**CSS Tutorial**



**CSS tutorial** or CSS 3 tutorial provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

* CSS stands for Cascading Style Sheet.
* CSS is used to design HTML tags.
* CSS is a widely used language on the web.
* HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

**CSS Example with CSS Editor**

In this tutorial, you will get a lot of CSS examples; you can edit and run these examples with our online CSS editor tool.

<!DOCTYPE**>**

**<html>**

**<head>**

**<style>**

h1{

color:white;

background-color:red;

padding:5px;

}

p{

color:blue;

}

**</style>**

**</head>**

**<body>**

**<h1>**Write Your First CSS Example**</h1>**

**<p>**This is Paragraph.**</p>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=csscss1)

Output:

**Write Your First CSS Example**

This is Paragraph.

**CSS 3 Tutorial**

In this tutorial, we will learn CSS 3 properties to design box model, apply opacity, radius etc.

**All CSS Properties**

In this tutorial, you will get details of all CSS properties such as background, border, font, float, display, margin, opacity, padding, text-align, vertical-align, position, color etc.

**What is CSS**

CSS is an acronym stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

**What does CSS do**

* You can add new looks to your old HTML documents.
* You can completely change the look of your website with only a few changes in CSS code.

**Why use CSS**

These are the three major benefits of CSS:

**1) Solves a big problem**

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

**2) Saves a lot of time**

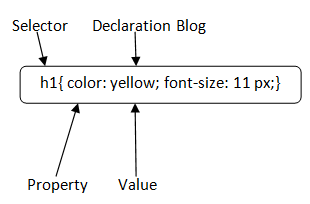
CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

**3) Provide more attributes**

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

**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.

Selector{Property1: value1; Property2: value2; ..........;}

# 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.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p{

    text-align: center;

    color: blue;

}

**</style>**

**</head>**

**<body>**

**<p>**This style will be applied on every paragraph.**</p>**

**<p** id="para1"**>**Me too!**</p>**

**<p>**And me!**</p>**

**</body>**

**</html>**

**Output:**

This style will be applied on every paragraph.

Me too!

And me!

## 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".

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#para1 {

    text-align: center;

    color: blue;

}

**</style>**

**</head>**

**<body>**

**<p** id="para1"**>**Hello Zerozilla.com**</p>**

**<p>**This paragraph will not be affected.**</p>**

**</body>**

**</html>**

**Output:**

Hello Zerozilla.com

This paragraph will not be affected.

## 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.

#### Note: A class name should not be started with a number.

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

.center {

    text-align: center;

    color: blue;

}

**</style>**

**</head>**

**<body>**

**<h1** class="center"**>**This heading is blue and center-aligned.**</h1>**

**<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**

**</body>**

**</html>**

**Output:**

## This heading is blue and center-aligned.

This paragraph is blue and center-aligned.

## 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.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.center {

    text-align: center;

    color: blue;

}

**</style>**

**</head>**

**<body>**

**<h1** class="center"**>**This heading is not affected**</h1>**

**<p** class="center"**>**This paragraph is blue and center-aligned.**</p>**

**</body>**

**</html>**

**Output:**

## This heading is not affected

This paragraph is blue and center-aligned.

## 4) CSS Universal Selector

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

\* {

   color: green;

   font-size: 20px;

}

**</style>**

**</head>**

**<body>**

**<h2>**This is heading**</h2>**

**<p>**This style will be applied on every paragraph.**</p>**

**<p** id="para1"**>**Me too!**</p>**

**<p>**And me!**</p>**

**</body>**

**</html>**

**Output:**

## This is heading

This style will be applied on every paragraph.

Me too!

And me!

## 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.

h1 {

    text-align: center;

    color: blue;

}

h2 {

    text-align: center;

    color: blue;

}

p {

    text-align: center;

    color: blue;

}

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

h1,h2,p {

    text-align: center;

    color: blue;

}

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h1, h2, p {

    text-align: center;

    color: blue;

}

**</style>**

**</head>**

**<body>**

**<h1>**Hello Zerozilla.com**</h1>**

**<h2>**Hello Zerozilla.com (In smaller font)**</h2>**

**<p>**This is a paragraph.**</p>**

**</body>**

**</html>**

**Output:**

## Hello Zerozilla.com

### Hello Zerozilla.com (In smaller font)

This is a paragraph.

