**Unit 1: Cascading Style Sheet (CSS)**

Cascading **S**tyle **S**heets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

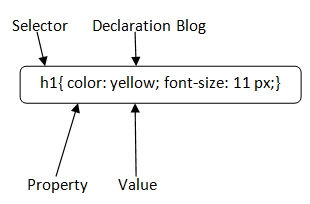
CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Advantages of CSS

* **CSS saves time** − you can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
* **Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
* **Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
* **Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
* **Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
* **Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CSS Syntax

A CSS rule set contains a selector and a declaration block.



**Selector:** Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

**Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. color: yellow;
2. font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

**Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

CSS Selector

**CSS selectors** are used *to select the content you want to style*. Selectors are the part of CSS rule set. CSS selectors select HTML elements according to its id, class, type, attribute etc.

There are several different types of selectors in CSS.

1. CSS Element Selector
2. CSS Id Selector
3. CSS Class Selector
4. CSS Universal Selector
5. CSS Group Selector

1) CSS Element Selector

The element selector selects the HTML element by name.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p{
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<p>**This style will be applied on every paragraph.**</p>**
13. **<p** id="para1"**>**Me too!**</p>**
14. **<p>**And me!**</p>**
15. **</body>**
16. **</html>**

2) CSS Id Selector

The id selector selects the id attribute of an HTML element to select a specific element. An id is always unique within the page so it is chosen to select a single, unique element.

It is written with the hash character (#), followed by the id of the element.

Let?s take an example with the id "para1".

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. #para1 {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<p** id="para1"**>**Hello Javatpoint.com**</p>**
13. **<p>**This paragraph will not be affected.**</p>**
14. **</body>**
15. **</html>**

3) CSS Class Selector

The class selector selects HTML elements with a specific class attribute. It is used with a period character . (full stop symbol) followed by the class name.

Let's take an example with a class "center".

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. .center {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1** class="center"**>**This heading is blue and center-aligned.**</h1>**
13. **<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**
14. **</body>**
15. **</html>**

CSS Class Selector for specific element

If you want to specify that only one specific HTML element should be affected then you should use the element name with class selector.

Let's see an example.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p.center {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1** class="center"**>**This heading is not affected**</h1>**
13. **<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**
14. **</body>**
15. **</html>**

4) CSS Universal Selector

The universal selector is used as a wildcard character. It selects all the elements on the pages.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. \* {
6. color: green;
7. font-size: 20px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h2>**This is heading**</h2>**
13. **<p>**This style will be applied on every paragraph.**</p>**
14. **<p** id="para1"**>**Me too!**</p>**
15. **<p>**And me!**</p>**
16. **</body>**
17. **</html>**

5) CSS Group Selector

The grouping selector is used to select all the elements with the same style definitions.

Grouping selector is used to minimize the code. Commas are used to separate each selector in grouping.

Let's see the CSS code without group selector.

1. h1 {
2. text-align: center;
3. color: blue;
4. }
5. h2 {
6. text-align: center;
7. color: blue;
8. }
9. p {
10. text-align: center;
11. color: blue;
12. }

As you can see, you need to define CSS properties for all the elements. It can be grouped in following ways:

1. h1,h2,p {
2. text-align: center;
3. color: blue;
4. }

Let's see the full example of CSS group selector.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h1, h2, p {
6. text-align: center;
7. color: blue;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1>**Hello Javatpoint.com**</h1>**
13. **<h2>**Hello Javatpoint.com (In smaller font)**</h2>**
14. **<p>**This is a paragraph.**</p>**
15. **</body>**
16. **</html>**

How to add CSS

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example

**<p** style="color:blue"**>**Hello CSS**</p>**

## 2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

1. **<style>**
2. p{color:blue}
3. **</style>**

## 3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

For example:

p{color:blue}

You need to link this style.css file to your html pages like this

**<link** rel="stylesheet" type="text/css" href="style.css"**>**

The link tag must be used inside head section of html.

CSS Properties

CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

## 1) CSS background-color

The background-color property is used to specify the background color of the element.

You can set the background color like this:

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. h2,p{
6. background-color: #b0d4de;
7. }
8. **</style>**
9. **</head>**
10. **<body>**
11. **<h2>**My first CSS page.**</h2>**
12. **<p>**Hello Javatpoint. This is an example of CSS background-color.**</p>**
13. **</body>**
14. **</html>**

2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. body {
6. background-image: url("paper1.gif");
7. margin-left:100px;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1>**Hello Javatpoint.com**</h1>**
13. **</body>**
14. **</html>**

3) CSS background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

The background looks better if the image repeated horizontally only.

