The groove outline. </p>
```

```
<p class="rid"> The ridge outline. </p>
```

```
<p class="ins"> The inset outline. </p>
```

```
<p class="out"> The outset outline. </p>
```

```
</body>
```

```
</html>
```

---

Learning CSS outline-style Property

The dotted outline

The dashed outline

The solid outline

The double outline

The groove outline.

The ridge outline.

The inset outline.

The outset outline.

Unless you have set up the outline property, other properties will have no effect. First, set it up and then apply other properties to get full function and least errors in the code.

## Width Property

CSS outline feature has the width property to make an outline thick and thin as per need. The width property specifies how much wide an outline should be. It can have the following values. If you select thin, it will be equal to 1px. If you select medium, it will be about 3px. If you select thick, it will be about 5px. You can set up a specific size by selecting any figure in px, em, pt, and cm.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.kaz1 {
  border: 2px solid grey;
  outline-style: solid;
  outline-color: blue;
  outline-width: thin;
}

p.kaz2 {
  border: 2px solid black;
  outline-style: solid;
  outline-color: skyblue;
  outline-width: medium;
```

```
}
```

```
p.kaz3 {  
  border: 2px solid yellow;  
  outline-style: solid;  
  outline-color: red;  
  outline-width: thick;  
}
```

```
p.kaz4 {  
  border: 2px solid green;  
  outline-style: solid;  
  outline-color: green;  
  outline-width: 6px;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>I am learning on the outline-width Property</p>
```

```
<p class="kaz1">Producing a thin outline.</p>
```

```
<p class="kaz2">Producing a medium outline.</p>
```

```
<p class="kaz3">This is a thick outline.</p>
```

```
<p class="kaz4">I am setting up 6px thickness for this outline.</p>
```

```
</body>
```

```
</html>
```

---

I am learning on the outline-width Property

Producing a thin outline.

Producing a medium outline.

This is a thick outline.

I am setting up 6px thickness for this outline.

## Color Property

You can use a CSS outline property to define or change the color of all the HTML elements on your website. You can use HEX, HSL, and RGB formulas. Writing names of colors is also standard practice.

The invert value executes colors' inversion, ensuring that the outline stays visible in an utter disregard to background shade.

In this code example, I will explain the code to award multiple colors to different types of outlines.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz1 {
```

```
border: 2px solid grey;
```

```
outline-style: solid;
```

```
outline-color: blue;
```

```
outline-width: thin;
}
```

```
p.kaz2 {
border: 2px solid black;
outline-style: solid;
outline-color: skyblue;
outline-width: medium;
}
```

```
p.kaz3 {
border: 2px solid yellow;
outline-style: solid;
outline-color: #ff0000;
outline-width: thick;
}
```

```
p.kaz4 {
border: 2px solid green;
outline-style: solid;
outline-color: rgb(50, 150, 255);
outline-width: 6px;
}
```

```
p.kaz5 {
```

```
border: 2px solid green;
outline-style: solid;
outline-color: hsl(255, 100%, 50%);
outline-width: 6px;
}
```

```
p.kaz6 {
border: 2px solid yellow;
outline-style: solid;
outline-color: invert;
outline-width: 6px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>I am learning on the outline-color Property</p>
```

```
<p class="kaz1">Color Names.</p>
```

```
<p class="kaz2">Color Names.</p>
```

```
<p class="kaz3">HEX colors.</p>
```

```
<p class="kaz4">rgb colors for outline.</p>
```

```
<p class="kaz5">HSL colors.</p>
```

```
<p class="kaz6">Inverting Colors.</p>
```

```
</body>
```

```
</html>
```

---

I am learning on the outline-color Property

Color Names.

Color Names.

HEX colors.

rgb colors for outline.

HSL colors.

Inverting Colors.

Instead of writing separate code lines, CSS offers a shorthand property that you can use to save time when writing lengthy code. However, you should only use it when you become an expert in coding. I will convert the above-mentioned example into short-hand code.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz1 {outline: solid blue thin;}
```

```
p.kaz2 {outline: solid skyblue medium;}
```

```
p.kaz3 {outline: dotted #ff0000 thick;}
```

```
p.kaz4 {outline: dashed rgb(50, 150, 255) 6px;}
```

```
p.kaz5 {outline: solid 2px green;}
```

```
p.kaz6 {outline: solid invert 6px;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>I am learning on the outline-color Property</p>
```

```
<p class="kaz1">Color Names.</p>
```

```
<p class="kaz2">Color Names.</p>
```

```
<p class="kaz3">HEX colors.</p>
```

```
<p class="kaz4">rgb colors for outline.</p>
```

```
<p class="kaz5">HSL colors.</p>
```

```
<p class="kaz6">Inverting Colors.</p>
```

```
</body>
```

```
</html>
```

---

I am learning on the outline-color Property

Color Names.

Color Names.

HEX colors.

rgb colors for outline.

HSL colors.

Inverting Colors.

## Outline Offset Property

This CSS property creates space between the border of an HTML element and an outline. The space is transparent all the time. You can decide how much space you need to add between the outline and border. I will explain by an example how much gap you can create and how much is enough.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
  margin: 40px;
```

```
  border: 2px dashed grey;
```

```
  outline: 1px solid red;
```

```
  outline-offset: 20px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning CSS outline-offset Property</p>
```

```
<p>You will see an outline 20px out of the edge of the border.</p>
```

```
</body>
```

```
</html>
```

---

## Learning CSS outline-offset Property

You will see an outline 20px out of the edge of the border.

When you produce the results in a browser, you will notice that the space between the outline and the element is transparent.

## ***CSS Texts***

When you are developing and designing a website, you have to work on formatting the texts and filling them up inside boxes to display them in a better way. CSS text formatting is an important feature because, without text and content, web design is meaningless. The following example will explain how many options you have to format the text.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  border: 2px solid blue;
```

```
  padding: 6px;
```

```
}
```

```
h4 {
```

```
  text-align: left;
```

```
  text-transform: uppercase;
```

```
  color: #8FDA99;
```

```
}
```

```
p {
```

```
  text-indent: 60px;
```

```
  text-align: justify;
```

```
  letter-spacing: 4px;
```

```
}
```

```
a {
  text-decoration: none;
  color: #990FDF;
}
</style>
</head>
<body>

<div>
  <h4>learning CSS text formatting feature</h4>
  <p>This text has a different formatting style. It has different properties like
text-align, color properties, and text-transform.

  I have properly indented the paragraph, aligned it on the page, and specified
the space in between the characters.

  <a target="_blank" href="pleasetryit.asp?filename=trycss_text">"Please
Give It A Try"</a> link.</p>
</div>

</body>
</html>
```

---

## ***LEARNING CSS TEXT FORMATTING FEATURE***

This text has a different formatting style. It has different properties like text-align, color properties, and text-transform. I have properly indented the paragraph, aligned it on the page, and specified the

space in between the characters.

You can use the color property to set the color of your text. You can use RGB, HEX, and color names to include in the property. You can add the background and text color properties to the code to change the color of the background and text on your web page. The process is simple. You just need to add the color properties to the code and see the results.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  border: 2px solid blue;
  padding: 6px;
}

h4 {
  text-align: left;
  text-transform: uppercase;
  color: #8FDA99;
  background-color: lightgrey;
  color: seagreen;
}

p {
  text-indent: 60px;
  text-align: justify;
  letter-spacing: 4px;
```

```
background-color: skyblue;
color: black;
}
```

```
a {
  text-decoration: none;
  color: #990FDF;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div>
```

```
<h4>learning CSS text formatting feature</h4>
```

```
<p>This text has a different formatting style. It has different properties like
text-align, color properties, and text-transform.
```

```
  I have properly indented the paragraph, aligned it on the page, and specified
  the space in between the characters.
```

```
    <a target="_blank" href="pleasetryit.asp?filename=trycss_text">"Please
  Give It A Try"</a> link.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

***LEARNING CSS TEXT FORMATTING FEATURE***

This text has a different formatting style. It has different properties like text-align, color properties, and text-transform. I have properly indented the paragraph, aligned it on the page, and specified the space in between the characters.

You can see that the text-alignment property tunes in the alignment of the text as per your needs. You can set up the alignment to horizontal and vertical as per your needs. You may set up the text as centered, right-aligned, or justified. When you set the alignment to justify, each line of the code will stretch so that it has equal width. The right and left margins will be straight as they appear in the newspapers and magazines.

## Text Direction

You can use Unicode-bidi and direction properties to alter the direction of the text of an element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.kaz1 {
```

```
  direction: rtl;
```

```
  unicode-bidi: bidi-override;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is going to be the default direction of the text.</p>
```

```
<p class="kaz1">Your text will have a right-to-left direction.</p>
```

```
</body>
```

```
</html>
```

---

This is going to be the default direction of the text.

Your text will have a right-to-left direction

## Vertical Alignment

This vertical-align property will set up vertical alignment for your HTML elements. You can add an image and set up vertical alignment for that.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img.topside {  
  vertical-align: top;  
}
```

```
img.middleside {  
  vertical-align: middle;  
}
```

```
img.bottomside {
```

```
vertical-align: bottom;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Your  image  
has default alignment.</p><br>
```

```
<p>Your  image will have top alignment.</p><br>
```

```
<p>Your  image has middle level alignment.</p><br>
```

```
<p>Your  image has the bottom alignment.</p>
```

```
</body>
```

```
</html>
```

---

Your  
alignment.

image has default

Your  
alignment.

image will have top

Your  
alignment.

image has middle-level

Your  
alignment.

image has the bottom

## Decoration

You can decorate your text in CSS by using the text-decoration property. This property sets up decorations and removes them as well. The value ' none ' for the property removes all the decorations.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.k {
```

```
  text-decoration: overline;
```

```
}
```

```
p.l {
```

```
  text-decoration: line-through;
```

```
}
```

```
p.m {
```

```
  text-decoration: underline;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class= "k">We have the first paragraph</p>
```

```
<p class= "l">This is the second paragraph</p>
```

```
<p class= "m">Working on the third paragraph</p>
```

```
</body>
```

```
</html>
```

---

We have the first paragraph

~~This is the second paragraph~~

Working on the third paragraph

## Text Transformation

This CSS property aims at specifying lowercase and uppercase letters in a particular text. You can use it to turn text into lowercase and uppercase letters or capitalize a particular word's maiden letter.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
<style>
p.ucase {
  text-transform: uppercase;
}

p.lcase {
  text-transform: lowercase;
}

p.capital {
  text-transform: capitalize;
}
</style>
</head>
<body>

<p class="ucase">This text is converted into uppercase.</p>
<p class="lcase">Formatting it in the lower case.</p>
<p class="capital">Capital case always looks well on a web page.</p>

</body>
</html>
```

---

THIS TEXT IS CONVERTED INTO UPPERCASE.

formatting it in the lower case.

Capital Case Always Looks Well On A Web Page.

## Indentation

You can indent your text as well if the design demands that. The text-indent property specifies the indentation of the very first line of the text.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p {
```

```
    text-indent: 60px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>I am writing this paragraph to show how you can indent paragraphs in CSS programming. Indentation is imperative in CSS because it makes the text look tidy on the web page. As the attention span of people drops, indentation makes text readable and eye-friendly. I hope you use the property well on the next website you design.</p>
```

```
</body>
```

```
</html>
```

---

I am writing this paragraph to show how you can indent paragraphs in CSS programming. Indentation is imperative in CSS because it makes the text look tidy on the web page. As the attention span of people drops, indentation makes text readable and eye-friendly. I hope you use the property well on the next website you design.

## Letter-Spacing

This CSS property is used for the specification of some space in the characters of a particular piece of text.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.k {
```

```
    letter-spacing: 5px;
```

```
}
```

```
p.l {
```

```
    letter-spacing: -1px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="k">This paragraph has 5px spacing in the letters</p>
```

```
<p class="l">Time to shrink the spaces. Negative spacing is also quite a feature in designing. It glamorizes the look of the text. A further decrease will
```

make the text muddled up.</p>

</body>

</html>

---

This paragraph has 5px spacing in the letters

Time to shrink the spaces. Negative spacing is also quite a feature in designing. It glamorizes the look of the text. A further decrease will make the text muddled up.

## Height

You can set up the height of the line for the specification of space in the lines.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.k {
```

```
  line-height: 0.7;
```

```
}
```

```
p.l {
```

```
  line-height: 2.0;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="k">This paragraph has 0.7 height in the letters <br>  
See how the height changes in the paragraph.<br>  
</p>
```

```
<p class="l">Time to shrink the spaces.<br>  
Negative spacing is also quite a feature in designing.<br>  
It glamorizes the look of the text. A further decrease will make the text  
muddled up.<br>  
</p>
```

```
</body>
```

```
</html>
```

---

This paragraph has 0.7 height in the letters  
See how the height changes in the paragraph.

Time to shrink the spaces.

Negative spacing is also quite a feature in designing.

It glamorizes the look of the text. A further decrease will make the  
text muddled up.

## Word Spacing

The word-spacing property of CSS texts aims at specifying the space in the words of a particular piece of text. In the following example, I will raise and reduce the space in the words.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
p.k {
  word-spacing: 10px;
}
```

```
p.l {
  word-spacing: -7px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="k">See how the spacing changes affects the design of the
paragraph.
```

```
</p>
```

```
<p class="l">Time to shrink the spaces.<br>
```

```
Negative spacing is also quite a feature in designing.<br>
```

```
</p>
```

```
</body>
```

```
</html>
```

---

See how the spacing changes affects the design of the paragraph.

Time to shrink the spaces.

Negative spacing is also quite a feature in designing.

## Text Shadow

You can create a shadow for your text to add a bit of style to the web page. The text-shadow property does the job of adding the shadow. See how you can use the property and how it works.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-shadow: 1px 1px;
}
</style>
</head>
<body>

<p>Learning the Text-shadow property!</p>

</body>
</html>
```

---

Learning the Text-shadow property!

I will add color to the text by the following method.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
p {
  text-shadow: 1px 1px 5px yellow;
}
</style>
</head>
<body>

<p>Learning the Text-shadow property!</p>

</body>
</html>
```

---

Learning the Text-shadow property!

## ***CSS Fonts***

Choosing the right fonts for the website is crucial. You choose the wrong one, and you ruin the page design. The right font exerts a huge impact on the overall experience of readers who use your website. The right font creates a powerful identity for your brand. The font style should have a single objective which is that it should be easy to read. The style of your font adds great value to the text of the page. Coupled with the style of the font are the color and size of the text, which is important.

You will have five font families in CSS.

- The first font family is serif. This family is known for small strokes at the edge of each letter. They give off a powerful sense of elegance and formality, and you might have seen it used on sales ads. This family contains Times New Roman, Georgia, and Garamond.
- The second font family is sans-serif. It has clean lines and no small strokes with the letters. They are more modern looking and give off a minimalistic air. This font family contains Verdana, Helvetica, and Arial.
- The third font family is known as monospace fonts. All the letters that belong to this family carry fixed width. They give off a mechanical look. It has Monaco, Courier New, and Lucida Console.
- The fourth font family is known as cursive. Cursive fonts are known for their similarity to handwriting, and this family contains Lucida Handwriting and Brush Script M7.
- The fifth font family is called fantasy. These fonts are bold, playful, and highly decorative, and this family contains Copperplate and Papyrus.