**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:**

**<style>**

p{color:blue}

**</style>**

For more visit here: [Internal CSS](https://www.javatpoint.com/internal-css)

**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.

**Inline CSS**

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

**Syntax:**

**<htmltag** style="cssproperty1:value; cssproperty2:value;"**>** **</htmltag>**

**Example:**

**<h2** style="color:red;margin-left:40px;"**>**Inline CSS is applied on this heading.**</h2>**

**<p>**This paragraph is not affected.**</p>**

**Output:**

**Inline CSS is applied on this heading.**

This paragraph is not affected.

**Disadvantages of Inline CSS**

* You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
* These styles cannot be reused anywhere else.
* These styles are tough to be edited because they are not stored at a single place.
* It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
* Inline CSS does not provide browser cache advantages.

**Internal CSS**

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

Example:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    background-color: linen;

}

h1 {

    color: red;

    margin-left: 80px;

}

**</style>**

**</head>**

**<body>**

**<h1>**The internal style sheet is applied on this heading.**</h1>**

**<p>**This paragraph will not be affected.**</p>**

**</body>**

**</html>**

**Output:**

# The internal style sheet is applied on this heading.

This paragraph will not be affected.

**External CSS**

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

**Example:**

**<head>**

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

**</head>**

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

Let's take an example of a style sheet file named "mystyle.css".

***File: mystyle.css***

body {

    background-color: lightblue;

}

h1 {

    color: navy;

    margin-left: 20px;

}

Note: You should not use a space between the property value and the unit. For example: It should be margin-left:20px not margin-left:20 px.

**CSS Comments**

CSS comments are generally written to explain your code. It is very helpful for the users who reads your code so that they can easily understand the code.

Comments are ignored by browsers.

Comments are single or multiple lines statement and written within /\*............\*/ .

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    color: blue;

    /\* This is a single-line comment \*/

    text-align: center;

}

/\* This is

a multi-line

comment \*/

**</style>**

**</head>**

**<body>**

**<p>**Hello Zerozilla.com**</p>**

**<p>**This statement is styled with CSS.**</p>**

**<p>**CSS comments are ignored by the browsers and not shown in the output.**</p>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=csscomments1)

**Output:**

Hello Zerozilla.com

This statement is styled with CSS.

CSS comments are ignored by the browsers and not shown in the output.

**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:

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h2,p{

    background-color: #b0d4de;

}

**</style>**

**</head>**

**<body>**

**<h2>**My first CSS page.**</h2>**

**<p>**Hello Zerozilla. This is an example of CSS background-color.**</p>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssbackground1)

Output:

**My first CSS page.**

Hello Zerozilla. This is an example of CSS background-color.

**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.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

background-image: url("paper1.gif");

margin-left:100px;

}

**</style>**

**</head>**

**<body>**

**<h1>**Hello Zerozilla.com**</h1>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssbackground2)

Note: The background image should be chosen according to text color. The bad combination of text and background image may be a cause of poor designed and not readable webpage.

**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;**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    background-image: url("gradient\_bg.png");

    background-repeat: repeat-x;

}

**</style>**

**</head>**

**<body>**

**<h1>**Hello Zerozilla.com**</h1>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssbackground3)

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    background-image: url("gradient\_bg.png");

    background-repeat: repeat-y;

}

**</style>**

**</head>**

**<body>**

**<h1>**Hello Zerozilla.com**</h1>**

**</body>**

**</html>**   [**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssbackground3y)

**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.

background: white url('bbb.gif');

background-repeat: no-repeat;

background-attachment: fixed;  [**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssbackground4)

**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 Border

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

The CSS 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.

|  |  |
| --- | --- |
| **Value** | **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. |