**background-repeat: repeat-x;**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. body {
6. background-image: url("gradient\_bg.png");
7. background-repeat: repeat-x;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1>**Hello Javatpoint.com**</h1>**
13. **</body>**
14. **</html>**

**background-repeat: repeat-y;**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. body {
6. background-image: url("gradient\_bg.png");
7. background-repeat: repeat-y;
8. }
9. **</style>**
10. **</head>**
11. **<body>**
12. **<h1>**Hello Javatpoint.com**</h1>**
13. **</body>**
14. **</html>**

4) CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window. If you set fixed the background image then the image will not move during scrolling in the browser. Let?s take an example with fixed background image.

1. background: white url('bbb.gif');
2. background-repeat: no-repeat;
3. background-attachment: fixed;

5) CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

1. center
2. top
3. bottom
4. left
5. right
6. background: white url('good-morning.jpg');
7. background-repeat: no-repeat;
8. background-attachment: fixed;
9. background-position: center;

CSS Colors

The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.

In [CSS](https://www.javatpoint.com/css-full-form)

, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.

We can define the color of an element by using the following ways:

* RGB format.
* RGBA format.
* Hexadecimal notation.
* HSL.
* HSLA.
* Built-in color.

Let's understand the syntax and description of the above ways in detail.

## RGB Format

RGB format is the short form of '**RED GREEN** and **BLUE**' that is used for defining the color of an [HTML](https://www.javatpoint.com/html-tutorial)

element simply by specifying the values of R, G, B that are in the range of 0 to 255.

The color values in this format are specified by using the **rgb()** property. This property allows three values that can either be in percentage or integer (range from 0 to 255).

This property is not supported in all browsers; that's why it is not recommended to use it.

**Syntax**

color: rgb(R, G, B);

## RGBA Format

It is almost similar to RGB format except that **RGBA** contains **A (Alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** is for fully transparent, and **1.0** is for not transparent.

**Syntax**

color:rgba(R, G, B, A);

Hexadecimal notation

Hexadecimal can be defined as a six-digit color representation. This notation starts with the **# symbol** followed by six characters ranges from **0 to F**. In hexadecimal notation, the first two digits represent the **red (RR)** color value, the next two digits represent the **green (GG)** color value, and the last two digits represent the **blue (BB)** color value.

The black color notation in hexadecimal is #000000, and the white color notation in hexadecimal is #FFFFFF. Some of the codes in hexadecimal notation are #FF0000, #00FF00, #0000FF, #FFFF00, and many more

**Syntax**

color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

## Short Hex codes

It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.

For example, #7B6 becomes #77BB66 in hexadecimal.

The black color notation in short hex is #000, and the white color notation in short hex is #FFF. Some of the codes in short hex are #F00, #0F0, #0FF, #FF0, and many more.

## HSL

It is a short form of **Hue, Saturation,** and **Lightness**. Let's understand them individually.

**Hue:** It can be defined as the degree on the color wheel from 0 to 360. 0 represents red, 120 represents green, 240 represents blue.

**Saturation:** It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.

**Lightness:** The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

**Syntax**

1. color:hsl(H, S, L);

HSLA

It is entirely similar to HSL property, except that it contains **A (alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** indicates fully transparent, and **1.0** indicates not transparent.

**Syntax**

1. color:hsla(H, S, L, A);

Built-in Color

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

**Syntax**

1. color: color-name;
2. Let's see the list of built-in colors along with their decimal and hexadecimal values.

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no.** | **Color name** | **Hexadecimal Value** | **Decimal Value or rgb() value** |
| **1.** | Red | #FF0000 | rgb(255,0,0) |
|  |  |  |  |
| **2.** | Orange | #FFA500 | rgb(255,165,0) |
|  |  |  |  |
| **3.** | Yellow | #FFFF00 | rgb(255,255,0) |
|  |  |  |  |
| **4.** | Pink | #FFC0CB | rgb(255,192,203) |
|  |  |  |  |
| **5.** | Green | #008000 | rgb(0,128,0) |
|  |  |  |  |
| **6.** | Violet | #EE82EE | rgb(238,130,238) |
|  |  |  |  |
| **7.** | Blue | #0000FF | rgb(0,0,255) |
|  |  |  |  |
| **8.** | Aqua | #00FFFF | rgb(0,255,255) |
|  |  |  |  |
| **9.** | Brown | #A52A2A | rgb(165,42,42) |
|  |  |  |  |
| **10.** | White | #FFFFFF | rgb(255,255,255) |
|  |  |  |  |
| **11.** | Gray | #808080 | rgb(128,128,128) |
|  |  |  |  |
| **12.** | Black | #000000 | rgb(0,0,0) |

The illustration of [CSS](https://www.javatpoint.com/css-full-form)

colors, which includes the above properties, is given below.