When designing a website, you can use the font-family property to decide which font style you need to display your text on the web page. The font-family property ought to hold several font names in the fallback system to ensure that maximum browsers are compatible with your website. Start with the one of your choice and end with the one that belongs to a generic family. The browser will pick similar fonts from the generic family if it does not find the one you have put at the top. Separate them with a comma, and everything is fine. For font names that have more than one word, use quotes to enclose

them.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
  font-family: "Times New Roman", Times, serif;
```

```
}
```

```
.kaz2 {
```

```
  font-family: Arial, Helvetica, sans-serif;
```

```
}
```

```
.kaz3 {
```

```
  font-family: "Lucida Console", "Courier New", monospace;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-family</h4>
```

```
<p class="kaz1">I plan to display this paragraph in the Times New Roman font.</p>
```

```
<p class="kaz2">Arial font is the best in terms of readability.</p>
```

```
<p class="kaz3">Lucida Console font looks like something written with hands.</p>
```

```
</body>
```

```
</html>
```

---

## ***Learning CSS Font-Family***

I plan to display this paragraph in the Times New Roman font.

Arial font is the best in terms of readability.

Lucida Console font looks like something written with hands.

## **CSS Font Style**

Another CSS property font-style is used to specify the style of the text. It is mostly used to transform a piece of text into italic. The property contains three values. You can either keep the text as normal and italic or turn it oblique. The normal value turns a piece of text into normal, while the italic value turns a piece of text into italic, just like you see in MS Word. The oblique value turns a piece of text into an oblique. Oblique looks like italic, but it is not as supported as italic is. Therefore, do not confuse oblique with italic.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
    font-style: normal;
```

```
}
```

```
.kaz2 {
```

```
font-style: italic;
}
```

```
.kaz3 {
font-style: oblique;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-style</h4>
```

```
<p class="kaz1">I plan to display this paragraph in normal font style.</p>
```

```
<p class="kaz2">Italic font is the best in terms of style.</p>
```

```
<p class="kaz3">Oblique font looks much like italic.</p>
```

```
</body>
```

```
</html>
```

---

## ***Learning CSS Font-Style***

I plan to display this paragraph in normal font style.

*Italic font is the best in terms of style.*

*Oblique font looks much like italic.*

## **Font-Weight Property**

Another property font-weight decides upon the weight of the font style.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
  font-weight: normal;
```

```
}
```

```
.kaz2 {
```

```
  font-weight: lighter;
```

```
}
```

```
.kaz3 {
```

```
  font-weight: bold;
```

```
}
```

```
.kaz4 {
```

```
  font-weight: 1000;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-weight property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
<p class="kaz3">You can design each HTML element on a web page, giving it a unique look.</p>
```

```
<p class="kaz4">Fonts make or break your website so design the text carefully.</p>
```

```
</body>
```

```
</html>
```

---

## *Learning CSS Font-Weight Property*

CSS is the best web designing language.

Before you learn CSS, you must learn HTML.

**You can design each HTML element on a web page, giving it a unique look.**

**Fonts make or break your website so design the text carefully.**

## **Font-Variant Property**

The CSS font-variant property tends to specify the small-caps font of the text. When you apply the property, all the lowercase letters will be converted into uppercase letters. Still, the converted letters will appear in a smaller font than that of the original uppercase letters.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
font-variant: normal;
}
```

```
.kaz2 {
font-variant: small-caps;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-weight property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

## ***Learning CSS Font-Weight Property***

CSS is the best web designing language.

BEFORE YOU LEARN CSS, YOU MUST LEARN HTML.

## **Font-Size Property**

You can use the font-size property to set up the size of the font. Sometimes you need a larger font size to display something important like a sales pitch or some other important information. Font-size is no replacement for

displaying a heading. For that purpose, you should always stick to h1, h2, and h3 headers.

You can use the absolute size to set up the text of the font to a specific size. Absolute size keeps the users from changing the text size in different browsers. This is a bit unfair due to accessibility reasons. Absolute size is highly useful when you know the physical size of your output. The second option is of relative size. This sets the size of the font relative to the elements that surround the targeted piece of text. It also allows users to change the size of the text in a browser window. If you do not specify the size of the font, the default size will be 16px.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
    font-size: 40px;
```

```
}
```

```
.kaz2 {
```

```
    font-size: 80px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-weight property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

### *Learning CSS Font-Weight Property*

CSS is the best web designing language.

Before you learn CSS, you must learn HTML.

You can use the em measure to set the font size as well. Programmers prefer em over px to give users the provision to resize pieces of text. Since the default size is 16 px, 1 em is equal to 16px.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {  
  font-size: 1.5em;  
}
```

```
.kaz2 {  
  font-size: 2em;  
}
```

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

<h4>Learning CSS font-size property</h4>
<p class="kaz1">CSS is the best web designing language.</p>
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>

</body>
</html>
```

---

### *Learning CSS Font-Size Property*

CSS is the best web designing language.

Before you learn CSS, you must learn HTML.

Now I will add percentage calculation feature to measure the font sizes. See the following example.

```
<!DOCTYPE html>
<html>
<head>
<style>
.kaz1 {
  font-size: 200%;
```

```
}
```

```
.kaz2 {  
  font-size: 2em;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-size property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

### *Learning CSS Font-Size Property*

CSS is the best web designing language.

Before you learn CSS, you must learn HTML.

If you want to make your web page responsive to a different browser, you

can use the vw unit, which is the short form of the viewport width. Using the vw feature, the text size follows and tunes into the browser's size.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {  
  font-size: 10vw;  
}
```

```
.kaz2 {  
  font-size: 2em;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-size property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

***Learning CSS Font-Size Property***

CSS is the best web designing language.

# Before you learn CSS, you must learn HTML.

## CSS Font-Pairings

Great fonts are an integral part of a great design. A great combination of font pairings complements each other, which is great for a great website design. You should ensure that font combinations harmonize with one another. A superfamily of fonts is like a set of fonts that work together. Use fonts from the same superfamily to ensure a smooth display of web pages.

Similar fonts end up in conflict. Contrasts, if you do it the right way, makes the best result.

```
<!DOCTYPE html>
<html>
<head>
<style>
.kaz1 {
  font-family: Verdana, sans-serif;
  font-size: 60px;
  color: gray;
}

.kaz2 {
font-family: Georgia, serif;
  font-size: 20px;
```

```
color: red;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-size property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

## *Learning CSS Font-Size Property*

CSS is the best web designing language.

**Before you learn CSS, you must learn HTML.**

Let us take a look at another viable pair.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
font-family: Muli, serif;
font-size: 40px;
color: gray;
}
```

```
.kaz2 {
font-family: 'Fjalla One', sans-serif;
font-size: 20px;
color: red;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS font-size property</h4>
```

```
<p class="kaz1">CSS is the best web designing language.</p>
```

```
<p class="kaz2">Before you learn CSS, you must learn HTML.</p>
```

```
</body>
```

```
</html>
```

---

## *Learning CSS Font-Size Property*

# CSS is the best web designing

language.

Before you learn CSS, you must learn HTML.

## **Chapter Six: Lists, Tables, Displays**

HTML has two types of lists. The list properties of CSS let you set up different markers for the ordered lists and unordered lists. You can insert an image as well to set it as a marker. You can add colors to the backgrounds of the list items and lists.

## ***CSS Lists***

Let us jump to the CSS property `list-style-type` that specifies what type of list items your visitors will see on the web page. You can use more than one type to display the bullets.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.k1 {
  list-style-type: square;
}

ul.k2 {
  list-style-type: circle;
}

ol.k3 {
  list-style-type: upper-roman;
}

ol.k4 {
  list-style-type: lower-alpha;
}
</style>
</head>
<body>
```

#### <h4>Learning List Property</h4>

<p>You see an example of the unordered lists:</p>

```
<ul class="k1">  
  <li>marinda</li>  
  <li>seven up</li>  
  <li>pepsi</li>  
</ul>
```

```
<ul class="k2">  
  <li>avocado</li>  
  <li>dragon fruit</li>  
  <li>guava</li>  
</ul>
```

<p>This is an example of the ordered lists:</p>

```
<ol class="k3">  
  <li>strawberry</li>  
  <li>blueberry</li>  
  <li>raspberry</li>  
</ol>
```

```
<ol class="k4">  
  <li>oranges</li>  
  <li>apples</li>
```

```
<li>cherries</li>
</ol>

</body>
</html>
```

---

## ***Learning List Property***

You see an example of the unordered lists:

- marinda
- seven up
- pepsi
- avocado
- dragon fruit
- guava

This is an example of the ordered lists:

- I. strawberry
- II. blueberry
- III. raspberry
  - a. oranges
  - b. apples
  - c. cherries

## **Adding Colors**

You can decorate your lists with a series of colors to make them look

interesting. Anything that you add to the <ul> or <ol> tags will affect the complete list. However, the properties that you add to the <li> tag will only affect the list's individual items.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul.k1 {
```

```
    background: #ff1122;
```

```
    padding: 20px;
```

```
}
```

```
ul.k2 {
```

```
    background: #dd9911;
```

```
    padding: 20px;
```

```
}
```

```
ol.k3 {
```

```
    background: #bbe9e9;
```

```
    padding: 6px;
```

```
    margin-left: 25px;
```

```
}
```

```
ol.k4 {
```

```
    background: #aab9ee;
```

```
    margin: 4px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning List Property</h4>
```

```
<p>You are seeing an example of the unordered lists:</p>
```

```
<ul class="k1">
```

```
<li>marinda</li>
```

```
<li>seven up</li>
```

```
<li>pepsi</li>
```

```
</ul>
```

```
<ul class="k2">
```

```
<li>avocado</li>
```

```
<li>dragon fruit</li>
```

```
<li>guava</li>
```

```
</ul>
```

```
<p>This is an example of the ordered lists:</p>
```

```
<ol class="k3">
```

```
<li>strawberry</li>
```

```
<li>blueberry</li>
```

```
<li>raspberry</li>
```

```
</ol>
```

```
<ol class="k4">
  <li>oranges</li>
  <li>apples</li>
  <li>cherries</li>
</ol>
```

```
</body>
</html>
```

---

## *Learning List Property*

You see an example of the unordered lists:

- marinda
- seven up
- pepsi
- avocado
- dragon fruit
- guava

This is an example of the ordered lists:

1. strawberry
2. blueberry
3. raspberry
4. oranges
5. apples

).

cherries

## ***CSS Tables***

You can change the look of HTML tables with the help of CSS. They will look amazing, decorated, and very well-formatted.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#listofcustomers {  
    font-family: Helvetica, Arial, sans-serif;  
    border-collapse: collapse;  
    width: 75%;  
}
```

```
#listofcustomers td, #listofcustomers th {  
    border: 2px solid #aaa;  
    padding: 6px;  
}
```

```
#listofcustomers tr:nth-child(even){background-color: #e2e2e2;}
```

```
#listofcustomers tr:hover {background-color: #ddd;}
```

```
#listofcustomers th {  
    padding-top: 15px;  
    padding-bottom: 15px;  
    text-align: left;
```

```
background-color: #4DFA90;
color: grey;
}
</style>
</head>
<body>

<table id="customers">
  <tr>
    <th>Name of Customers</th>
    <th>Color Allotted</th>
    <th>Name of Country</th>
  </tr>
  <tr>
    <td>Alfredo</td>
    <td>Red</td>
    <td>Ireland</td>
  </tr>
  <tr>
    <td>Jasmine Buckky</td>
    <td>Green</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centrum Chuccky</td>
```

```
<td>Violet</td>
<td>Iceland</td>
</tr>
<tr>
<td>Fanny Line</td>
<td>Purple</td>
<td>Scotland</td>
</tr>
<tr>
<td>Fanny Brown</td>
<td>Yellow</td>
<td>Neverland</td>
</tr>
<tr>
<td>Peter</td>
<td>White</td>
<td>Treasure Island</td>
</tr>
<tr>
<td>Jackson Ramsey</td>
<td>Green</td>
<td>Namibia</td>
</tr>
<tr>
<td>Magot Rosy</td>
```

```
<td>Brown</td>
<td>Spain</td>
</tr>
<tr>
<td>Nany</td>
<td>Pink</td>
<td>Egypt</td>
</tr>
<tr>
<td>Paris Hilton</td>
<td>Tanned Pink</td>
<td>France</td>
</tr>
</table>

</body>
</html>
```

## The Border Property

You can add the border property to the code to create a beautiful and neat border to the result. There will be no messing up of the items in the table.

```
<!DOCTYPE html>
<html>
<head>
<style>
#listofcustomers {
```

```
font-family: Helvetica, Arial, sans-serif;
border-collapse: collapse;
width: 75%;
}
```

```
#listofcustomers td, #listofcustomers th {
border: 2px solid #aaa;
padding: 6px;
}
```

```
#listofcustomers tr:nth-child(even){background-color: #e2e2e2;}
```

```
#listofcustomers tr:hover {background-color: #ddd;}
```

```
#listofcustomers th {
padding-top: 15px;
padding-bottom: 15px;
text-align: left;
background-color: #4DFA90;
color: grey;
}
```

```
table, th, td {
border: 2px solid green;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table id="customers">
```

```
<tr>
```

```
<th>Name of Customers</th>
```

```
<th>Color Allotted</th>
```

```
<th>Name of Country</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Alfredo</td>
```

```
<td>Red</td>
```

```
<td>Ireland</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Jasmine Buckky</td>
```

```
<td>Green</td>
```

```
<td>Germany</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Centrum Chuccky</td>
```

```
<td>Violet</td>
```

```
<td>Iceland</td>
```

```
</tr>
```

```
<tr>
  <td>Fanny Line</td>
  <td>Purple</td>
  <td>Scotland</td>
</tr>
```

```
<tr>
  <td>Fanny Brown</td>
  <td>Yellow</td>
  <td>Neverland</td>
</tr>
```

```
<tr>
  <td>Peter</td>
  <td>White</td>
  <td>Treasure Island</td>
</tr>
```

```
<tr>
  <td>Jackson Ramsey</td>
  <td>Green</td>
  <td>Namibia</td>
</tr>
```

```
<tr>
  <td>Magot Rosy</td>
  <td>Brown</td>
  <td>Spain</td>
</tr>
```

```
<tr>
  <td>Nany</td>
  <td>Pink</td>
  <td>Egypt</td>
</tr>
<tr>
  <td>Paris Hilton</td>
  <td>Tanned Pink</td>
  <td>France</td>
</tr>
</table>

</body>
</html>
```

---

<b>Name of Customers</b>	<b>Color Allotted</b>	<b>Name of Country</b>
Alfredo	Red	Ireland
Jasmine Buckky	Green	Germany
Centrum Chuccky	Violet	Iceland
Fanny Line	Purple	Scotland
Fanny Brown	Yellow	Neverland
Peter	White	Treasure Island
Jackson Ramsey	Green	Namibia
Magot Rosy	Brown	Spain

Nany	Pink	Egypt
Paris Hilton	Tanned Pink	France

## The Width Property

The width property for CSS tables helps you set up the width of the columns and rows of your table. This will cut down the possibility of jumbling up of the columns. It will make the cases wide so that the text appears spacious. The table, after you add the width property, will expand to the full screen.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#listofcustomers {  
  font-family: Helvetica, Arial, sans-serif;  
  border-collapse: collapse;  
  width: 75%;  
}
```

```
#listofcustomers td, #listofcustomers th {  
  border: 2px solid #aaa;  
  padding: 6px;  
}
```

```
#listofcustomers tr:nth-child(even){background-color: #e2e2e2;}
```

```
#listofcustomers tr:hover {background-color: #ddd;}
```

```
#listofcustomers th {
  padding-top: 15px;
  padding-bottom: 15px;
  text-align: left;
  background-color: #4DFA90;
  color: grey;
}
```

```
table, th, td {
  border: 2px solid green;
}
```

```
table {
  width: 100%;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<table id="customers">
```

```
<tr>
```

```
<th>Name of Customers</th>
```

```
<th>Color Allotted</th>
```

```
<th>Name of Country</th>
```

```
</tr>
```

```
<tr>
```