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.none {border-style: none;}

p.dotted {border-style: dotted;}

p.dashed {border-style: dashed;}

p.solid {border-style: solid;}

p.double {border-style: double;}

p.groove {border-style: groove;}

p.ridge {border-style: ridge;}

p.inset {border-style: inset;}

p.outset {border-style: outset;}

p.hidden {border-style: hidden;}

**</style>**

**</head>**

**<body>**

**<p** class="none"**>**No border.**</p>**

**<p** class="dotted"**>**A dotted border.**</p>**

**<p** class="dashed"**>**A dashed border.**</p>**

**<p** class="solid"**>**A solid border.**</p>**

**<p** class="double"**>**A double border.**</p>**

**<p** class="groove"**>**A groove border.**</p>**

**<p** class="ridge"**>**A ridge border.**</p>**

**<p** class="inset"**>**An inset border.**</p>**

**<p** class="outset"**>**An outset border.**</p>**

**<p** class="hidden"**>**A hidden border.**</p>**

**</body>**

**</html>**

**Output:**

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

## 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.

#### Note: The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.one {

    border-style: solid;

    border-width: 5px;

}

p.two {

    border-style: solid;

    border-width: medium;

}

p.three {

    border-style: solid;

    border-width: 1px;

}

**</style>**

**</head>**

**<body>**

**<p** class="one"**>**Write your text here.**</p>**

**<p** class="two"**>**Write your text here.**</p>**

**<p** class="three"**>**Write your text here.**</p>**

**</body>**

**</html>**

**Output:**

Write your text here.

Write your text here.

Write your text here.

## 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.

#### Note: The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.one {

    border-style: solid;

    border-color: red;

}

p.two {

    border-style: solid;

    border-color: #98bf21;

}

**</style>**

**</head>**

**<body>**

**<p** class="one"**>**This is a solid red border**</p>**

**<p** class="two"**>**This is a solid green border**</p>**

**</body>**

**</html>**

**Output:**

This is a solid red border

This is a solid green border

**CSS Display**

CSS display is the most important property of CSS which is used to control the layout of the element. It specifies how the element is displayed.

Every element has a default display value according to its nature. Every element on the webpage is a rectangular box and the CSS property defines the behavior of that rectangular box.

**CSS Display default properties**

|  |  |
| --- | --- |
| default value | inline |
| inherited | no |
| animation supporting | no |
| version | css1 |
| javascript syntax | object.style.display="none" |

**Syntax**

display:value;

**CSS display values**

There are following CSS display values which are commonly used.

1. display: inline;
2. display: inline-block;
3. display: block;
4. display: run-in;
5. display: none;

**1) CSS display inline**

The inline element takes the required width only. It doesn't force the line break so the flow of text doesn't break in inline example.

The inline elements are:

* <span>
* <a>
* <em>
* <b> etc.

Let's see an example of CSS display inline.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

display: inline;

}

**</style>**

**</head>**

**<body>**

**<p>**Hello Zerozilla.com**</p>**

**<p>**Java Tutorial.**</p>**

**<p>**SQL Tutorial.**</p>**

**<p>**HTML Tutorial.**</p>**

**<p>**CSS Tutorial.**</p>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssdisplay1)

**Output:**

Hello Zerozilla.com Java Tutorial. SQL Tutorial. HTML Tutorial. CSS Tutorial.

**2) CSS display inline-block**

The CSS display inline-block element is very similar to inline element but the difference is that you are able to set the width and height.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

display: inline-block;

}

**</style>**

**</head>**

**<body>**

**<p>**Hello Zerozilla.com**</p>**

**<p>**Java Tutorial.**</p>**

**<p>**SQL Tutorial.**</p>**

**<p>**HTML Tutorial.**</p>**

**<p>**CSS Tutorial.**</p>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssdisplay2)

**Output:**

Hello Zerozilla.com Java Tutorial. SQL Tutorial. HTML Tutorial. CSS Tutorial.

**3) CSS display block**

The CSS display block element takes as much as horizontal space as they can. Means the block element takes the full available width. They make a line break before and after them.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

display: block;

}

**</style>**

**</head>**

**<body>**

**<p>**Hello Zerozilla.com**</p>**

**<p>**Java Tutorial.**</p>**

**<p>**SQL Tutorial.**</p>**

**<p>**HTML Tutorial.**</p>**

**<p>**CSS Tutorial.**</p>**

**</body>**

**</html>**

[**T**](http://www.javatpoint.com/oprweb/test.jsp?filename=cssdisplay3)

**Output:**

Hello Zerozilla.com

Java Tutorial.