Example

1. **<html>**
2. **<head>**
3. **<title>**CSS hsl color property**</title>**
4. **<style>**
5. h1{
6. text-align:center;
7. }
8. #rgb{
9. color:rgb(255,0,0);
10. }
11. #rgba{
12. color:rgba(255,0,0,0.5);
13. }
14. #hex{
15. color:#EE82EE;
16. }
17. #short{
18. color: #E8E;
19. }
20. #hsl{
21. color:hsl(0,50%,50%);
22. }
23. #hsla{
24. color:hsla(0,50%,50%,0.5);
25. }
26. #built{
27. color:green;
28. }
29. **</style>**
30. **</head>**
31. **<body>**
32. **<h1** id="rgb"**>**
33. Hello World. This is RGB format.
34. **</h1>**
35. **<h1** id="rgba"**>**
36. Hello World. This is RGBA format.
37. **</h1>**
38. **<h1** id="hex"**>**
39. Hello World. This is Hexadecimal format.
40. **</h1>**
41. **<h1** id="short"**>**
42. Hello World. This is Short-hexadecimal format.
43. **</h1>**
44. **<h1** id="hsl"**>**
45. Hello World. This is HSL format.
46. **</h1>**
47. **<h1** id="hsla"**>**
48. Hello World. This is HSLA format.
49. **</h1>**
50. **<h1** id="built"**>**
51. Hello World. This is Built-in color format.
52. **</h1>**
53. **</body>**
54. **</html>**

CSS Border

The CSS border is a shorthand property used to set the border on an element.

The [CSS](https://www.javatpoint.com/css-tutorial)

border properties are use to specify the style, color and size of the border of an element. The CSS border properties are given below

* border-style
* border-color
* border-width
* border-radius

1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

|  |  |
| --- | --- |
| **alue** | **Description** |
| none | It doesn't define any border. |
| dotted | It is used to define a dotted border. |
| dashed | It is used to define a dashed border. |
| solid | It is used to define a solid border. |
| double | It defines two borders wIth the same border-width value. |
| groove | It defines a 3d grooved border. effect is generated according to border-color value. |
| ridge | It defines a 3d ridged border. effect is generated according to border-color value. |
| inset | It defines a 3d inset border. effect is generated according to border-color value. |
| outset | It defines a 3d outset border. effect is generated according to border-color value. |

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p.none {border-style: none;}
6. p.dotted {border-style: dotted;}
7. p.dashed {border-style: dashed;}
8. p.solid {border-style: solid;}
9. p.double {border-style: double;}
10. p.groove {border-style: groove;}
11. p.ridge {border-style: ridge;}
12. p.inset {border-style: inset;}
13. p.outset {border-style: outset;}
14. p.hidden {border-style: hidden;}
15. **</style>**
16. **</head>**
17. **<body>**
18. **<p** class="none"**>**No border.**</p>**
19. **<p** class="dotted"**>**A dotted border.**</p>**
20. **<p** class="dashed"**>**A dashed border.**</p>**
21. **<p** class="solid"**>**A solid border.**</p>**
22. **<p** class="double"**>**A double border.**</p>**
23. **<p** class="groove"**>**A groove border.**</p>**
24. **<p** class="ridge"**>**A ridge border.**</p>**
25. **<p** class="inset"**>**An inset border.**</p>**
26. **<p** class="outset"**>**An outset border.**</p>**
27. **<p** class="hidden"**>**A hidden border.**</p>**
28. **</body>**
29. **</html>**

## 2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p.one {
6. border-style: solid;
7. border-width: 5px;
8. }
9. p.two {
10. border-style: solid;
11. border-width: medium;
12. }
13. p.three {
14. border-style: solid;
15. border-width: 1px;
16. }
17. **</style>**
18. **</head>**
19. **<body>**
20. **<p** class="one"**>**Write your text here.**</p>**
21. **<p** class="two"**>**Write your text here.**</p>**
22. **<p** class="three"**>**Write your text here.**</p>**
23. **</body>**
24. **</html>**

## 3) CSS border-color

There are three methods to set the color of the border.