<td>Alfredo</td>

<td>Red</td>

<td>Ireland</td>

</tr>

<tr>

<td>Jasmine Buckky</td>

<td>Green</td>

<td>Germany</td>

</tr>

<tr>

<td>Centrum Chuccky</td>

<td>Violet</td>

<td>Iceland</td>

</tr>

<tr>

<td>Fanny Line</td>

<td>Purple</td>

<td>Scotland</td>

</tr>

<tr>

<td>Fanny Brown</td>

<td>Yellow</td>

<td>Neverland</td>

</tr>

<tr>

```
<td>Peter</td>
<td>White</td>
<td>Treasure Island</td>
</tr>
<tr>
  <td>Jackson Ramsey</td>
  <td>Green</td>
  <td>Namibia</td>
</tr>
<tr>
  <td>Magot Rosy</td>
  <td>Brown</td>
  <td>Spain</td>
</tr>
<tr>
  <td>Nany</td>
  <td>Pink</td>
  <td>Egypt</td>
</tr>
<tr>
  <td>Paris Hilton</td>
  <td>Tanned Pink</td>
  <td>France</td>
</tr>
</table>
```

</body>

</html>

<b>Name of Customers</b>	<b>Color Allotted</b>	<b>Name of Country</b>
Alfredo	Red	Ireland
Jasmine Buckky	Green	Germany
Centrum Chuccky	Violet	Iceland
Fanny Line	Purple	Scotland
Fanny Brown	Yellow	Neverland
Peter	White	Treasure Island
Jackson Ramsey	Green	Namibia
Magot Rosy	Brown	Spain
Nany	Pink	Egypt
Paris Hilton	Tanned Pink	France

The height and width of CSS tables are usually defined as properties. The following example will set up the width of the table to 100%. You can set the height at whatever point you like.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#listofcustomers {
```

```
font-family: Helvetica, Arial, sans-serif;
border-collapse: collapse;
width: 75%;
}
```

```
#listofcustomers td, #listofcustomers th {
border: 2px solid #aaa;
padding: 6px;
}
```

```
#listofcustomers tr:nth-child(even){background-color: #e2e2e2;}
```

```
#listofcustomers tr:hover {background-color: #ddd;}
```

```
#listofcustomers th {
padding-top: 15px;
padding-bottom: 15px;
text-align: left;
background-color: #4DFA90;
color: grey;
}
```

```
table, th, td {
border: 2px solid green;
}
```

```
table {
```

```
width: 100%;
}
th {
height: 50px;
}
</style>
</head>
<body>

<table id="customers">
<tr>
<th>Name of Customers</th>
<th>Color Allotted</th>
<th>Name of Country</th>
</tr>
<tr>
<td>Alfredo</td>
<td>Red</td>
<td>Ireland</td>
</tr>
<tr>
<td>Jasmine Buckky</td>
<td>Green</td>
<td>Germany</td>
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

---

<b>Name of Customers</b>	<b>Color Allotted</b>	<b>Name of Country</b>
Alfredo	Red	Ireland
Jasmine Buckky	Green	Germany

You can see how spacious the first column looked when I increased the height of the first column.

## Aligning Table

CSS offers you a text-align property that sets up the horizontal alignment of content. The content will be aligned to the center, left or right. In the following example, I will add three aligning features in the same code: aligning the td, aligning the th, and aligning td in a vertical format.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
#listofcustomers {
```

```
font-family: Helvetica, Arial, sans-serif;
```

```
border-collapse: collapse;
```

```
width: 75%;
```

```
}
```

```
#listofcustomers td, #listofcustomers th {
```

```
  border: 2px solid #aaa;
```

```
  padding: 6px;
```

```
}
```

```
#listofcustomers tr:nth-child(even){background-color: #e2e2e2;}
```

```
#listofcustomers tr:hover {background-color: #ddd;}
```

```
#listofcustomers th {
```

```
  padding-top: 15px;
```

```
  padding-bottom: 15px;
```

```
  text-align: left;
```

```
  background-color: #4DFA90;
```

```
  color: grey;
```

```
}
```

```
table, th, td {
```

```
  border: 2px solid green;
```

```
}
```

```
table {
```

```
  width: 100%;
```

```
}
```

```
th {
```

```
    height: 50px;
}
td {
    text-align: center;
}
th {
    text-align: left;
}
td {
    height: 50px;
    vertical-align: top;
}
</style>
</head>
<body>
```

```
<table id="customers">
  <tr>
    <th>Name of Customers</th>
    <th>Color Allotted</th>
    <th>Name of Country</th>
  </tr>
  <tr>
    <td>Alfredo</td>
    <td>Red</td>
```

```
<td>Ireland</td>
</tr>
<tr>
  <td>Jasmine Buckky</td>
  <td>Green</td>
  <td>Germany</td>
</tr>
</table>

</body>
</html>
```

---

<b>Name of Customers</b>	<b>Color Allotted</b>	<b>Name of Country</b>
Alfredo	Red	Ireland
Jasmine Buckky	Green	Germany

## ***CSS Layout***

If you want to produce a clean layout in CSS, you need to use CSS's display property. This property can help you control the layout in a better way. This property specifies how an HTML element on your web page ought to be displayed. Each HTML element on your page carries a default display value based on the type of the element. The default display value for most of the elements is inline or block. The block elements take a new line to start. They also take up a full width that stretches to the right and left as far as possible.

The elements like <form>, <div>, <header>, <footer>, <p> and <h1> are block elements.

Block elements always take the option of full width, which is available. If you set up the block-level elements' width, it will keep you from stretching out to the extreme edges of the container. After that, you can set up your margins to auto and to horizontally center that specific element in the container. The specific element will inherit the width you have specified and the remaining space that will stand split up between margins.

You may hit upon a stumbling block if the browser window turns out to be smaller than the width of your HTML element. After that, the browser will add a horizontal scrollbar to your web page. By using max-width, you will be able to improve the process as to how your browser handles small windows. This is extremely important when you are making a website that you can use on small devices.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.kaz1 {
  width: 1000px;
  margin: auto;
  border: 5px solid #99FA99;
```

```
}
```

```
div.kaz2 {  
  max-width: 700px;  
  margin: auto;  
  border: 5px solid #73AD21;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning CSS Max-width Feature</h4>
```

```
<div class="kaz1">This specific div element carries 1000 px width;</div>
```

```
<br>
```

```
<div class="kaz2">This specific div element carries 700 px width;</div>
```

```
<p><strong>Pro Tip:</strong> You may go on and drag the browser window  
down to smaller than 500px wide in order to see the difference!</p>
```

```
</body>
```

```
</html>
```

---

## ***Learning CSS Max-width Feature***

This specific div element carries 1000 px width;

This specific div element carries 700 px width;

**Pro Tip:** You may go on and drag the browser window down to smaller than 500px wide to see the difference!

## Positioning Property

The CSS position property tends to specify which type of positioning method you are using for a certain element. You may use the relative, sticky, static, fixed, and absolute methods. Once you have decided which position you have to choose, you can position the HTML element using the right, left, top, and bottom properties. These properties, however, are not going to work unless you have set up the position property in the code. Also, they work according to the position value you fill in the code.

## Static

The first layout position for the website is static. All HTML elements on your website take the static position by default. These elements are the least affected by the left, top, bottom, and right properties. An HTML element with the static option is not positioned specially. It is rather positioned according to the page flow.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ThisIsStatic {
  position: static;
  border: 4px solid #FF9DA97;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Learning static position property;</h4>
```

```
<p>The best thing about the static position is that it is not positioned in a certain way. It molds its position according to the flow of your web page:</p>
```

```
<div class="ThisIsStatic">
```

```
  See how the static position works on this web page;
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## ***Learning Static Position Property***

The best thing about the static position is that it is not positioned in a certain way. It molds its position according to the flow of your web page:

See how the static position works on this web page;

## **Relative**

The second layout position is relative. It adjusts the HTML element relative to the normal position to an HTML element. Setting the bottom, top, left, and right properties of an HTML element that you have relatively positioned will

cause it to stand away from the normal position.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ThisIsRelative {
  position: relative;
  left: 40px;
  border: 4px solid black;
}
</style>
</head>
<body>
```

```
<h4>Learning Relative position property;</h4>
```

```
<p>The best thing about the static position is that it is not positioned in a certain way. It molds its position according to the flow of your web page:
</p>
```

```
<div class="ThisIsRelative">
```

```
  See how the static position works on this web page;
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## *Learning Relative Position Property;*

The best thing about the static position is that it is not positioned in a certain way. It molds its position according to the flow of your web page:

See how the static position works on this web page;

### **Fixed**

An HTML element that has a fixed position should be positioned relative to viewport. This means that the element always stays inside the same place even you scroll the page down to the bottom. You can use the right, left, top, and bottom properties to set up the HTML element position. A fixed HTML element does not leave a gap in your web page where it stays.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ThisIsFixed {
  position: fixed;
  bottom: 50;
  right: 0;
  width: 500px;
  border: 4px solid red;
}
</style>
</head>
<body>
```

```
<h4>Learning Fixed position property;</h4>
```

```
<p>CSS, if used properly, has the power to make your website highly responsive to different media devices and interactive as well:</p>
```

```
<div class="ThisIsFixed">
```

```
  This example shows how to adjust the layout on your website;
```

```
</div>
```

```
</body>
```

```
</html>
```

---

### ***Learning Fixed Position Property***

CSS, if used properly, has the power to make your website highly responsive to different media devices and interactive as well:

This example shows how to adjust the layout on your website;

If you give 0 value to the bottom property, the div element will sit at the bottom of the web page. No matter how much you scroll down, it will stay fixed and stay visible to the users. Be careful with using the fixed feature. Unless you have to show extremely useful information that users need to know when they stay on your website, do not use it. It takes no time to turn from a fun thing to an annoying thing to the point that users start hating your website.

### **Absolute**

An HTML element that has an absolute position is positioned in a relative position to the closest positioned ancestor. If your HTML element that is

absolute positioned lacks ancestors, it deploys the element's body, and it moves as you scroll the page. A positioned element is the one that has any position except static.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ThisIsRelative {
  position: relative;
  width: 500px;
  height: 100px;
  border: 5px solid red;
}
```

```
div.ThisIsAbsolute {
  position: absolute;
  top: 100px;
  right: 0;
  width: 300px;
  height: 100px;
  border: 5px solid yellow;
}
```

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

```
<h4>Learning Fixed position property;</h4>
```

```
<div class="ThisIsRelative">CSS, if used properly, has the power to make  
your website highly responsive to different media devices and interactive as  
well:</div>
```

```
<div class="ThisIsAbsolute">
```

This example shows how to adjust the layout on your website;

```
</div>
```

```
</body>
```

```
</html>
```

---

### ***Learning Fixed Position Property***

CSS, if used properly, has the power to make your website highly responsive to different media devices and interactive as well:

This example shows how to adjust the layout on your website;

### **Sticky**

An HTML element that has sticky property is adjusted as a user scroll downs a website. A sticky element tends to toggle between the fixed and relative properties.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
div.stickyscrolling {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 4px;
  background-color: skyblue;
  border: 2px solid grey;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning position property;</p>
```

```
<div class="stickyscrolling">This is the sticky scrolling!</div>
```

```
<div style="padding-bottom:2500px">
```

```
<p>CSS, if used properly, has the power to make your website highly responsive to different media devices and interactive as well:</p>
```

```
<p> This example shows how to adjust the layout on your website.</p>
```

```
<p> This is the end of the page. Scroll down to see if the line sticks to the top of the page or not. </p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## ***Learning Position Property***

This is the sticky scrolling!

CSS, if used properly, has the power to make your website highly responsive to different media devices and interactive as well:

This example shows how to adjust the layout of your website.

This is the end of the page. Scroll down to see if the line sticks to the top of the page or not.

If you think the webpage is not scrolling down, add more text to the code to enable scrolling down.

## **Elements Overlapping**

When you have positioned elements, they will overlap with different other HTML elements. An HTML element may have a negative or a positive stack order.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {
```

```
  position: absolute;
```

```
  left: 2px;
```

```
  top: 2px;
```

```
z-index: -2;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>Learning Overlapping Property</p>
```

```

```

```
<p>As the image keeps a z-index of -2, you will be able to place it behind the  
text of the webpage.</p>
```

```
</body>
```

```
</html>
```

## ***Layout Overflow***

The CSS layout overflow property is used to control the content that appears too big for a particular area. The overflow property specifies if a programmer wants to pin the content or add scrollbars to it when the content of a certain element appears to be too for an area on your web page. The CSS overflow property uses several values to allow programmers freedom for experimentation. One important thing to remember is that the overflow property works for block elements with a specified height.

### **Visible**

The first overflow property is named as visible. By default, a webpage overflow stays visible. It means that it is never clipped, and it tends to render out of the box of an element.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  background-color: skyblue;
  width: 150px;
  height: 80px;
  border: 2px dotted red;
  overflow: visible;
}
</style>
</head>
<body>
```

```
<p>This is CSS Overflow</p>
```

```
<p>CSS, if paired with HTML 5, revolutionizes the look of your website</p>
```

```
<div>The Covid-19 pandemic spread really fast across the world. The initial scare of the disease was beyond words. However, gradually people learned that it was not as deadly as it appeared at first. </div>
```

```
</body>
```

```
</html>
```

---

This is CSS Overflow

CSS, if paired with HTML 5, revolutionizes the look of your website

The Covid-19 pandemic spread really fast across the world. The initial scare of the disease was beyond words. However, gradually people learned that it was not as deadly as it appeared at first.

Although the text here appears inside the body, it will be displayed out of the body on a web page.

## Hidden

The second property is known as hidden. With this value, your overflow will appear as clipped. Whatever is out of the box will be hidden.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {  
  background-color: skyblue;  
  width: 150px;  
  height: 80px;  
  border: 2px dotted red;  
  overflow: hidden;  
}  
</style>  
</head>  
<body>
```

```
<p>This is CSS Overflow</p>
```

```
<p>CSS, if paired with HTML 5, revolutionizes the look of your website</p>
```

```
<div>The Covid-19 pandemic spread really fast across the world. The initial  
scare of the disease was beyond words. However, gradually people learned  
that it was not as deadly as it appeared at first. </div>
```

```
</body>
```

```
</html>
```

---

This is CSS Overflow

CSS, if paired with HTML 5, revolutionizes the look of your website

The Covid-19 pandemic spread really fast across the world. The initial scare of the disease was beyond words. However, gradually

people learned that it was not

You can see that the text has been cut. The text that could not fit in the box has been slashed.

## Scroll

Another value is scroll. The overflow stays clipped, but a scrollbar pops up on the scroll in the box. Users can use the scrollbar vertically and horizontally to read the full text.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
  background-color: skyblue;
```

```
  width: 150px;
```

```
  height: 80px;
```

```
  border: 2px dotted red;
```

```
  overflow: scroll;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS Overflow</p>
```

```
<p>CSS, if paired with HTML 5, revolutionizes the look of your website</p>
```

```
<div>The Covid-19 pandemic spread really fast across the world. The initial  
scare of the disease was beyond words. However, gradually people learned  
that it was not as deadly as it appeared at first. </div>
```

```
</body>
```

```
</html>
```

The auto value is like the scroll value as it also adds scrollbars to the code.