SQL Tutorial.

HTML Tutorial.

CSS Tutorial.

**4) CSS display run-in**

This property doesn?t work in Mozilla Firefox. These elements don't produce a specific box by themselves.

* If the run-in box contains a bock box, it will be same as block.
* If the block box follows the run-in box, the run-in box becomes the first inline box of the block box.
* If the inline box follows the run-in box, the run-in box becomes a block box.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

display: run-in;

}

**</style>**

**</head>**

**<body>**

**<p>**Hello Zerozilla.com**</p>**

**<p>**Java Tutorial.**</p>**

**<p>**SQL Tutorial.**</p>**

**<p>**HTML Tutorial.**</p>**

**<p>**CSS Tutorial.**</p>**

**</body>**

**</html>**

**Output:**

Hello Zerozilla.com

Java Tutorial.

SQL Tutorial.

HTML Tutorial.

CSS Tutorial.

**5) CSS display none**

The "none" value totally removes the element from the page. It will not take any space.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h1.hidden {

    display: none;

}

**</style>**

**</head>**

**<body>**

**<h1>**This heading is visible.**</h1>**

**<h1** class="hidden"**>**This is not visible.**</h1>**

**<p>**You can see that the hidden heading does not contain any space.**</p>**

**</body>**

**</html>**

**Output:**

**This heading is visible.**

You can see that the hidden heading does not contain any space.

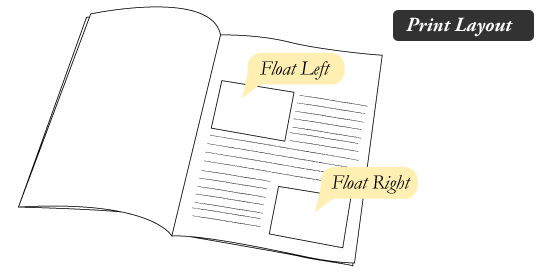
**Other CSS display values**

|  |  |
| --- | --- |
| **Property-value** | **Description** |
| flex | It is used to display an element as an block-level flex container. It is new in css3. |
| inline-flex | It is used to display an element as an inline-level flex container. It is new in css3. |
| inline-table | It displays an element as an inline-level table. |
| list-Item | It makes the element behave like a <li> element. |
| table | It makes the element behave like a <table> element. |
| table-caption | It makes the element behave like a <caption> element. |
| table-column-group | It makes the element behave like a <colgroup> element. |
| table-header-group | It makes the element behave like a <thead> element. |
| table-footer-group | It makes the element behave like a <tfoot> element. |
| table-row-group | It makes the element behave like a <tbody> element. |
| table-cell | It makes the element behave like a <td> element. |
| table-row | It makes the element behave like a <tr> element. |
| table-column | It makes the element behave like a <col> element. |

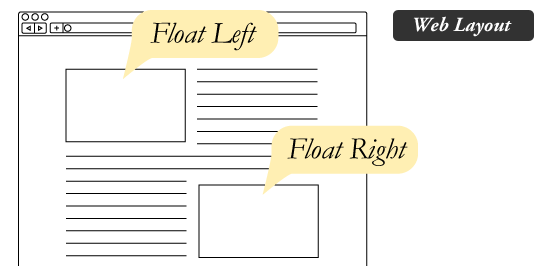
**CSS Float**

The **CSS float property** is *a positioning property*. It is used *to push an element to the left or right*, allowing other element to wrap around it. It is generally used with images and layouts.

To understand its purpose and origin, let's take a look to its print display. In the print display, image is set into the page such that text wraps around it as needed.



Its web layout is also just similar to print layout.



**How it works**

Elements are floated only horizontally. So it is possible only to float elements left or right, not up or down.

1. A floated element may be moved as far to the left or the right as possible. Simply, it means that a floated element can display at extreme left or extreme right.
2. The elements after the floating element will flow around it.
3. The elements before the floating element will not be affected.
4. If the image floated to the right, the texts flow around it, to the left and if the image floated to the left, the text flows around it, to the right.