* Name: It specifies the color name. For example: "red".
* RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
* Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p.one {
6. border-style: solid;
7. border-color: red;
8. }
9. p.two {
10. border-style: solid;
11. border-color: #98bf21;
12. }
13. **</style>**
14. **</head>**
15. **<body>**
16. **<p** class="one"**>**This is a solid red border**</p>**
17. **<p** class="two"**>**This is a solid green border**</p>**
18. **</body>**
19. **</html>**

# CSS Margin

CSS Margin property is used to define the space around elements. It is completely transparent and doesn't have any background color. It clears an area around the element.

Top, bottom, left and right margin can be changed independently using separate properties. You can also change all properties at once by using shorthand margin property.

There are following [CSS](https://www.javatpoint.com/css-tutorial)

margin properties:

## CSS Margin Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| margin | This property is used to set all the properties in one declaration. |
| margin-left | it is used to set left margin of an element. |
| margin-right | It is used to set right margin of an element. |
| margin-top | It is used to set top margin of an element. |
| margin-bottom | It is used to set bottom margin of an element. |

## CSS Margin Values

These are some possible values for margin property.

|  |  |
| --- | --- |
| **Value** | **Description** |
| auto | This is used to let the browser calculate a margin. |
| length | It is used to specify a margin pt, px, cm, etc. its default value is 0px. |
| % | It is used to define a margin in percent of the width of containing element. |
| inherit | It is used to inherit margin from parent element. |

CSS margin Example

You can define different margin for different sides for an element.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.ex {
9. margin-top: 50px;
10. margin-bottom: 50px;
11. margin-right: 100px;
12. margin-left: 100px;
13. }
14. **</style>**
15. **</head>**
16. **<body>**
17. **<p>**This paragraph is not displayed with specified margin. **</p>**
18. **<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**
19. **</body>**
20. **</html>**

Margin: Shorthand Property

CSS shorthand property is used to shorten the code. It specifies all the margin properties in one property.

There are four types to specify the margin property. You can use one of them.

1. margin: 50px 100px 150px 200px;
2. margin: 50px 100px 150px;
3. margin: 50px 100px;
4. margin 50px;

1) margin: 50px 100px 150px 200px;

It identifies that:

**top** margin value is 50px

**right** margin value is 100px

**bottom** margin value is 150px

**left** margin value is 200px

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.ex {
9. margin: 50px 100px 150px 200px;
10. }
11. **</style>**
12. **</head>**
13. **<body>**
14. **<p>**This paragraph is not displayed with specified margin. **</p>**
15. **<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**
16. **</body>**
17. **</html>**

2) margin: 50px 100px 150px;

It identifies that:

**top** margin value is 50px

**left and right** margin values are 100px

**bottom** margin value is 150px

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.ex {
9. margin: 50px 100px 150px;
10. }
11. **</style>**
12. **</head>**
13. **<body>**
14. **<p>**This paragraph is not displayed with specified margin. **</p>**
15. **<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**
16. **</body>**
17. **</html>**

3) margin: 50px 100px;

It identifies that:

**top and bottom** margin values are 50px

**left and right** margin values are 100px

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.ex {
9. margin: 50px 100px;
10. }
11. **</style>**
12. **</head>**
13. **<body>**
14. **<p>**This paragraph is not displayed with specified margin. **</p>**
15. **<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**
16. **</body>**
17. **</html>**

4) Margin: 50px;

It identifies that:

**top right bottom and left** margin values are 50px

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.ex {
9. margin: 50px;
10. }
11. **</style>**
12. **</head>**
13. **<body>**
14. **<p>**This paragraph is not displayed with specified margin. **</p>**
15. **<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**
16. **</body>**
17. **</html>**

CSS Padding

**CSS Padding property** is used *to define the space between the element content and the element border*.

It is different from CSS margin in the way that CSS margin defines the space around elements. CSS padding is affected by the background colors. It clears an area around the content.

Top, bottom, left and right padding can be changed independently using separate properties. You can also change all properties at once by using shorthand padding property.