## ***CSS Float***

The CSS float property adds a bit of style to your web page. It specifies how different HTML elements on a webpage should float. Coupled with the float property is the clear property that adjusts the floating element. You can position and format your content by using the float property to suspend an image on a webpage to the right or the left sides. Additional values for the float property are none, left, right, and inherit.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {
```

```
    float: right;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS Float</p>
```

```
<p          img          src="Covid.jpg"          alt="Covid"
style="width:180px;height:180px;margin-left:20px;">The          Covid-19
pandemic spread really fast across the world. The initial scare of the disease
was beyond words. However, gradually people learned that it was not as
deadly as it appeared at first. </p>
```

```
</body>
```

```
</html>
```

---

## This is CSS Float

The Covid-19 pandemic spread really fast across the world. The initial scare of the disease was beyond words. However, gradually people learned that it was not as deadly as it appeared at first.

Upload a real image to the code and run it on the browser to see how the image floats to the code's right side. You can apply the left value in the same manner. The text will shift to the right side, and the image will come to the left side. You can create parallel floating items on the webpage. Each item will float against the other. See the following example.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div {
```

```
    float: left;
```

```
    padding: 20px;
```

```
}
```

```
.kaz1 {
```

```
    background: skyblue;
```

```
}
```

```
.kaz2 {
```

```
    background: yellow;
```

```
}
```

```
.kaz3 {  
  background: skyblue;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is CSS Float</p>
```

```
<div class="kaz1">The Covid-19 pandemic spread really fast across the  
world.</div>
```

```
<div class="kaz2">The initial scare of the disease was beyond words.</div>
```

```
<div class="kaz3">However, gradually people learned that it was not as  
deadly. </div>
```

```
</body>
```

```
</html>
```

---

This is CSS Float

The Covid-19 pandemic spread really fast across the world.

The initial scare of the disease was beyond words.

However, gradually people learned that it was not as deadly.

## Clear Property

The clear property decides upon the fate of the floating elements. It specifies

which elements will float parallel to the cleared HTML elements. It also decides on which they will float. You can add up to the property different values like none, left, inherit, right, and both. One of the most common ways to use this property is after using the float property on a certain element. When you are clearing floats, you must match the clear property to the float property. If HTML elements float to the web page's left side, you must clear to the opposite side. The floated element will keep floating while the cleared element will pop up under the floated element.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.kaz1 {
```

```
  float: left;
```

```
  width: 150px;
```

```
  height: 40px;
```

```
  margin: 20px;
```

```
  border: 4px solid skyblue;
```

```
}
```

```
.kaz2 {
```

```
  border: 2px solid yellow;
```

```
}
```

```
.kaz3 {
```

```
  float: left;
```

```
  width: 150px;
```

```
height: 40px;
margin: 20px;
border: 4px solid skyblue;
}
```

```
.kaz4 {
border: 2px solid yellow;
clear: left;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>No clear</p>
```

```
<div class="kaz1">Paragraph 1</div>
```

```
<div class="kaz2">Paragraph 2- Please note that the paragraph 2 comes after paragraph 1 in HTML code. However, since we have a float property for paragraph 1, it will float to the left side of the page, the text inside paragraph 2 will flow around paragrpah1.</div>
```

```
<br><br>
```

```
<p>Yes clear</p>
```

```
<div class="kaz3">Paragraph 3</div>
```

```
<div class="kaz4">Paragraph 4 will drop down the floating paragraph 3. The value "left" will clear the HTML elements that have been floated to the left. You can alter the code by using clear "right" or "both" to experiment how they affect the layout of your web page.</div>
```

```
</body>
```

```
</html>
```

---

No clear

Paragraph 1

Paragraph 2- Please note that paragraph 2 comes after paragraph 1 in HTML code. However, since we have a float property for paragraph 1, it will float to the left side of the page, the text inside paragraph 2 will flow around paragraph1.

Yes clear

Paragraph 3

Paragraph 4 will drop down the floating paragraph 3. The value "left" will clear the HTML elements that have been floated to the left. You can alter the code by using clear "right" or "both" to experiment with how they affect your web page's layout.

## ***CSS Float Examples***

With the help of the CSS float property, you can float boxes side by side to one another.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.boxing {
  float: left;
  width: 30%;
  padding: 40px;
}

.clearfixingfeature::after {
  content: "";
  clear: both;
  display: table;
}
</style>
</head>
<body>
```

```
<p>This is a Grid of Boxes</p>
```

```
<p>You are looking at Float boxes put side by side:</p>
```

```
<div class="clearfixingfeature">
```

```
  <div class="boxing" style="background-color:skyblue">
```

```
    <p>The first suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:red">
```

```
    <p>The second suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:yellow">
```

```
    <p>The third suspended box.</p>
```

```
  </div>
```

```
</div>
```

```
<p>The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding. </p>
```

```
</body>
```

```
</html>
```

---

This is a Grid of Boxes

You are looking at Float boxes put side by side:

The first suspended box.

The second suspended box.

## The third suspended box.

The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding.

Now I will remove the boxsize feature to see how it affects the layout of the webpage. See the example.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.boxing {
```

```
  float: left;
```

```
  width: 30%;
```

```
  padding: 40px;
```

```
}
```

```
.clearfixingfeature::after {
```

```
  content: "";
```

```
  clear: both;
```

```
  display: table;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a Grid of Boxes</p>
```

```
<p>You are looking at Float boxes put side by side:</p>
```

```
<div class="clearfixingfeature">
```

```
  <div class="boxing" style="background-color:skyblue">
```

```
    <p>The first suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:red">
```

```
    <p>The second suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:yellow">
```

```
    <p>The third suspended box.</p>
```

```
  </div>
```

```
</div>
```

<p>The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding. </p>

```
</body>
```

```
</html>
```

---

If you load it up to a webpage, the boxes will appear suspended but they will not be of perfect size.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
* {
  box-sizing: border-box;
}
```

```
.image {
  float: left;
  width: 30%;
  padding: 4px;
}
```

```
.clearfixingfeature::after {
  content: "";
  clear: both;
  display: table;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a Grid of Boxes</p>
```

```
<p>You are looking at Float boxes put side by side:</p>
```

```
<div class="clearfixingfeature">
```

```
  <div class="image">
```

```

</div>
<div class="image">

</div>
<div class="image">

</div>
</div>
<p>The clearfix hack ameliorates the layout flow. The box-sizing property
ensures that your boxes do not break after you add extra padding. </p>

</body>
</html>
```

---

This is a Grid of Boxes

You are looking at Float boxes put side by side:

The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding.

When you create boxes, you can fill them up and make them float by this code. However, if you add something that slips out of the box's width, the box will break. The box-sizing property allows you to adjust to the changing width and height and keep it from breaking.

## Equal Height Boxes

You have already learned how you can float the boxes while keeping the width the same. However, the real challenge is to create boxes and make them float with equal height. You can set up the fixed height to resolve the problem.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

.boxing {
  float: left;
  width: 30%;
  padding: 40px;
  height: 200px;
}

.clearfixingfeature::after {
  content: "";
  clear: both;
  display: table;
}
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a Grid of Boxes</p>
```

```
<p>You are looking at Float boxes put side by side:</p>
```

```
<div class="clearfixingfeature">
```

```
  <div class="boxing" style="background-color:skyblue">
```

```
    <p>The first suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:red">
```

```
    <p>The second suspended box.</p>
```

```
  </div>
```

```
  <div class="boxing" style="background-color:yellow">
```

```
    <p>The third suspended box.</p>
```

```
  </div>
```

```
</div>
```

```
<p>The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding. </p>
```

```
</body>
```

```
</html>
```

---

This is a Grid of Boxes

You are looking at Float boxes put side by side:

The first suspended box.

The second suspended box.

The third suspended box.

The clearfix hack ameliorates the layout flow. The box-sizing property ensures that your boxes do not break after you add extra padding.

This formula works, but it is the least flexible. This will keep working for a long time if you believe that the boxes will have the same amount of content. However, the content does not stay the same. It tends to change over time. If a user opens your website on a mobile, the content is likely to slip out of the boxes. You need something flexible that can adjust the boxes' content to web browsers and mobile browsers in the same way. To make it flexible and responsive, I will use the Flexbox property of CSS in the following code.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flexbox1 {
  display: flex;
  flex-wrap: nowrap;
  background-color: black;
}

.flexbox1 .box1 {
  background-color: black;
  width: 40%;
  margin: 15px;
```

```
text-align: left;
line-height: 50px;
font-size: 20px;
}
</style>
</head>
<body>

<p>This is a Grid of Boxes</p>
<p>You are looking at Float boxes put side by side:</p>

<div class="flexbox1">
  <div class="box1" style="background-color:skyblue">
    <p>The first suspended box.</p>
  </div>
  <div class="box1" style="background-color:red">
    <p>The second suspended box.</p>
  </div>
  <div class="box1" style="background-color:yellow">
    <p>The third suspended box.</p>
  </div>
</div>

</body>
</html>
```

---

This is a Grid of Boxes

You are looking at Float boxes put side by side:

The first suspended box.

The second suspended box.

The third suspended box.

## ***Designing Menu***

You can use the float property to design a horizontal menu in CSS.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: ;
  padding: 0;
  overflow: hidden;
  background-color: skyblue;
}

li {
  float: left;
}

li a {
  display: inline-block;
  color: black;
  text-align: left;
  padding: 12px 14px;
  text-decoration: none;
}
```

```
li a:hover {  
  background-color: yellow;  
}
```

```
.active {  
  background-color: red;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#homep" class="active">My Home Page</a></li>
```

```
<li><a href="#newsc">News Corner</a></li>
```

```
<li><a href="#contactn">Contact Number & Email</a></li>
```

```
<li><a href="#aboutu">About Us</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

- 
- My Home Page
  - News Corner
  - Contact Number & Email

- About Us

In the following example, I will use the float property to design a website's complete layout. You can use the code, edit, design your web page.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
* {
```

```
  box-sizing: border-box;
```

```
}
```

```
.headers, .footers {
```

```
  background-color: lightgrey;
```

```
  color: black;
```

```
  padding: 20px;
```

```
}
```

```
.columns {
```

```
  float: left;
```

```
  padding: 20px;
```

```
}
```

```
.clearfixing::after {
```

```
  content: "";
```

```
  clear: both;
```

```
  display: table;
```

```
}
```

```
.menus {  
  width: 30%;  
}
```

```
.contents {  
  width: 85%;  
}
```

```
.menus ul {  
  list-style-type: none;  
  margin: 0;  
  padding: 0;  
}
```

```
.menus li {  
  padding: 10px;  
  margin-bottom: 10px;  
  background-color: skyblue;  
  color: white;  
}
```

```
.menus li:hover {  
  background-color: yellow;
```

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

<div class="headers">
  <p><b>Designing a web page</b></p>
</div>
```

```
<div class="clearfixing">
  <div class="columns menus">
    <ul>
      <li>Sky</li>
      <li>Pine trees</li>
      <li>Beaches</li>
      <li>Organic food</li>
    </ul>
  </div>
```

```
<div class="columns contents">
  <p><b>Mysterious Places</b></p>
  <p>The world is full of mysterious place. You can find giant spiders, lofty
  trees, ground level clouds, miles long fish, and coconuts, the size of an air
  balloon. </p>
  <p>Web designing gets very excited when you have mastered the art of
  coding in CSS.</p>
```

```
</div>
```

```
</div>
```

```
<div class="footers">
```

```
<p>Footer Text</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## Designing a web page

Sky

Pine trees

Beaches

Organic food

## Mysterious Places

The world is full of mysterious places. You can find giant spiders, lofty trees, ground-level clouds, miles-long fish, and coconuts, the size of an air balloon.

Web designing gets very excited when you have mastered the art of coding in CSS.

Footer Text

## **Chapter Seven: CSS Display Features**

There are many features to set up the display of your website. You have already learned how to set float to design a web page. One of the most common display features is the inline-block feature. This allows to set up the height and width of the element. With the help of the inline-block feature, your paddings and margins will be respected by browsers.

## ***Inline-Block***

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
span.z {
```

```
    display: inline; /* this is the default feature for span */
```

```
    width: 110px;
```

```
    height: 110px;
```

```
    padding: 6px;
```

```
    border: 2px solid skyblue;
```

```
    background-color: grey;
```

```
}
```

```
span.y {
```

```
    display: inline-block;
```

```
    width: 110px;
```

```
    height: 110px;
```

```
    padding: 6px;
```

```
    border: 2px solid blue;
```

```
    background-color: yellow;
```

```
}
```

```
span.x {
```

```
    display: block;
```

```
width: 110px;
height: 110px;
padding: 6px;
border: 2px solid skyblue;
background-color: yellow;
}
</style>
</head>
<body>

<p>Learning display Property</p>
```

```
<p>This is display: inline</p>
```

```
<div> The world's tallest building sits in Dubai. However, Burj Khalifa is
going to have a rival very soon.
```

```
    <span class="z">United Arab Emirates</span> <span class="z">Burj
Khalifa</span> No matter what, the grandeur of the building is matchless.
</div>
```

```
<p>This is display: inline-block</p>
```

```
<div>The world's tallest building sits in Dubai. However, Burj Khalifa is
going to have a rival very soon.
```

```
    <span class="y">UAE</span> <span class="y">Dubai</span>
Sometimes, it clouds shower rains standing at half of the building. </div>
```

```
<p>This is display: block</p>
```

```
<div>The world's tallest building sits in Dubai. However, Burj Khalifa is
```

going to have a rival very soon.

```
<span class="x">Deserts</span> <span class="x">Skyscrapers</span> It  
also sways back and forth when it gets windy and stormy. </div>
```

```
</body>
```

```
</html>
```

---

## *Learning Display Property*

This is display: inline

The world's tallest building sits in Dubai. However, Burj Khalifa is going to have a rival very soon. United Arab Emirates Burj Khalifa No matter what, the grandeur of the building is matchless.

This is display: inline-block

The world's tallest building sits in Dubai. However, Burj Khalifa is going to have a rival very soon. UAE Dubai Sometimes, it clouds shower rains standing at half of the building.

This is display: block

The world's tallest building sits in Dubai. However, Burj Khalifa is going to have a rival very soon. DesertsSkyscrapers It also sways back and forth with it gets windy and stormy.

## ***Navigation Links***

You can use the inline-block to add more interaction on the web page. Creating navigation links is the most common practice in website designing. It is good for providing ease of use to visitors. You can display items in a horizontal format instead of a vertical format. The following example will create a navigation link.

```
<!DOCTYPE html>
<html>
<head>
<style>
.navigation {
  background-color: yellow;
  list-style-type: none;
  text-align: center;
  margin: 0;
  padding: 0;
}

.navigation li {
  display: inline-block;
  font-size: 20px;
  padding: 20px;
}

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

```
<p>Learning display Property</p>
```

```
<p>This is display: inline</p>
```

```
<p> The world's tallest building sits in Dubai. However, Burj Khalifa is going to have a rival very soon. No matter what, the grandeur of the building is matchless. </p>
```

```
<ul class="navigation">
```

```
  <li><a href="#home">Home Page</a></li>
```

```
  <li><a href="#about">About Us</a></li>
```

```
  <li><a href="#clients">About My Clients</a></li>
```

```
  <li><a href="#contact">Contact Email</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

---

## *Learning Display Property*

This is display: inline

The world's tallest building sits in Dubai. However, Burj Khalifa is going to have a rival very soon. No matter what, the grandeur of the building is matchless.