**CSS Float Properties**

|  |  |  |
| --- | --- | --- |
| **Property** | **Description** | **Values** |
| clear | The clear property is used to avoid elements after the floating elements which flow around it. | left, right, both, none, inherit |
| float | It specifies whether the box should float or not. | left, right, none, inherit |

**CSS Float Property Values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| none | It specifies that the element is not floated, and will be displayed just where it occurs in the text. this is a default value. |
| left | It is used to float the element to the left. |
| right | It is used to float the element to the right. |
| initial | It sets the property to its initial value. |
| inherit | It is used to inherit this property from its parent element. |

**CSS Float Property Example**

Let's see a simple example to understand the CSS float property.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

img {

    float: right;

}

**</style>**

**</head>**

**<body>**

**<p>**The following paragraph contains an image with style

**<b>**float:right**</b>**. The result is that the image will float to the right in the paragraph.**</p>**

**<img** src="good-morning.jpg" alt="Good Morning Friends"**/>**

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

This is some text. This is some text. This is some text.

**</p>**

**</body>**

**</html>**

**Output:**

The following paragraph contains an image with style **float:right**. The result is that the image will float to the right in the paragraph.

This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

# CSS Font

CSS Font property is used to control the look of texts. By the use of CSS font property you can change the text size, color, style and more. You have already studied how to make text bold or underlined. Here, you will also know how to resize your font using percentage.

**These are some important font attributes:**

1. **CSS Font color**: This property is used to change the color of the text. (standalone attribute)
2. **CSS Font family**: This property is used to change the face of the font.
3. **CSS Font size**: This property is used to increase or decrease the size of the font.
4. **CSS Font style**: This property is used to make the font bold, italic or oblique.
5. **CSS Font variant**: This property creates a small-caps effect.
6. **CSS Font weight**: This property is used to increase or decrease the boldness and lightness of the font.

## 1) CSS Font Color

CSS font color is a standalone attribute in CSS although it seems that it is a part of CSS fonts. It is used to change the color of the text.

There are three different formats to define a color:

* By a color name
* By hexadecimal value
* By RGB

In the above example, we have defined all these formats.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    font-size: 100%;

}

h1 { color: red; }

h2 { color: #9000A1; }

p { color:rgb(0, 220, 98); }

}

**</style>**

**</head>**

**<body>**

**<h1>**This is heading 1**</h1>**

**<h2>**This is heading 2**</h2>**

**<p>**This is a paragraph.**</p>**

**</body>**

**</html>**

**Output:**

## This is heading 1

### This is heading 2

This is a paragraph.

## 2) CSS Font Family

CSS font family can be divided in two types:

* Generic family: It includes Serif, Sans-serif, and Monospace.
* Font family: It specifies the font family name like Arial, New Times Roman etc.

**Serif**: Serif fonts include small lines at the end of characters. Example of serif: Times new roman, Georgia etc.

**Sans-serif**: A sans-serif font doesn't include the small lines at the end of characters. Example of Sans-serif: Arial, Verdana etc.



<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

font-size: 100%;

}

h1 { font-family: sans-serif; }

h2 { font-family: serif; }

p { font-family: monospace; }

}

**</style>**

**</head>**

**<body>**

**<h1>**This heading is shown in sans-serif.**</h1>**

**<h2>**This heading is shown in serif.**</h2>**

**<p>**This paragraph is written in monospace.**</p>**

**</body>**

**</html>**

**Output:**

## This heading is shown in sans-serif.

### This heading is shown in serif.

This paragraph is written in monospace.

## 3) CSS Font Size

CSS font size property is used to change the size of the font.