CSS Padding Properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| padding | It is used to set all the padding properties in one declaration. |
| padding-left | It is used to set left padding of an element. |
| padding-right | It is used to set right padding of an element. |
| padding-top | It is used to set top padding of an element. |
| padding-bottom | It is used to set bottom padding of an element. |

CSS Padding Values

|  |  |
| --- | --- |
| **Value** | **Description** |
| length | It is used to define fixed padding in pt, px, em etc. |
| % | It defines padding in % of containing element. |

CSS Padding Example

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. p {
6. background-color: pink;
7. }
8. p.padding {
9. padding-top: 50px;
10. padding-right: 100px;
11. padding-bottom: 150px;
12. padding-left: 200px;
13. }
14. **</style>**
15. **</head>**
16. **<body>**
17. **<p>**This is a paragraph with no specified padding.**</p>**
18. **<p** class="padding"**>**This is a paragraph with specified paddings.**</p>**
19. **</body>**
20. **</html>**

CSS Text

Text Color

The color property is used to set the color of the text. The color is specified by:

* a color name - like "red"
* a HEX value - like "#ff0000"
* an RGB value - like "rgb(255,0,0)"

Look at [CSS Color Values](https://www.w3schools.com/cssref/css_colors_legal.asp) for a complete list of possible color values.

The default text color for a page is defined in the body selector.

### Example

body{

color:blue;  
 }  
  
h1{  
 color:green;  
}

Text Color and Background Color

Body

{background-color:lightgrey;  
 color:blue;  
 }  
  
h1{

background-color:black;  
 color:white;  
}  
  
div{  
 background-color:blue;  
 color:white;  
}

CSS Text Alignment and Text Direction

* text-align
* text-align-last
* direction
* unicode-bidi
* vertical-align

h1{  
 text-align:center;  
}  
  
h2{  
  text-align:left;  
}  
  
h3{  
  text-align:right;  
}

CSS Text Alignment

* text-align
* text-align-last
* direction
* unicode-bidi
* vertical-align

A text can be left or right aligned, centered, or justified.

h1 {

text-align: center;

}

The text-align-last property specifies how to align the last line of a text.

p.a {

  text-align-last: right;

}

CSS Text Decoration

* text-decoration-line
* text-decoration-color
* text-decoration-style
* text-decoration-thickness
* text-decoration

The text-decoration-line property is used to add a decoration line to text.

h1 {

text-decoration-line: overline;

}

h2 {

text-decoration-line: line-through;

}

h3 {

text-decoration-line: underline;

}

p {

text-decoration-line: overline underline;

}

The text-decoration-color property is used to set the color of the decoration line.

h3 {

text-decoration-line: underline;

text-decoration-color: blue;

}

The text-decoration-style property is used to set the style of the decoration line values can be solid, double, dotted, dashed, wavy

h3 {

text-decoration-line: underline;

text-decoration-style: dashed;

}

CSS Flexbox:

CSS3 Flexible boxes also known as CSS Flexbox, is a new layout mode in CSS3.

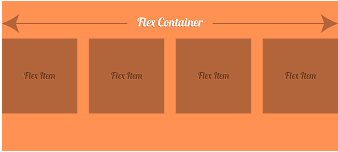
The CSS3 flexbox is used to make the elements behave predictably when they are used with different screen sizes and different display devices. It provides a more efficient way to layout, align and distribute space among items in the container.

It is mainly used to make CSS3 capable to change its items width and height to best fit for all available spaces. It is preferred over block model.

The CSS3 flexbox contains flex containers and flex items.

**Flex container:** The flex container specifies the properties of the parent. It is declared by setting the **display property of an element to either flex or inline-flex.**

**Flex items:** The flex items specify properties of the children. There may be one or more flex items inside a flex container.



**Note:** Flexbox specifies how flex items are set inside a flex container. It sets the flex items inside a flex container along a flex line. By default, there is only one flex line per flex container. Everything outside a flex container and inside a flex item is considered as usual.