- [Home Page](#)

- [About Us](#)

- [About My Clients](#)

- [Contact Email](#)

Now users can click on the desired button and move to the other pages of your website. You can add more pages as you desire.

## ***Alignment***

The display of a website is connected to how beautifully the pages are aligned. You can set the alignment as center. If you also set up the width of HTML elements, you can keep them from stretching out toward the edges. The element will adjust to the specified width, and the space that will be left out will split up between margins.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.centersaligned {
```

```
  margin: auto;
```

```
  width: 70%;
```

```
  border: 2px solid red;
```

```
  padding: 15px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>These are Center Align Elements on a web page</h4>
```

```
<p>I am using the margin: auto property to set up the alignment on the web page;</p>
```

```
<div class="centersaligned">
```

```
  <p>This is CSS and you are creating a web page!</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## ***These are Center Align Elements on a Web Page***

I am using the margin: auto property to set up the alignment on the web page;

This is CSS, and you are creating a web page!

### **Center-Aligning Text**

If you want to center align the text in the box, you can use the text-align property.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.centersaligned {
```

```
  margin: auto;
```

```
  width: 70%;
```

```
  border: 2px solid red;
```

```
  padding: 15px;
```

```
  text-align: center;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>These are Center Align Elements on a web page</h4>
```

```
<p>I am using the margin: auto property to set up the alignment on the web page;</p>
```

```
<div class="centersaligned">
```

```
<p>This is CSS and you are creating a web page!</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

### ***These are Center Align Elements on a Web Page***

I am using the margin: auto property to set up the alignment on the web page;

This is CSS, and you are creating a web page!

### **Image Aligning**

You can add an image and align it on the page as per your will. See the following code.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
img {  
  display: block;  
  margin-left: auto;  
  margin-right: auto;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>These are Center Align Elements on a web page</h4>
```

```
<p>I am using the margin: auto property to set up the alignment on the web  
page;</p>
```

```
  <p>This is CSS and you are creating a web page!</p>
```

```

```

```
</body>
```

```
</html>
```

---

### ***These are Center Align Elements on a Web Page***

I am using the margin: auto property to set up the alignment on the web page;

This is CSS, and you are creating a web page!

## ***Left & Right Aligning***

You can use CSS to set up right or left alignment to place HTML elements at the right locations.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.leftalignment {
```

```
  position: absolute;
```

```
  left: 0px;
```

```
  width: 250px;
```

```
  border: 2px solid black;
```

```
  padding: 15px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Aligned Elements on the web page</h4>
```

```
<div class="leftalignment">
```

```
<p>I am using position: absolute property to set up the alignment on the web page.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

### ***Aligned Elements on the Web Page***

I am using position: absolute property to set up the alignment on the web page.

## *Float for Alignment*

You can use another method, namely, float, to set up the alignment to the right or left. The float property is smoother and makes snug alignment. Let us have a demonstration of the float property.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.rightalign {
```

```
  float: left;
```

```
  width: 250px;
```

```
  border: 2px solid black;
```

```
  padding: 10px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Aligned Elements on the web page</h4>
```

```
<div class="rightalignment">
```

```
<p>I am using float property to set up alignment on the web page.</p>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

### ***Aligned Elements on the Web Page***

I am using float property to set up alignment on the web page.

## **Chapter Eight: CSS Combinators**

A CSS combinatory explains the relationship between different selectors - a CSS selector may contain more than a simple selector. Selectors are of different types. This chapter will explain how these selectors work and how combinators form and analyze the relationship between them.

## ***Descendant Selector***

The descendant selector tends to match the elements that descend from an element. The next example will help you select the <p> and <div> elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div p {
```

```
  background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>This is Descendant Selector</h4>
```

```
<p>This descendant selector will match the elements that descend from a particular element.</p>
```

```
<div>
```

```
  <p>This is the first line.</p>
```

```
  <p>This is the second line.</p>
```

```
  <section><p>This is the third and last line of the div.</p></section>
```

```
</div>
```

```
<p>This line is out of the div.</p>
```

```
<p>This line is out of the div.</p>
```

```
</body>
```

```
</html>
```

---

### ***This is Descendant Selector***

This descendant selector will match the elements that descend from a particular element.

This is the first line.

This is the second line.

This is the third and last line of the div.

This line is out of the div.

This line is out of the div.

## ***Child Selector***

The second type is the child selector. It will select elements that have one specified parent element. In the following example, I will select the child elements of a parent selector.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div > p {
```

```
  background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>This is Child Selector</h4>
```

```
<p>This child selector (>) should link to all the that belong to a specified element.</p>
```

```
<div>
```

```
  <p>This line falls inside the div.</p>
```

```
  <p>This line falls inside the div.</p>
```

```
  <section><p>This line falls inside the div.</p></section>
```

```
  <p>This line also falls inside the div.</p>
```

```
</div>
```

<p>This line falls outside of the div.</p>

<p>This line falls outside of the div.</p>

</body>

</html>

---

### ***This is Child Selector***

This child selector (>) should link to all that belong to a specified element.

This line falls inside the div.

This line falls inside the div.

This line falls inside the div.

This line also falls inside the div.

This line falls outside of the div.

This line falls outside of the div.

## ***Adjacent Sibling Selector***

The third type is known as adjacent sibling selector. It is used for the selection of the elements that are after a specific element. Sibling elements possess a single parent element. I will sort out all the adjacent elements in the following example.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div + p {
```

```
    background-color: skyblue;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>This is Adjacent Sibling Selector</h4>
```

```
<p>This adjacent sibling selector (+) selects the elements that are after a  
specific element.</p>
```

```
<div>
```

```
    <p>This line falls inside the div.</p>
```

```
    <p>This line falls inside the div.</p>
```

```
</div>
```

```
<p>This line falls inside the div.</p>
```

```
<p>This line also falls inside the div.</p>
```

```
<div>
<p>This line falls outside of the div.</p>
<p>This line falls outside of the div.</p>
</div>
```

```
<p>This line falls outside of the div.</p>
<p>This line falls outside of the div.</p>
```

```
</body>
</html>
```

---

### ***This is Adjacent Sibling Selector***

This adjacent sibling selector (+) selects the elements that are after a specific element.

This line falls inside the div.

This line falls inside the div.

This line falls inside the div.

This line also falls inside the div.

This line falls outside of the div.

This line falls outside of the div.

This line falls outside of the div.

This line falls outside of the div.

## ***General Sibling Selector***

The last type is of general sibling selector. It will select the elements that appear to be siblings of a particular element in the code. The ~ operator is used for this purpose.

```
<!DOCTYPE html>
<html>
<head>
<style>
div ~ p {
  background-color: skyblue;
}
</style>
</head>
<body>

<h4>This is General Sibling Selector</h4>

<div>
  <p>This line falls inside the div.</p>
</div>

  <p>This line falls inside the div.</p>

<div>
<p>This line falls outside of the div.</p>
</div>
```

<p>This line falls outside of the div.</p>

</body>

</html>

---

### ***This is General Sibling Selector***

This line falls inside the div.

This line falls inside the div.

This line falls outside of the div.

This line falls outside of the div.

## Chapter Nine: CSS Bars, Dropdowns & Galleries

When designing a website, you need navigation bars to allow users to move around the website and perform different tasks. If you have an easy-to-use navigation system for the website, your visitors will love your website. With the help of CSS, boring HTML menus appear to be quite good-looking navigation bars. A navigation bar is made of standard HTML elements. I will build a navigation bar in the following example.

```
<!DOCTYPE html>
<html>
<body>

<ul>
  <li><a href="#homepage">Home Page</a></li>
  <li><a href="#newspage">News Corner</a></li>
  <li><a href="#contactpage">Contact Number</a></li>
  <li><a href="#aboutpage">About Us</a></li>
</ul>

<p>I am using href="#" just for testing links. When you are building a real
website, you should use URLs.</p>

</body>
</html>
```

- 
- [Home Page](#)
  - [News Corner](#)

- [Contact Number](#)
- [About Us](#)

I am using href="#" just for testing links. When you are building a real website, you should use URLs.

I am sure this does not look attractive to you. You might be thinking about how it will attract users. Of course, it will not. The good thing is that you can change that by tuning the same code and by adding margins and padding. I will explain that step by step. See the following example.

```
<!DOCTYPE html>
<html>
<body>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}
</style>

<ul>
  <li><a href="#homepage">Home Page</a></li>
  <li><a href="#newspage">News Corner</a></li>
  <li><a href="#contactpage">Contact Number</a></li>
  <li><a href="#aboutpage">About Us</a></li>
</ul>
```

<p>I am using href="#" just for testing links. When you are building a real website, you should use URLs.</p>

</body>

</html>

- 
- [Home Page](#)
  - [News Corner](#)
  - [Contact Number](#)
  - [About Us](#)

I am using href="#" just for testing links. When you are building a real website, you should use URLs.

The list-style-type feature removes the bullets from the navigation bar. This code is the standard for creating navigation bars in horizontal and vertical shapes.

## Vertical Bars

When you want to build a vertical navigation bar, you can use <a> element in the list.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<style>
```

```
ul {
```

```
list-style-type: none;
```

```
margin: 0;
```

```
padding: 0;
}
```

```
li a {
display: block;
width: 80px;
background-color: skyblue;
}
```

```
</style>
```

```
<ul>
```

```
<li><a href="#homepage">Home Page</a></li>
```

```
<li><a href="#newspage">News Corner</a></li>
```

```
<li><a href="#contactpage">Contact Number</a></li>
```

```
<li><a href="#aboutpage">About Us</a></li>
```

```
</ul>
```

<p>I am using href="#" just for testing links. When you are building a real website, you should use URLs.</p>

```
</body>
```

```
</html>
```

---

[Home Page](#)

[News Corner](#)

[Contact Number](#)

[About Us](#)

I am using href="#" just for testing links. When you are building a real website, you should use URLs.

When you display the links as elements of the block, it will make the links area fully clickable and not only the text. It will allow you to specify the width, margin, height, and padding as well. The block elements will take up the complete width that is available to you by default. The width can always be removed. As a result, the elements will cover the available width when you display them in the form of block elements. This should produce the same result as the previous example does.

```
<!DOCTYPE html>
<html>
<body>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

li a {
  display: block;
  background-color: skyblue;
}

</style>
```

```
<ul>
  <li><a href="#homepage">Home Page</a></li>
  <li><a href="#newspage">News Corner</a></li>
  <li><a href="#contactpage">Contact Number</a></li>
  <li><a href="#aboutpage">About Us</a></li>
</ul>
```

<p>I am using href="#" just for testing links. When you are building a real website, you should use URLs.</p>

```
</body>
</html>
```

You can get the output in the browser. The elements of the bar will cover the entire width available.

An amazing feature in the vertical navigation bars is the change of color when a user hovers the mouse over an element. In the following example, I will create a navigation bar that changes colors when you move the mouse over it. I have added some colors. The background is white, and the elements of the navigation bar are of sky blue color. When you hover the mouse over the bar, it will turn red. You can test different color combinations to produce the result of your choice.

```
<!DOCTYPE html>
<html>
<body>
<style>
ul {
  list-style-type: none;
```

```
margin: 0;
padding: 0;
width: 250px;
background-color: grey;
}
```

```
li a {
display: block;
color: skyblue;
padding: 6px 14px;
text-decoration: none;
}
```

```
/* The following lines of code will change the color of the link on mouse
hover */
```

```
li a:hover {
background-color: red;
color: white;
}
```

```
</style>
```

```
<ul>
```

```
<li><a href="#homepage">Home Page</a></li>
```

```
<li><a href="#newspage">News Corner</a></li>
```

```
<li><a href="#contactpage">Contact Number</a></li>
<li><a href="#aboutpage">About Us</a></li>
</ul>
```

```
<p>I am using href="#" just for testing links. When you are building a real
website, you should use URLs.</p>
```

```
</body>
</html>
```

---

[Home Page](#)

[News Corner](#)

[Contact Number](#)

[About Us](#)

I am using href="#" just for testing links. When you are building a real website, you should use URLs.

## ***Active Class***

You can add an active class to the code to let the user know which page he or she is on.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<style>
```

```
ul {
```

```
list-style-type: none;
```

```
margin: 0;
```

```
padding: 0;
```

```
width: 250px;
```

```
background-color: grey;
```

```
}
```

```
li a {
```

```
display: block;
```

```
color: skyblue;
```

```
padding: 6px 14px;
```

```
text-decoration: none;
```

```
}
```

```
li a.activeclass {
```

```
background-color: lightgrey;
```

```
color: white;
```

```
}
```

```
li a:hover:not(.activeclass) {  
  background-color: yellow;  
  color: white;  
}
```

```
</style>
```

```
<ul>
```

```
<li><a class="activeclass" href="#homepage">Home Page</a></li>
```

```
<li><a href="#newspage">News Corner</a></li>
```

```
<li><a href="#contactpage">Contact Number</a></li>
```

```
<li><a href="#aboutpage">About Us</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

- 
- [Home Page](#)
  - [News Corner](#)
  - [Contact Number](#)
  - [About Us](#)

## ***Adding Borders***

You can further decorate your navigation bars by adding borders and central links. The border property is added to <ul> around the navigation bar. You also can place it inside the bar.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<style>
```

```
ul {
```

```
  list-style-type: none;
```

```
  margin: 0;
```

```
  padding: 0;
```

```
  width: 250px;
```

```
  background-color: #f1f1f1;
```

```
  border: 2px solid black;
```

```
}
```

```
li a {
```

```
  display: block;
```

```
  color: seagreen;
```

```
  padding: 6px 14px;
```

```
  text-decoration: none;
```

```
}
```

```
li {
```

```
text-align: center;
border-bottom: 2px solid seagreen;
}
```

```
li:last-child {
border-bottom: none;
}
```

```
li a.activeclass {
background-color: lightgrey;
color: white;
}
```

```
li a:hover:not(.activeclass) {
background-color: yellow;
color: white;
}
```

```
</style>
```

```
<ul>
```

```
<li><a class="activeclass" href="#homepage">Home Page</a></li>
```

```
<li><a href="#newspage">News Corner</a></li>
```

```
<li><a href="#contactpage">Contact Number</a></li>
```

```
<li><a href="#aboutpage">About Us</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

- 
- [Home Page](#)
  - [News Corner](#)
  - [Contact Number](#)
  - [About Us](#)

## ***Sticky Navbar***

To add more flavor to your website, you can create a sticky navigation bar. The navigation bar will be of full length, and it will be sticky as well. Whenever the user scrolls down or up, the bar will stick to the side of the page. Beware that you should use it after doing thorough customer research because some users may find it annoying. A test before you go for it is necessary. No matter whether users like it or not, you should know how to create that.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
  margin: 0;
```

```
}
```

```
ul {
```

```
  list-style-type: none;
```

```
  margin: 0;
```

```
  padding: 0;
```

```
  width: 20%;
```

```
  background-color: #f1f1f1;
```

```
  position: fixed;
```

```
  height: 110%;
```

```
  overflow: auto;
```

```
}
```

```
li a {
  display: block;
  color: grey;
  padding: 6px 14px;
  text-decoration: none;
}
```

```
li a.activeclass {
  background-color: skyblue;
  color: white;
}
```

```
li a:hover:not(.activeclass) {
  background-color: lightgrey;
  color: white;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a class="activeclass" href="#home">Home Page</a></li>
```

```
<li><a href="#news">News Corner</a></li>
```

```
<li><a href="#contact">Contact No</a></li>
```

```
<li><a href="#about">About Us</a></li>
```

</ul>

<div style="margin-left:20%;padding:2px 14px;height:900px;">

<p>This is Full-height Side Nav Bar</p>

<P>When you scroll down this area you will see that the side nav bar will stick to the web page</h3>

<p>Put Some text here so that you can scroll down the web page. Insufficient text will keep you from scrolling down and you will never test the sticky bar.</p>

<p>Put Some text here..</p>

<p>Put Some text here..</p>

<p>Put Some text here..</p>

</div>

</body>

</html>

## ***Horizontal NavBar***

Another type of navigation bar is the horizontal bar. You can either create a floating bar or an inline to list different items on the web page.