These are the possible values that can be used to set the font size:

|  |  |
| --- | --- |
| **Font Size Value** | **Description** |
| xx-small | used to display the extremely small text size. |
| x-small | used to display the extra small text size. |
| Small | used to display small text size. |
| Medium | used to display medium text size. |
| Large | used to display large text size. |
| x-large | used to display extra large text size. |
| xx-large | used to display extremely large text size. |
| Smaller | used to display comparatively smaller text size. |
| Larger | used to display comparatively larger text size. |
| size in pixels or % | used to set value in percentage or in pixels. |

**<html>**

**<head>**

**<title>**Practice CSS font-size property**</title>**

**</head>**

**<body>**

**<p** style="font-size:xx-small;"**>**  This font size is extremely small.**</p>**

**<p** style="font-size:x-small;"**>**  This font size is extra small**</p>**

**<p** style="font-size:small;"**>**  This font size is small**</p>**

**<p** style="font-size:medium;"**>**  This font size is medium. **</p>**

**<p** style="font-size:large;"**>**  This font size is large. **</p>**

**<p** style="font-size:x-large;"**>**  This font size is extra large. **</p>**

**<p** style="font-size:xx-large;"**>** This font size is extremely large. **</p>**

**<p** style="font-size:smaller;"**>**  This font size is smaller. **</p>**

**<p** style="font-size:larger;"**>**  This font size is larger. **</p>**

**<p** style="font-size:200%;"**>**  This font size is set on 200%. **</p>**

**<p** style="font-size:20px;"**>**  This font size is 20 pixels.  **</p>**

**</body>**

**</html>**

**Output:**

This font size is extremely small.

This font size is extra small

This font size is small

This font size is medium.

This font size is large.

This font size is extra large.

This font size is extremely large.

This font size is smaller.

This font size is larger.

This font size is set on 200%.

This font size is 20 pixels.

## 4) CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

font-size: 100%;

}

h2 { font-style: italic; }

h3 { font-style: oblique; }

h4 { font-style: normal; }

}

**</style>**

**</head>**

**<body>**

**<h2>**This heading is shown in italic font.**</h2>**

**<h3>**This heading is shown in oblique font.**</h3>**

**<h4>**This heading is shown in normal font.**</h4>**

**</body>**

**</html>**

**Output:**

## *This heading is shown in italic font.*

### *This heading is shown in oblique font.*

#### This heading is shown in normal font.

## 5) CSS Font Variant

CSS font variant property specifies how to set font variant of an element. It may be normal and small-caps.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p { font-variant: small-caps; }

h3 { font-variant: normal; }

**</style>**

**</head>**

**<body>**

**<h3>**This heading is shown in normal font.**</h3>**

**<p>**This paragraph is shown in small font.**</p>**

**</body>**

**</html>**

**Output:**

### This heading is shown in normal font.

This paragraph is shown in small font.

## 6) CSS Font Weight

CSS font weight property defines the weight of the font and specify that how bold a font is. The possible values of font weight may be normal, bold, bolder, lighter or number (100, 200..... upto 900).

<!DOCTYPE html**>**

**<html>**

**<body>**

**<p** style="font-weight:bold;"**>**This font is bold.**</p>**

**<p** style="font-weight:bolder;"**>**This font is bolder.**</p>**

**<p** style="font-weight:lighter;"**>**This font is lighter.**</p>**

**<p** style="font-weight:100;"**>**This font is 100 weight.**</p>**

**<p** style="font-weight:200;"**>**This font is 200 weight.**</p>**

**<p** style="font-weight:300;"**>**This font is 300 weight.**</p>**

**<p** style="font-weight:400;"**>**This font is 400 weight.**</p>**

**<p** style="font-weight:500;"**>**This font is 500 weight.**</p>**

**<p** style="font-weight:600;"**>**This font is 600 weight.**</p>**

**<p** style="font-weight:700;"**>**This font is 700 weight.**</p>**

**<p** style="font-weight:800;"**>**This font is 800 weight.**</p>**

**<p** style="font-weight:900;"**>**This font is 900 weight.**</p>**

**</body>**

**</html>**

**Output:**

**This font is bold.**

**This font is bolder.**

This font is lighter.

This font is 100 weight.

This font is 200 weight.

This font is 300 weight.

This font is 400 weight.

This font is 500 weight.

**This font is 600 weight.**

**This font is 700 weight.**

**This font is 800 weight.**

**This font is 900 weight.**

**CSS Line Height**

The **CSS line height property** is used *to define the minimal height of line boxes within the element*. It sets the differences between two lines of your content.

It defines the amount of space above and below inline elements. It allows you to set the height of a line of independently from the font size.