Let's take an example to show three flex items within a flex container. By default, they are set along the horizontal flex line, from left to right:

**See this example:**

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head><meta** http-equiv="Content-Type" content="text/html; charset=windows-1252"**>**
4. **<style>**
5. .flex-container {
6. display: -webkit-flex;
7. display: flex;
8. width: 400px;
9. height: 200px;
10. background-color: lightpink;
11. }
12. .flex-item {
13. background-color: brown;
14. width: 100px;
15. height: 100px;
16. margin: 10px;
17. }
18. **</style>**
19. **</head>**
20. **<body>**
21. **<div** class="flex-container"**>**
22. **<div** class="flex-item"**>**flex item 1**</div>**
23. **<div** class="flex-item"**>**flex item 2**</div>**
24. **<div** class="flex-item"**>**flex item 3**</div>**
25. **</div>**
26. **</body>**
27. **</html>**

CSS3 Flexbox Properties

Following is a list of CSS3 Flexbox properties:

|  |  |
| --- | --- |
| **property** | **description** |
| display | it is used to specify the type of box used for an html element. |
| flex-direction | it is used to specify the direction of the flexible items inside a flex container.  Values can be: row-reverse;  column;  column-reverse; |
| justify-content | it is used to align the flex items horizontally when the items do not use all available space on the main-axis.  Values can be: flex-end;  center;  space-between; space-around; |
| align-items | it is used to align the flex items vertically when the items do not use all available space on the cross-axis.  Values can be: stretch;  flex-start;  flex-end; center; baseline; |
| flex-wrap | it specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line.   Values can be: nowrap; wrap; wrap-reverse; |
| align-content | it is used to modify the behavior of the flex-wrap property. it is similar to align-items, but instead of aligning flex items, it aligns flex lines. |
| flex-flow | it specifies a shorthand property for flex-direction and flex-wrap. |
| order | it specifies the order of a flexible item relative to the rest of the flex items inside the same container. |
| align-self | it is used on flex items. it overrides the container's align-items property. |
| flex | it specifies the length of a flex item, relative to the rest of the flex items inside the same |

# CSS Grid

A grid can be defined as an intersecting set of horizontal lines and vertical lines. CSS Grid layout divides a page into major regions. It defines the relationship between the parts of a control built from HTML primitives in terms of layer, position, and size. Grid property offers a grid-based layout system having rows and columns. It makes the designing of web pages easy without positioning and floating.

Similar to the table, it enables a user to align the elements into rows and columns. But compare to tables, it is easy to design layout with the CSS grid. We can define columns and rows on the grid by using **grid-template-rows** and **grid-template-columns** properties.

## Grid Container

We can define the grid container by setting the **display** property to **grid** or **inline-grid** on an element.

Grid container contains grid items that are placed inside rows and columns.

Let's see a simple example of a grid in CSS.

### Example

1. <!DOCTYPE html**>**
2. **<html>**
4. **<head>**
6. **<style>**
7. .main {
8. display: grid;
9. grid: auto auto / auto auto auto auto;
10. grid-gap: 10px;
11. background-color: black;
12. padding: 10px;
13. }
15. .num {
16. background-color: grey;
17. text-align: center;
18. color: white;
19. padding: 10px 10px;
20. font-size: 30px;
21. }
22. **</style>**
23. **</head>**
25. **<body>**
26. **<div** class="main"**>**
27. **<div** class="num"**>**One**</div>**
28. **<div** class="num"**>**Two**</div>**
29. **<div** class="num"**>**Three**</div>**
30. **<div** class="num"**>**Four**</div>**
31. **<div** class="num"**>**Five**</div>**
32. **<div** class="num"**>**Six**</div>**
33. **<div** class="num"**>**Seven**</div>**
34. **<div** class="num"**>**Eight**</div>**
35. **</div>**
37. **</body>**
39. **</html>**

CSS Animation

**CSS Animation property** is used *to create animation on the webpage*. It can be used as a replacement of animation created by Flash and [JavaScript](https://javatpoint.com/javascript-tutorial).

CSS3 @keyframes Rule

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a [CSS](https://javatpoint.com/css-tutorial) animation sequence.

What animation does:

An animation *makes an element change gradually* from one style to another. You can add as many as properties you want to add. You can also specify the changes in percentage.0% specify the start of the animation and 100% specify its completion.

How CSS animation works

When the animation is created in the [@keyframe rule](https://www.javatpoint.com/css-keyframes-rule), it must be bound with selector; otherwise the animation will have no effect.