### **Inline**

In this type of coding, you have to specify the elements of <li> as inline. This will be an extra layer of coding to the standard code on the web page.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
    list-style-type: none;
```

```
    margin: 1px;
```

```
    padding: 1px;
```

```
}
```

```
li {
```

```
    display: inline;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#home">Home Page</a></li>
<li><a href="#news">News Corner</a></li>
<li><a href="#contact">Contact No</a></li>
<li><a href="#about">About Us</a></li>
</ul>
```

```
</body>
</html>
```

---

[Home Page](#) [News Corner](#) [Contact No](#) [About Us](#)

The <li> elements are by default block elements. You may line breaks before and after each item of the list to display the same on a single line on the web page.

## Floats

If you want to create floating list items, you will have to specify first the layout for all the navigation links on the web page.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 1px;
  padding: 1px;
```

```
overflow: hidden;  
}
```

```
li {  
  float: left;  
}
```

```
li a {  
  display: block;  
  padding: 10px;  
  background-color: skyblue;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
  <li><a href="#home">Home Page</a></li>
```

```
  <li><a href="#news">News Corner</a></li>
```

```
  <li><a href="#contact">Contact No</a></li>
```

```
  <li><a href="#about">About Us</a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

---

[Home Page](#) [News Corner](#) [Contact No](#) [About Us](#)

The float: left property will use set the block elements next to each other on the page. The display: block will allow you to pinpoint the padding like margins, heights, and widths if you need. Since the block elements consume the total available width, they are unable to float side by side. Some sort of padding adds beauty to the bar.

## Glamorizing Horizontal Bars

You can change the code to make your horizontal navigation look fanciful. In this example, I will explain how you can create a horizontal navigation bar with some background color and how the color will change as you hover the mouse pointer over the navigation links.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
list-style-type: none;
```

```
margin: 1px;
```

```
padding: 1px;
```

```
overflow: hidden;
```

```
background-color: lightgrey;
```

```
}
```

```
li {  
  float: left;  
}
```

```
li a {  
  display: block;  
  color: green;  
  text-align: center;  
  padding: 12px 14px;  
  text-decoration: none;  
}
```

```
li a:hover {  
  background-color: yellow;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a href="#home">Home Page</a></li>
```

```
<li><a href="#news">News Corner</a></li>
```

```
<li><a href="#contact">Contact No</a></li>
```

```
<li><a href="#about">About Us</a></li>
</ul>
```

```
</body>
</html>
```

---

[Home Page](#) [News Corner](#) [Contact No](#) [About Us](#)

## Active Links

We can add more code to the bar code by including active class, telling users which page they are browsing. I will set a color that the active link will carry.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 1px;
  padding: 1px;
  overflow: hidden;
  background-color: lightgrey;
}

li {
  float: left;
```

```
}
```

```
li a {
```

```
  display: block;
```

```
  color: green;
```

```
  text-align: center;
```

```
  padding: 12px 14px;
```

```
  text-decoration: none;
```

```
}
```

```
li a:hover {
```

```
  background-color: yellow;
```

```
}
```

```
.activeclass {
```

```
  background-color: skyblue;
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a class= "activeclass" href="#home">Home Page</a></li>
```

```
<li><a href="#news">News Corner</a></li>
```

```
<li><a href="#contact">Contact No</a></li>
```

```
<li><a href="#about">About Us</a></li>
</ul>
```

```
</body>
</html>
```

---

[Home Page](#) [News Corner](#) [Contact No](#) [About Us](#)

The active link will have sky blue color, and you can change the color of the active link later on. Active links make your code more interactive and fanciful.

## Right Alignment

You can create a code that will float some of the navigation bar elements to the right side of the page. Sometimes you need some of the elements like ‘contact no’ or ‘about us’ to be separate from the other elements. This is where the right alignment feature becomes of great help.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
list-style-type: none;
margin: 1px;
padding: 1px;
overflow: hidden;
background-color: lightgrey;
```

```
}
```

```
li {  
  float: left;  
}
```

```
li a {  
  display: block;  
  color: green;  
  text-align: center;  
  padding: 12px 14px;  
  text-decoration: none;  
}
```

```
li a:hover {  
  background-color: yellow;  
}
```

```
.activeclass {  
  background-color: skyblue;
```

```
</style>
```

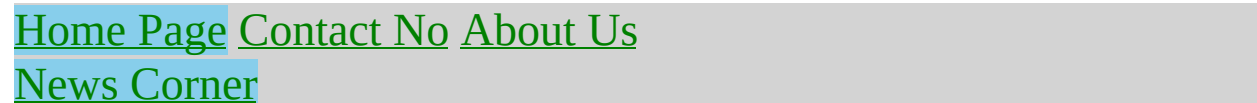
```
</head>
```

```
<body>
```

```
<ul>
  <li><a class= "activeclass" href="#home">Home Page</a></li>
  <li style="float:right"><a class="activeclass" href="#news">News
  Corner</a></li>
  <li><a href="#contact">Contact No</a></li>
  <li><a href="#about">About Us</a></li>
</ul>

</body>
</html>
```

---



[Home Page](#) [Contact No](#) [About Us](#)  
[News Corner](#)

## Fixed Navigation Bar

The following example carries the code for creating a horizontal navigation bar that sticks at the bottom or the top of a web page. Even if the user scrolls down the web page, the bar will stick to its place. When you are creating a sticky navigation bar, you should make sure that you keep the bar's width to the minimum so that it does not cover half of the page. Thick sticky horizontal bars are annoying because they sometimes obstruct the complete view of the page. Users find it hard to read the content or move around other elements on the web page. Therefore, it is better to get it tested.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
body {margin:1px;}
```

```
ul {  
  list-style-type: none;  
  margin: 1px;  
  padding: 1px;  
  overflow: hidden;  
  background-color: lightgrey;  
  position: fixed;  
  top: 1px;  
  width: 150%;  
}
```

```
li {  
  float: center;  
}
```

```
li a {  
  display: block;  
  color: white;  
  text-align: center;  
  padding: 12px 14px;  
  text-decoration: none;  
}
```

```
li a:hover:not(.activeclass) {  
  background-color: skyblue;  
}
```

```
.activeclass {  
  background-color: #3BAC60;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul>
```

```
<li><a class="activeclass" href="#home">Home Page</a></li>
```

```
<li><a href="#news">News Corner</a></li>
```

```
<li><a href="#contact">Contact No</a></li>
```

```
<li><a href="#about">About Us</a></li>
```

```
</ul>
```

```
<div style="padding:15px;margin-top:20px;background-color:yellow;height:1200px;">
```

```
<h4>This is the Fixed Top Navigation Bar</h4>
```

```
<h4>You will have to scroll down the web page to see how it works.</h4>
```

```
<h4>This navigation bar tends to stick to the top of the page as you scroll through the web page.</h4>
```

```
<p>Put some text to scroll.</p>
```

```
<p>Put some text to scroll.</p>
<p>Put some text to scroll.</p>
</div>

</body>
</html>
```

---

Home Page

News Corner

Contact No

About Us

***This is the Fixed Top Navigation Bar***

***You will have to scroll down the web page to see how it works.***

***This navigation bar tends to stick to the top of the page as you scroll through the web page.***

Put some text to scroll.

Put some text to scroll.

Put some text to scroll.

## Sticky Bar

The following code example will make the navigation bar stick to the top. If you fill in the web page with lots of text, you will see the sticky effect.

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
```

```
body {
```

```
  font-size: 31px;
```

```
}
```

```
ul {
```

```
  list-style-type: none;
```

```
  margin: 1px;
```

```
  padding: 1px;
```

```
  overflow: hidden;
```

```
  background-color: lightgrey;
```

```
  position: -webkit-sticky;
```

```
  position: sticky;
```

```
  top: 1px;
```

```
}
```

```
li {
```

```
  float: left;
```

```
}
```

```
li a {
```

```
  display: block;
```

```
  color: black;
```

```
  text-align: center;
```

```
  padding: 12px 14px;
```

```
text-decoration: none;
}
```

```
li a:hover {
  background-color: yellow;
}
```

```
.activeclass {
  background-color: green;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```
<h4>Sticky Horizontal Bar</h4>
```

```
<p>You must scroll down the web page to see the sticky effect.</p>
```

```
</div>
```

```
<ul>
```

```
<li><a class="activeclass" href="#home">Blog</a></li>
```

```
<li><a href="#news">News Page</a></li>
```

```
<li><a href="#contact">Poke Me</a></li>
```

```
</ul>
```

#### <h4>Sticky Navbar Explained</h4>

<p>This navbar tends to <strong>stick</strong> to the top as you reach the scroll position.</p>

<p><strong>Please note that</strong> Internet Explorer does not support this CSS feature. Also, Safari needs a -webkit- prefix.</p>

<p>Fill in some text to enable the sticky scroll.....Write more here.</p>

<p>Fill in some text to enable the sticky scroll.....Write more here.</p>

<p>Fill in some text to enable the sticky scroll.....Write more here.</p>

</body>

</html>

---

### *Sticky Horizontal Bar*

You must scroll down the web page to see the sticky effect.

[Blog](#) [News Page](#) [Poke Me](#)

### *Sticky Navbar Explained*

This navbar tends to **stick** to the top as you reach the scroll position.

**Please note that** Internet Explorer does not support this CSS feature.

Also, Safari needs a -webkit- prefix.

Fill in some text to enable the sticky scroll.....Write more here.

Fill in some text to enable the sticky scroll.....Write more here.

Fill in some text to enable the sticky scroll.....Write more here.

## **Chapter Ten: CSS Dropdowns & Images**

Dropdowns are amazing as they make your website interactive and easy to navigate. CSS allows you to create a brilliantly hoverable dropdown. You may create a dropdown box that will pop up when a user moves the mouse pointer over a specific element.

## ***A Basic Dropdown***

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.letusdropdown {  
  position: relative;  
  display: inline-block;  
}
```

```
.thisisdropdown-content {  
  display: none;  
  position: absolute;  
  background-color: lightgrey;  
  min-width: 180px;  
  box-shadow: 1px 7px 17px 1px rgba(0,0,0,0.2);  
  padding: 14px 18px;  
  z-index: 1;  
}
```

```
.letusdropdown:hover .thisisdropdown-content {  
  display: block;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>You are studying Hoverable Dropdown</h4>
```

```
<p>Users will move the mouse over text on the web page right below to pop open your dropdown content.</p>
```

```
<div class="letusdropdown">
```

```
<span>Please Hover Your Mouse Here</span>
```

```
<div class="thisisdropdown-content">
```

```
<p>CSS is amazing! Isn't it?</p>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

## ***You are Studying Hoverable Dropdown***

Users will move the mouse over text on the web page right below to pop open your dropdown content.

Please Hover Your Mouse Here

When a user hovers over the Please Hover Your Mouse Here, a drop-down menu will pop up. Users can click the link to reach another page or to initiate a task. You can also use the `<button>` element to create drop-down content. You can add anything inside the `<div>` element to create your dropdown content. I have used a class and relative position to set up the position of the menu. The second class, namely `thisisdropdown-content`, holds the content. The menu will stay hidden by default. You can only see it when you move your mouse pointer over it. You can change the width to experiment with the

design of the dropdown menu. You can use the border property instead of the box-shadow property if you are not comfortable with it.

## ***A Dropdown List***

The challenge is to add a list to the dropdown menu. You can add as many options in the dropdown list as you like. I will add navigation links to the dropdown menu to allow the user to switch between pages and do different tasks. You also can style the links and the menu to make them appear snazzy and stylish. Here is the code example which you can edit and customize for your website.

```
<!DOCTYPE html>
<html>
<head>
<style>
.mydropbutton {
  background-color: lightgrey;
  color: black;
  padding: 14px;
  font-size: 14px;
  border: none;
  cursor: pointer;
}

.letusdropdown {
  position: relative;
  display: inline-block;
}

.thisisdropdown-content {
  display: none;
```

```
position: absolute;
background-color: skyblue;
min-width: 180px;
box-shadow: 1px 7px 17px 1px rgba(0,0,0,0.2);
z-index: 1;
}

.thisisdropdown-content a {
color: black;
padding: 10px 14px;
text-decoration: none;
display: block;
}

.thisisdropdown-content a:hover {background-color: green}

.letusdropdown:hover .thisisdropdown-content {
display: block;
}

.letusdropdown:hover .mydropbutton {
background-color: red;
}

</style>
```

```
</head>
<body>

<h4>Hoverable Dropdown menus</h4>

<div class="letusdropdown">
  <button class="mydropbutton">My Dropdown Menu</button>
  <div class="thisisdropdown-content">
    <a href="#">Iceland</a>
    <a href="#">Ireland</a>
    <a href="#">India</a>
  </div>
</div>

</body>
</html>
```

---

## ***Hoverable Dropdown Menus***

### **My Dropdown Menu**

### **Dropdown Menu Navbar**

Let us take it to another level by creating a dropdown menu inside the navigation bar. You can add the dropdown menu to the bar if you need more interaction from the users. In fact, this combination of the navigation bar and dropdown menu is becoming standard in website designing. Users save time by hovering over the navigation bar and learn what the website contains. This

CSS feature speeds up decision-making at the end of users. This is how it boosts conversions. Users love the business that facilitates them in terms of saving their time.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 1px;
  padding: 1px;
  overflow: hidden;
  background-color: skyblue;
}

li {
  float: left;
}

li a, .mydropbutton {
  display: inline-block;
  color: grey;
  text-align: center;
  padding: 12px 14px;
  text-decoration: none;
}
```

```
li a:hover, .letusdropdown:hover .mydropbutton {  
  background-color: red;  
}
```

```
li.letusdropdown {  
  display: inline-block;  
}
```

```
.thisisdropdown-content {  
  display: none;  
  position: absolute;  
  background-color: skyblue;  
  min-width: 180px;  
  box-shadow: 1px 7px 17px 1px rgba(0,0,0,0.2);  
  z-index: 1;  
}
```

```
.thisisdropdown-content a {  
  color: black;  
  padding: 10px 14px;  
  text-decoration: none;  
  display: block;  
}
```

```
.thisisdropdown-content a:hover {background-color: green}
```

```
.letusdropdown:hover .thisisdropdown-content {  
  display: block;  
}
```

```
.letusdropdown:hover .mydropbutton {  
  background-color: red;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Hoverable Dropdown menus</h4>
```

```
<ul>
```

```
<li><a href="#home">Home Page</a></li>
```

```
<li><a href="#news">News Page</a></li>
```

```
<li class="letusdropdown">
```

```
<a href="javascript:void(0)" class="mydropbutton">Browse Countries  
You Need To Visit</a>
```

```
<div class="thisisdropdown-content">
```

```
<a href="#">Iceland</a>
```

```
<a href="#">Ireland</a>
```

```
<a href="#">India</a>
```

</div>

</li>

</ul>

</body>

</html>

---

## *Hoverable Dropdown Menus*

[Home Page](#) [News Page](#) [Browse Countries You Need To Visit](#)

## ***Image Gallery***

You can add images to your website with a simple code. If you are working on creating an online store for your tourism website, you might need to add images of tourist destinations to your website. You can stack them up by creating a code. Once you have created a sample of the code, you can keep adding more images to add more tourist destinations to the website. Remove the old and worn-out images and replace them with the new ones. See how you can create a code for that and then edit it to see how far you can go with the experiments.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div.1stgallery {
```

```
    margin: 7px;
```

```
    border: 2px solid lightgrey;
```

```
    float: left;
```

```
    width: 200px;
```

```
}
```

```
div.1stgallery:hover {
```

```
    border: 2px solid red;
```

```
}
```

```
div.1stgallery img {
```

```
    width: 120%;
```

```
    height: auto;
```

```
}
```

```
div.description {  
  padding: 20px;  
  text-align: center;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="1stgallery">
```

```
  <a target="_blank" href="italy.jpg">
```

```
    
```

```
  </a>
```

```
  <div class="desc">This is where you can add description for the tourist  
  destination</div>
```

```
</div>
```

```
<div class="1stgallery">
```

```
  <a target="_blank" href="spain.jpg">
```

```
    
```

```
  </a>
```

```
  <div class="desc">Description Space</div>
```

```
</div>
```

```
<div class="1stgallery">
```

```
<a target="_blank" href="Rome.jpg">
  
</a>
<div class="desc">Description Space for the image</div>
</div>

<div class="1stgallery">
  <a target="_blank" href="Germany.jpg">
    
  </a>
  <div class="desc">Description Space</div>
</div>

</body>
</html>
```

---

This is where you can add a description for the tourist destination.