The animation could be bound to the selector by specifying at least these two properties:

* The name of the animation
* The duration of the animation

AD CSS animation properties

|  |  |
| --- | --- |
| **Property** | **Description** |
| @keyframes | It is used to specify the animation. |
| animation | This is a shorthand property, used for setting all the properties, except the animation-play-state and the animation-fill- mode property. |
| animation-delay | It specifies when the animation will start. |
| animation-direction | It specifies if or not the animation should play in reserve on alternate cycle. |
| animation-duration | It specifies the time duration taken by the animation to complete one cycle. |
| animation-fill-mode | it specifies the static style of the element. (when the animation is not playing) |
| animation-iteration-count | It specifies the number of times the animation should be played. |
| animation-play-state | It specifies if the animation is running or paused. |
| animation-name | It specifies the name of @keyframes animation. |
| animation-timing-function | It specifies the speed curve of the animation. |

CSS animation example: changing background color

Let's see a simple CSS animation example that changes background color of rectangle from RED to BLACK.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div {
6. width: 100px;
7. height: 100px;
8. background: red;
9. -webkit-animation: myfirst 6s; /\* Chrome, Safari, Opera \*/
10. animation: myfirst 5s;
11. }
12. /\* Chrome, Safari, Opera \*/
13. @-webkit-keyframes myfirst {
14. from {background: red;}
15. to {background: green;}
16. }
17. /\* Standard syntax \*/
18. @keyframes myfirst {
19. from {background: red;}
20. to {background: green;}
21. }
22. **</style>**
23. **</head>**
24. **<body>**
25. **<p><b>**Note:**</b>** The IE 9 and earlier versions don't support CSS3 animation property. **</p>**
26. **<div></div>**
27. **</body>**
28. **</html>**

CSS animation example: Moving Rectangle

Let's take another example to display animation with percentage value.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div {
6. width: 100px;
7. height: 100px;
8. background: red;
9. position: relative;
10. -webkit-animation: myfirst 5s; /\* Chrome, Safari, Opera \*/
11. animation: myfirst 5s;
12. }
13. /\* Chrome, Safari, Opera \*/
14. @-webkit-keyframes myfirst {
15. 0%   {background:red; left:0px; top:0px;}
16. 25%  {background:yellow; left:300px; top:0px;}
17. 50%  {background:blue; left:200px; top:300px;}
18. 75%  {background:green; left:0px; top:200px;}
19. 100% {background:red; left:0px; top:0px;}
20. }
21. /\* Standard syntax \*/
22. @keyframes myfirst {
23. 0%   {background:red; left:0px; top:0px;}
24. 25%  {background:yellow; left:300px; top:0px;}
25. 50%  {background:blue; left:300px; top:200px;}
26. 75%  {background:green; left:0px; top:200px;}
27. 100% {background:red; left:0px; top:0px;}
28. }
29. **</style>**
30. **</head>**
31. **<body>**
32. **<p><b>**Note:**</b>** The Internet Explorer 9 and its earlier versions don't support this example.**</p>**
33. **<div></div>**
34. **</body>**
35. **</html>**

CSS Transition

The CSS transitions are effects that are added to change the element gradually from one style to another, without using flash or JavaScript.

You should specify two things to create CSS transition.

* The CSS property on which you want to add an effect.
* The time duration of the effect.

Let's take an example which defines transition effect on width property and duration of 3 seconds.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div{
6. width: 100px;
7. height: 100px;
8. background: orange;
9. -webkit-transition: width 1s; /\* For Safari 3.1 to 6.0 \*/
10. transition: width 1s;
11. }
12. div:hover {
13. width: 300px;
14. }
15. **</style>**
16. **</head>**
17. **<body>**
18. **<div></div>**
19. **<p>**Move the cursor over the div element above, to see the transition effect.**</p>**
20. **</body>**
21. **</html>**

CSS Multiple Transition Effect

It is used to add transition effect for more than one CSS property. If you want to add transition effect on more than one property, separate those properties with a comma.

Let's take an example. Here, the transition effects on width, height and transformation.

1. <!DOCTYPE html**>**
2. **<html>**
3. **<head>**
4. **<style>**
5. div {
6. padding:15px;
7. width: 150px;
8. height: 100px;
9. background: violet;
10. -webkit-transition: width 2s, height 2s, -webkit-transform 2s; /\* For Safari 3.1 to 6.0 \*/
11. transition: width 2s, height 2s, transform 2s;
12. }
13. div:hover {
14. width: 300px;
15. height: 200px;
16. -webkit-transform: rotate(360deg); /\* Chrome, Safari, Opera \*/
17. transform: rotate(360deg);
18. }
19. **</style>**
20. **</head>**
21. **<body>**
22. **<div><b>**JavaTpoint.com**</b><p>** (Move your cursor on me to see the effect)**</p></div>**
23. **</body>**
24. **</html>**