Description Space

## Description Space for the Image

## ***Image Sprites***

An image sprite is usually multiple images that you have put inside one image. A web page that has more than one image takes a long time to load. It tends to generate multiple requests for the server, which makes the process of loading slow. A slow webpage annoys users, and they are less likely to stay on the site. More likely is the fact that they will not return to the web page after one bad experience. Here image sprites come to help. If you use image sprites, you will automatically reduce the total number of requests and save the bandwidth.

Instead of using multiple images, you can use one that has all the images packed inside. You can show a part of the image you need to show to the users. In the following, I will explain which part of the image you should show to the users.

```
<!DOCTYPE html>
<html>
<head>
<style>
#homepage {
    width: 36px;
    height: 54px;
    background: url(spain.gif) 0 0;
}

#nextthingontheline {
    width: 53px;
    height: 34px;
```

```
background: url(italy.gif) -71px 0;
}
</style>
</head>
<body>




</body>
</html>
```

The image's width and height features set up the image portion you need to see. You also can define the background image and set up its position. You can use this CSS feature to form a navigation list. In the next example, I will create an HTML list. See the following code sample.

```
<!DOCTYPE html>
<html>
<head>
<style>
#mynavigationlist {
  position: relative;
}

#mynavigationlist li {
```

```
margin: 1px;
padding: 1px;
list-style: none;
position: absolute;
top: 1px;
}
```

```
#mynavigationlist li, #mynavigationlist a {
  height: 54px;
  display: block;
}
```

```
#myhome {
  left: 1px;
  width: 56px;
  background: url('italy.gif') 0 0;
}
```

```
#mypreviouslist {
  left: 73px;
  width: 53px;
  background: url('italy.gif') -57px 0;
}
```

```
#nextlist {
```

```
left: 119px;
width: 53px;
background: url('italy.gif') -71px 0;
}
</style>
</head>
<body>

<ul id="mynavigationlist">
  <li id="myhome"><a href="mydefaultpage.asp"></a></li>
  <li id="mypreviouslist"><a href="about.asp"></a></li>
  <li id="nextlist"><a href="contact.asp"></a></li>
</ul>

</body>
</html>
```

The position: relative code sets up the position to relative to allow the page to have absolute positioning. You can set up the margins and padding as per your design needs. You can position all the items as absolute positioned. You can take the image sprites to the next level by adding the hover effect to the list. The :hover selector will be used on the elements, excluding the links. I will add three images that will have the hover effect.

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
#mynavigationlist {
  position: relative;
}

#mynavigationlist li {
  margin: 1px;
  padding: 1px;
  list-style: none;
  position: absolute;
  top: 1px;
}

#mynavigationlist li, #mynavigationlist a {
  height: 54px;
  display: block;
}

#myhome {
  left: 1px;
  width: 56px;
  background: url('italy.gif') 0 0;
}

#mypreviouslist {
```

```
left: 73px;
width: 53px;
background: url('italy.gif') -57px 0;
}
```

```
#nextlist {
left: 119px;
width: 53px;
background: url('italy.gif') -71px 0;
}
```

```
#myhome a:hover {
background: url('italy.gif') 0 -35px;
}
```

```
#mypreviouslist a:hover {
background: url('italy.gif') -57px -55px;
}
```

```
#nextlist a:hover {
background: url('italy.gif') -81px -65px;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ul id="mynavigationlist">
```

```
<li id="myhome"><a href="mydefaultpage.asp"></a></li>
```

```
<li id="mypreviouslist"><a href="about.asp"></a></li>
```

```
<li id="nextlist"><a href="contact.asp"></a></li>
```

```
</ul>
```

```
</body>
```

```
</html>
```

## Chapter Eleven: CSS Attributes & Forms

You can style different HTML elements that carry attribute values. The attribute selector is usually used for selecting different elements with the help of a specified attribute. You can use the attribute selector for the selection of elements with the help of a specified attribute. The following example will select the elements with the help of a target attribute.

```
<!DOCTYPE html>
<html>
<head>
<style>
a[target] {
  background-color: lightgrey;
}
</style>
</head>
<body>

<h4>Learning CSS Selector</h4>

<p>The following links that have a target attribute will have yellow
background:</p>

<a href="https://www.sinoform.com">This is sinoform.com</a>
<a href="http://www.treeisgreen.com" target="_blank">This is
treeisgreen.com</a>
<a href="http://www.printmedia.org" target="_top">This is
printmedia.org</a>
```

</body>

</html>

---

## ***Learning CSS Selector***

The following links that have a target attribute will have yellow background:

[This is sinoform.com](#)  
[printmedia.org](#)

[This is treeisgreen.com](#)

[This is](#)

## ***CSS Forms***

The appearance of your HTML form needs to be greatly ameliorated with the help of CSS.

```
<!DOCTYPE html>
<html>
<style>
input[type=thetext], select {
    width: 150%;
    padding: 14px 24px;
    margin: 10px 0;
    display: inline-block;
    border: 2px solid lightgrey;
    border-radius: 6px;
    box-sizing: border-box;
}

input[type=submityourresponse] {
    width: 250%;
    background-color: skyblue;
    color: white;
    padding: 16px 24px;
    margin: 6px 0;
    border: none;
    border-radius: 6px;
    cursor: pointer;
}
```

```
}
```

```
input[type=submityourresponse]:hover {  
  background-color: yellow;  
}
```

```
div {  
  border-radius: 7px;  
  background-color: seagreen;  
  padding: 15px;  
}
```

```
</style>
```

```
<body>
```

```
<h4>Learning CSS to style your HTML Form</h4>
```

```
<div>
```

```
  <form action="/action_page.php">
```

```
    <label for="firstname">First Name</label>
```

```
    <input type="text" id="firstname" name="firstname"  
placeholder="Your first name..">
```

```
    <label for="lastname">Last Name</label>
```

```
    <input type="text" id="lastname" name="lastname" placeholder="Your  
last name..">
```

```
<label for="yourcountry">Country</label>
<select id="yourcountry" name="yourcountry">
  <option value="Ireland">Ireland</option>
  <option value="canada">Canada</option>
  <option value="Iceland">Iceland</option>
</select>

<input type="submityourresponse" value="Submit">
</form>
</div>

</body>
</html>
```

---

## *Learning CSS to style your HTML Form*

---

First Name Last Name Country

You can add more style to the input fields by using the width property.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
input {
```

```
  width: 150%;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is the full-width input field:</p>
```

```
<form>
```

```
  <label for="firstname">Your First Name</label>
```

```
<input type="text" id="firstname" name="firstname">
</form>

</body>
</html>
```

---

This is the full-width input field:

---

Your First Name

---

## Padding Property

You can use the padding property to give some extra padding to the text field. When you are working with more than one input, you may need to add a bit of margin to the code. This will add extra spacing to the outside.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 120%;
  padding: 14px 25px;
  margin: 10px 0;
  box-sizing: border-box;
}
</style>
```

```
</head>
<body>

<p>These are Padded text fields:</p>

<form>
  <label for="firstname">First Name</label>
  <input type="text" id="firstname" name="firstname">
  <label for="lastname">Last Name</label>
  <input type="text" id="lastname" name="lastname">
</form>

</body>
</html>
```

---

These are Padded text fields:

---

First Name

Last Name

---

## More Design Space

Let us add more design to the forms by integrating the CSS border property in the code to alter the color, size, and length of the form. Border-radius property can be used to shape the corners of the input field.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=thetext] {
  width: 120%;
  padding: 14px 25px;
  margin: 10px 0;
  box-sizing: border-box;
  border: 4px solid green;
  border-radius: 6px;
}
}
</style>
</head>
<body>
```

```
<p>These are Padded text fields:</p>
```

```
<form>
  <label for="firstname">First Name</label>
  <input type="thetext" id="firstname" name="firstname">
  <label for="lastname">Last Name</label>
  <input type="thetext" id="lastname" name="lastname">
</form>
```

```
</body>
```

```
</html>
```

---

You can just add the bottom border property to add only the bottom border to the display.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
input[type=thetext] {
```

```
  width: 120%;
```

```
  padding: 14px 25px;
```

```
  margin: 10px 0;
```

```
  box-sizing: border-box;
```

```
  border: none;
```

```
  border-bottom: 4px solid red;
```

```
}
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>These are Padded text fields:</p>
```

```
<form>
```

```
<label for="firstname">First Name</label>
<input type="text" id="firstname" name="firstname">
<label for="lastname">Last Name</label>
<input type="text" id="lastname" name="lastname">
</form>

</body>
</html>
```

---

## Colored Inputs

You also can use the background-color to the code to create a colored background effect for the form. You can give any color to the background.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 120%;
  padding: 14px 25px;
  margin: 10px 0;
  box-sizing: border-box;
  background-color: skyblue;
  color: green;
}
}
</style>
```

```
</head>
<body>

<p>These are Padded text fields:</p>

<form>
  <label for="firstname">First Name</label>
  <input type="text" id="firstname" name="firstname">
  <label for="lastname">Last Name</label>
  <input type="text" id="lastname" name="lastname">
</form>

</body>
</html>
```

---

## Focused Inputs

You can change the style by adding a property that changes the color of the input field when a user focuses on that. I will add the focus property in the following code to see the result.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=text] {
  width: 120%;
  padding: 14px 25px;
```

```
margin: 10px 0;
box-sizing: border-box;
border: 2px solid skyblue;
outline: none;
}
```

```
input[type=thetext]:focus {
  background-color: lightgrey;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>These are focused text fields:</p>
```

```
<form>
```

```
  <label for="firstname">First Name</label>
```

```
  <input type="thetext" id="firstname" name="firstname">
```

```
  <label for="lastname">Last Name</label>
```

```
  <input type="thetext" id="lastname" name="lastname">
```

```
</form>
```

```
</body>
```

```
</html>
```

---

Now the focused fields will have borders.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=thetext] {
  width: 120%;
  padding: 14px 25px;
  margin: 10px 0;
  box-sizing: border-box;
  border: 4px solid red;
  -webkit-transition: 0.5s;
  transition: 0.5s;
  outline: none;
}

input[type=thetext]:focus {
  border: 4px solid yellow;
}

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

<p>These are focused text fields:</p>

<form>

<label for="firstname">First Name</label>

<input type="text" id="firstname" name="firstname">

<label for="lastname">Last Name</label>

<input type="text" id="lastname" name="lastname">

</form>

</body>

</html>

---

You can further modify your search bar by adding a background image to the bar. You can position the same by using the background property. Please note that you should create sufficient space for the icon to adjust in the search bar. The code for the search bar is as under:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
input[type=text] {
```

```
  width: 150%;
```

```
  box-sizing: border-box;
```

```
  border: 4px solid lightgrey;
```

```
  border-radius: 6px;
```

```
  font-size: 14px;
```

```
background-color: skyblue;
background-image: url('someicon.png');
background-position: 12px 12px;
background-repeat: no-repeat;
padding: 14px 25px 14px 60px;
}
</style>
</head>
<body>
```

```
<p>See this input with an icon:</p>
```

```
<form>
  <input type="text" name="search" placeholder="Search here..">
</form>

</body>
</html>
```

---

## Text Areas

You may use the `resize` property for the prevention of text areas from getting resized.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
textarea {
  width: 150%;
  height: 130px;
  padding: 14px 25px;
  box-sizing: border-box;
  border: 4px solid red;
  border-radius: 6px;
  background-color: skyblue;
  font-size: 18px;
  resize: none;
}
</style>
</head>
<body>
```

<p><strong>Pro Tip:</strong> You can apply the resize property for preventing resizing:</p>

```
<form>
  <textarea>Some text...</textarea>
</form>

</body>
</html>
```

---

**Pro Tip:** You can apply the resize property for preventing resizing:



## Menu Styling

You can use the same code with a little edit to style the menus as well. Keep in mind that a single raw HTML element can destroy the look of your website.

```
<!DOCTYPE html>
<html>
<head>
<style>
select {
  width: 150%;
  padding: 18px 25px;
  border: none;
  border-radius: 6px;
  background-color: skyblue;
}
</style>
</head>
<body>

<p>This is a styled menu.</p>
```

```
<form>
  <select id="listofcountry" name="listofcountry">
    <option value="ir">We have reached Ireland.</option>
    <option value="ca">Canada is inviting immigrants.</option>
    <option value="usa">USA is still the super power.</option>
  </select>
</form>

</body>
</html>
```

---

The following code helps in styling the buttons like the submit button or subscribe button.

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type=subscribe], input[type=submit], input[type=Enter] {
  background-color: skyblue;
  border: none;
  color: red;
  padding: 18px 36px;
  text-decoration: none;
  margin: 6px 4px;
  cursor: pointer;
}
```

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

<p>These are Styled Buttons.</p>

<input type="subscribe" value="Subscribe">
<input type="submit" value="Submit">
<input type="Enter" value="Enter">

</body>
</html>
```

---

These are Styled Buttons.



## Responsive Form

You can resize your browser window to see how the form behaves. Mostly, forms are not responsive to changing browser window sizes. When your screen drops below 600px in width, it will be displayed in the form of stacked up columns atop each other. You can create forms to do many tasks like collecting information for contacting or collecting financial information to sell a product. The latter is used for eCommerce stores.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
* {  
  box-sizing: border-box;  
}
```

```
input[type=thetext], theselect, thetextarea {  
  width: 150%;  
  padding: 14px;  
  border: 2px solid skyblue;  
  border-radius: 6px;  
  resize: vertical;  
}
```

```
label {  
  padding: 14px 14px 0;  
  display: inline-block;  
}
```

```
input[type=thesubmit] {  
  background-color: lightgrey;  
  color: red;  
  padding: 14px 25px;  
  border: none;  
  border-radius: 8px;  
  cursor: pointer;  
  float: right;
```

```
}
```

```
input[type=thsubmit]:hover {  
  background-color: yellow;  
}
```

```
.class1 {  
  border-radius: 7px;  
  background-color: seagreen;  
  padding: 30px;  
}
```

```
.class2 {  
  float: left;  
  width: 25%;  
  margin-top: 8px;  
}
```

```
.class3 {  
  float: left;  
  width: 85%;  
  margin-top: 8px;  
}
```

```
/* I am producing clear floats after columns */
```

```
.rowing:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
/* This is a Responsive layout being produced - while the screen has been  
less than the standard 600px wide, making two columns properly stacked  
atop of one another instead of putting them next to one other */
```

```
@media screen and (max-width: 600px) {  
  .class2, .class3, input[type=thesubmit] {  
    width: 150%;  
    margin-top: 0;  
  }  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>You are seeing a Responsive Form</h4>
```

```
<p>Now, you can resize the window of the browser and see the shifting  
effect. When your screen drops below 600px width, it will turn two columns  
properly stacked atop of each other.</p>
```

```
<div class="class1">
```

```
  <form action="/theaction_page.php">
```

```
    <div class="rowing">
```

```
<div class="class2">
  <label for="firstname">First Name</label>
</div>
<div class="class3">
  <input type="text" id="firstname" name="firstname"
placeholder="Your name..">
</div>
</div>
<div class="rowing">
  <div class="class2">
    <label for="lastname">Last Name</label>
  </div>
  <div class="class3">
    <input type="text" id="lastname" name="lastname"
placeholder="Your last name..">
  </div>
</div>
<div class="rowing">
  <div class="class2">
    <label for="mycountry">Country</label>
  </div>
  <div class="class3">
    <select id="mycountry" name="mycountry">
      <option value="ireland">Ireland is green</option>
      <option value="iceland">Iceland is volcanic</option>
      <option value="usa">USA is the superpower</option>
    </select>
  </div>
</div>
```

```
</select>
</div>
</div>
<div class="rowing">
  <div class="class2">
    <label for="thesubject">Submit</label>
  </div>
<div class="class3">
  <textarea id="thesubject" name="thesubject" placeholder="Write
something.." style="height:200px"></textarea>
  </div>
</div>
<div class="rowing">
  <input type="submit" value="Submit">
</div>
</form>
</div>

</body>
</html>
```

---

## ***CSS Counters***

CSS counters are like variables that are maintained by CSS whose values ought to be incremented by the rules that govern CSS. You can track how many times you can use them. CSS counters allow you to adjust to the appearance of the content based on how it is placed inside the document. You will see some properties being used in the Counter feature.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  counter-reset: topic;
}

h4::before {
  counter-increment: topic;
  content: "Topic " counter(topic) ": ";
}
</style>
</head>
<body>

<h4>I am now using the CSS Counters:</h4>
<h4>You are reading HTML Tutorial</h4>
<h4>You are reading CSS Tutorial</h4>
<h4>You are reading JavaScript Tutorial</h4>
```

```
</body>
```

```
</html>
```

---

***Topic 1: I am now using the CSS Counters:***

***Topic 2: You are reading HTML Tutorial***

***Topic 3: You are reading CSS Tutorial***

***Topic 4: You are reading JavaScript Tutorial***

## **Nesting Counters**

You can nest CSS counters on the web page. In the following example, I will create a counter for my web page. Another counter will be for the subsections of h4. We will count the section counter for the h4 elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    counter-reset: topic;
```

```
}
```

```
h4 {
```

```
    counter-reset: subtopic;
```

```
}
```

```
h4::before {
```

```
    counter-increment: topic;
```

```
content: "Topic " counter(topic) ". ";  
}
```

```
p::before {  
  counter-increment: subtopic;  
  content: counter(topic) "." counter(subtopic) " ";  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h4>Europe:</h4>
```

```
<p>Sweden</p>
```

```
<p>Italy</p>
```

```
<p>Spain</p>
```

```
<h4>Asia:</h4>
```

```
<p>India</p>
```

```
<p>China</p>
```

```
<p>Pakistan</p>
```

```
<h4>Americas:</h4>
```

```
<p>Brazil</p>
```

```
<p>USA</p>
```

```
<p>Canada</p>
```

```
</body>
```

```
</html>
```

---

You also can create properly, and well-formatted outlined lists because a single new instance is created inside the child elements.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ol {
```

```
  counter-reset: topic;
```

```
  list-style-type: none;
```

```
}
```

```
li::before {
```

```
  counter-increment: topic;
```

```
  content: counters(topic, ".") " ";
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<ol>
```

```
  <li>content</li>
```

<li>content

<ol>

<li>content</li>

<li>content</li>

<li>content

<ol>

<li>content</li>

<li>content</li>

<li>content</li>

</ol>

</li>

<li>content</li>

</ol>

</li>

<li>content</li>

<li>content</li>

</ol>

<ol>

<li>content</li>

<li>content</li>

</ol>

</body>

</html>

---

1. content

2. content

2.1 content

2.2 content

2.3 content

2.31 content

2.32 content

2.33 content

2.4 content

3 content

4 content

1. content

2. content

## Chapter Twelve: CSS Web Layout

You can divide the layout of your website into menus, headers, a footer, and some content. There are different layout designs to choose from. Mostly, a website has a header, a footer, and content. There are some menus to decorate the pages. A header sits at the top of a web page. It can also be set up just below the top navigation menu. Very often, it contains the name or logo of the website.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Decoding CSS Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
body {
    margin: 1px;
}

/* I am now Styling the website header */
.header {
    background-color: lightgrey;
    padding: 25px;
    text-align: center;
}
</style>
</head>
```

```
<body>
```

```
<div class="header">
```

```
<h4>This Is The Header</h4>
```

```
</div>
```

```
</body>
```

```
</html>
```

---

***This Is The Header***

## ***Navbar***

The second component of the layout is the navigation bar. It is almost inevitable. You must have one as it carries a list of navigation links to navigate your users through a website.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Website Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 1px;
}

/* This code is for Styling your header */
.headerclass {
  background-color: skyblue;
  padding: 25px;
  text-align: center;
}
```

```
/* This code is for Styling your top navigation bar */
```

```
.topnavigationbar {  
  overflow: hidden;  
  background-color: red;  
}
```

```
/* This code is for Styling the topnavigation links */
```

```
.topnavigationbar a {  
  float: left;  
  display: block;  
  color: black;  
  text-align: center;  
  padding: 16px 18px;  
  text-decoration: none;  
}
```

```
/* This code is for Changing the color on the mouse hover */
```

```
.topnavigationbar a:hover {  
  background-color: lightgrey;  
  color: black;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="theheader">
  <h4>This Is Header</h4>
</div>
```

```
<div class="topnavigationbar">
  <a href="#">Home Page</a>
  <a href="#">About Us</a>
  <a href="#">News Corner</a>
</div>
```

```
</body>
</html>
```

---

***This Is Header***

[Home Page](#)[About Us](#)[News Corner](#)

## ***Content***

This section of the layout depends on target users. The most common layout in the website is the one that combines the following different columns. You can use the three-column layout to display your blogs on the page.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>This is my CSS Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 1px;
}

/* This code will Style your header */
.headerwebsite {
  background-color: lightgrey;
  padding: 25px;
  text-align: center;
}
```

```
/* This code will Style your top navigation bar */
```

```
.topnavigationbar {  
  overflow: hidden;  
  background-color: skyblue;  
}
```

```
/* This code will Style your links */
```

```
.topnavigationbar a {  
  float: left;  
  display: block;  
  color: red;  
  text-align: center;  
  padding: 16px 18px;  
  text-decoration: none;  
}
```

```
/* This code will Change the color on the hover */
```

```
.topnavigationbar a:hover {  
  background-color: red;  
  color: black;  
}
```

```
/* Create three equal columns that floats next to each other */
```

```
.columns {
```

```
float: left;
width: 30%;
padding: 15px;
}
```

```
/* Clear floats after the columns */
```

```
.rowing:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* This is a Responsive web layout. You will see three columns properly
stacked atop of each other rather than side by side */
```

```
@media screen and (max-width:600px) {
  .column {
    width: 100%;
  }
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="headerwebsite">
```

```
<h4>This is a Header</h4>
```

```
<p>I am now resizing the browser window so that users can see the
```

responsive effect.</p>

</div>

<div class="topnavigationbar">

<a href="#">Home</a>

<a href="#">Blog</a>

<a href="#">News</a>

</div>

<div class="rowing">

<div class="columns">

<h4>First Columns</h4>

<p>I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

<div class="columns">

<h4>Second Column</h4>

<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

<div class="columns">

<h4>Third Column</h4>

<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

</div>

</body>

</html>

---

## ***This is a Header***

I am now resizing the browser window so that users can see the responsive effect.

[Home](#) [Blog](#) [News](#)

### ***First Columns***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

### ***Second Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

### ***Third Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

## ***Unequal Columns***

Your website may carry unequal columns. The main content is mostly of the largest size. Therefore, it is common to have unequal column widths so that most of the space stays reserved for your major content. The side content is used in the form of alternative navigation to specify the information that is relevant to major content. You can change the width as you like.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>This is my CSS Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 1px;
}

/* This code will Style your header */
.headerwebsite {
  background-color: lightgrey;
  padding: 25px;
  text-align: center;
```

```
}
```

```
/* This code will Style your top navigation bar */
```

```
.topnavigationbar {  
  overflow: hidden;  
  background-color: skyblue;  
}
```

```
/* This code will Style your links */
```

```
.topnavigationbar a {  
  float: left;  
  display: block;  
  color: red;  
  text-align: center;  
  padding: 16px 18px;  
  text-decoration: none;  
}
```

```
/* This code will Change the color on the hover */
```

```
.topnavigationbar a:hover {  
  background-color: red;  
  color: black;  
}
```

```
/* Create three equal columns that floats next to each other */
```

```
.columns {  
  float: left;  
  width: 30%;  
  padding: 15px;  
}
```

```
/* Clear floats after the columns */
```

```
.rowing:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

```
/* These are the right and left columns */
```

```
.columns.side {  
  width: 20%;  
}
```

```
/* This is the Middle column */
```

```
.columns.middle {  
  width: 60%;  
}
```

```
/* This clears the float after your columns */
```

```
.rowing:after {
```

```
content: "";
display: table;
clear: both;
}
```

```
/* This is a Responsive web layout. You will see three columns properly
stacked atop of each other rather than side by side */
```

```
@media screen and (max-width:600px) {
  .column {
    width: 100%;
  }
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="headerwebsite">
```

```
<h4>This is a Header</h4>
```

```
<p>I am now resizing the browser window so that users can see the
responsive effect.</p>
```

```
</div>
```

```
<div class="topnavigationbar">
```

```
<a href="#">Home</a>
```

```
<a href="#">Blog</a>
```

```
<a href="#">News</a>
```

</div>

<div class="rowing">

<div class="columns">

<h4>First Columns</h4>

<p>I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

<div class="columns">

<h4>Second Column</h4>

<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course. I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

<div class="columns">

<h4>Third Column</h4>

<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>

</div>

</div>

</body>

</html>

---

## ***This is a Header***

I am now resizing the browser window so that users can see the responsive effect.

[Home](#) [Blog](#) [News](#)

### ***First Columns***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

### ***Second Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course. I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

### ***Third Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

## ***Footer***

Another component of the website layout is to set up the footer of the website. More often, it is packed up with information such as contact information and copyright information.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>This is my CSS Layout</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 1px;
}

/* This code will Style your header */
.headerwebsite {
  background-color: lightgrey;
  padding: 25px;
  text-align: center;
}
```

```
/* This code will Style your top navigation bar */
```

```
.topnavigationbar {  
  overflow: hidden;  
  background-color: skyblue;  
}
```

```
/* This code will Style your links */
```

```
.topnavigationbar a {  
  float: left;  
  display: block;  
  color: red;  
  text-align: center;  
  padding: 16px 18px;  
  text-decoration: none;  
}
```

```
/* This code will Change the color on the hover */
```

```
.topnavigationbar a:hover {  
  background-color: red;  
  color: black;  
}
```

```
/* Create three equal columns that floats next to each other */
```

```
.columns {
```

```
float: left;
width: 30%;
padding: 15px;
}
```

```
/* Clear floats after the columns */
```

```
.rowing:after {
  content: "";
  display: table;
  clear: both;
}
```

```
/* These are the right and left columns */
```

```
.columns.side {
  width: 20%;
}
```

```
/* This is the Middle column */
```

```
.columns.middle {
  width: 60%;
}
```

```
/* This clears the float after your columns */
```

```
.rowing:after {
  content: "";
```

```
display: table;
clear: both;
}
```

```
/* This is a Responsive web layout. You will see three columns properly
stacked atop of each other rather than side by side */
```

```
@media screen and (max-width:600px) {
  .column {
    width: 100%;
  }
}
```

```
/* This code will Style your footer */
```

```
.footerwebsite {
  background-color: skyblue;
  padding: 12px;
  text-align: center;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="headerwebsite">
```

```
<h4>This is a Header</h4>
```

```
<p>I am now resizing the browser window so that users can see the
responsive effect.</p>
```

```
</div>
```

```
<div class="topnavigationbar">
```

```
<a href="#">Home</a>
```

```
<a href="#">Blog</a>
```

```
<a href="#">News</a>
```

```
</div>
```

```
<div class="rowing">
```

```
<div class="columns">
```

```
<h4>First Columns</h4>
```

```
<p>I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>
```

```
</div>
```

```
<div class="columns">
```

```
<h4>Second Column</h4>
```

```
<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course. </p>
```

```
</div>
```

```
<div class="columns">
```

```
<h4>Third Column</h4>
```

```
<p> I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.</p>
```

```
</div>
```

</div>

<div class="footerwebsite">

<p>This is a Footer</p>

</div>

</body>

</html>

---

*This is a Header*

I am now resizing the browser window so that users can see the responsive effect.

[HomeBlogNews](#)

***First Columns***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

***Second Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

***Third Column***

I like the United Kingdom because of the Victorian-style buildings, castles, pink fields, colorful autumn, and harry potter, of course.

This is a Footer

## **Conclusion**

Now that you have made it to the end of the book, I hope you have a good grasp of CSS. CSS can make or break your website design. If you do it right, it will make the website wonderful, responsive, and user-friendly. If you do it wrong, it will destroy the look of your website. It will drive away users instead of attracting them. It will drop your revenue instead of raising it.

I recommend that you practice all the concepts that I have outlined in the book. You can use the code and alter the values and properties to get alternate displays. The more you practice by changing the code's values, the better you will be at designing a brilliant website.

# References

- CSS Dropdowns*. (n.d.). W3Schools Online Web Tutorials. [https://www.w3schools.com/css/css\\_dropdowns.asp](https://www.w3schools.com/css/css_dropdowns.asp)
- CSS overflow*. (n.d.). W3Schools Online Web Tutorials. [https://www.w3schools.com/css/css\\_overflow.asp](https://www.w3schools.com/css/css_overflow.asp)
- CSS website layout*. (n.d.). W3Schools Online Web Tutorials. [https://www.w3schools.com/css/css\\_website\\_layout.asp](https://www.w3schools.com/css/css_website_layout.asp)
- Duckett, J. (2011). *HTML & CSS*. PDF Drive - Search and download PDF files for free. <https://www.pdfdrive.com/html-css-d16578853.html>
- Nixon, R. (n.d.). PDF Drive - Search and download PDF files for free. <https://www.pdfdrive.com/learning-php-mysql-javascript-css-html5-a-step-by-step-guide-to-creating-dynamic-websites-d158605798.html>