**CSS line-height values**

There are some property values which are used with CSS line-height property.

|  |  |
| --- | --- |
| **value** | **description** |
| normal | This is a default value. it specifies a normal line height. |
| number | It specifies a number that is multiplied with the current font size to set the line height. |
| length | It is used to set the line height in px, pt,cm,etc. |
| % | It specifies the line height in percent of the current font. |
| initial | It sets this property to its default value. |
| inherit | It inherits this property from its parent element. |

**CSS line-height example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h3.small {

    line-height: 70%;

}

h3.big {

    line-height: 200%;

}

**</style>**

**</head>**

**<body>**

**<h3>**

This is a heading with a standard line-height.**<br>**

This is a heading with a standard line-height.**<br>**

The default line height in most browsers is about 110% to 120%.**<br>**

**</h3>**

**<h3** class="small"**>**

This is a heading with a smaller line-height.**<br>**

This is a heading with a smaller line-height.**<br>**

This is a heading with a smaller line-height.**<br>**

This is a heading with a smaller line-height.**<br>**

**</h3>**

**<h3** class="big"**>**

This is a heading with a bigger line-height.**<br>**

This is a heading with a bigger line-height.**<br>**

This is a heading with a bigger line-height.**<br>**

This is a heading with a bigger line-height.**<br>**

**</h3>**

**</body>**

**</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 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. |

#### Note: You can also use negative values to overlap content.

## CSS margin Example

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.ex {

    margin-top: 50px;

    margin-bottom: 50px;

    margin-right: 100px;

    margin-left: 100px;

}

**</style>**

**</head>**

**<body>**

**<p>**This paragraph is not displayed with specified margin. **</p>**

**<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**

**</body>**

**</html>**

**Output:**

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

## 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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.ex {

    margin: 50px 100px 150px 200px;

}

**</style>**

**</head>**

**<body>**

**<p>**This paragraph is not displayed with specified margin. **</p>**

**<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**

**</body>**

**</html>**

Output:

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

## 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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.ex {

    margin: 50px 100px 150px;

}

**</style>**

**</head>**

**<body>**

**<p>**This paragraph is not displayed with specified margin. **</p>**

**<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**

**</body>**

**</html>**

**Output:**

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

## 3) margin: 50px 100px;

It identifies that:

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

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.ex {

    margin: 50px 100px;

}

**</style>**

**</head>**

**<body>**

**<p>**This paragraph is not displayed with specified margin. **</p>**

**<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**

**</body>**

**</html>**

**Output:**

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

## 4) margin: 50px;

It identifies that:

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

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.ex {

    margin: 50px;

}

**</style>**

**</head>**

**<body>**

**<p>**This paragraph is not displayed with specified margin. **</p>**

**<p** class="ex"**>**This paragraph is displayed with specified margin.**</p>**

**</body>**

**</html>**

**Output:**

This paragraph is not displayed with specified margin.

This paragraph is displayed with specified margin.

# CSS Opacity

The CSS opacity property is used to specify the transparency of an element. In simple word, you can say that it specifies the clarity of the image.

In technical terms, Opacity is defined as degree in which light is allowed to travel through an object.

## How to apply CSS opacity setting

Opacity setting is applied uniformly across the entire object and the opacity value is defined in term of digital value less than 1. The lesser opacity value displays the greater opacity. Opacity is not inherited.

## CSS Opacity Example: transparent image

Let's see a simple CSS opacity example of image transparency.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

img.trans {

    opacity: 0.4;

    filter: alpha(opacity=40); /\* For IE8 and earlier \*/

}

**</style>**

**</head>**

**<body>**

**<p>**Normal Image**</p>**

**<img** src="rose.jpg" alt="normal rose"**>**

**<p>**Transparent Image**</p>**

**<img** class="trans" src="rose.jpg" alt="transparent rose"**>**

**</body>**

**</html>**

**Output:**

Normal Image Transparent Image



#### Note 1: Chrome, Firefox, Opera, Safari, and IE9 use the opacity property for transparency. The opacity value ranges from 0.1 to 1.0. Lower value produces the greater opacity.

#### Note 2: The older versions of IE use filter: alpha(opacity=x). Here x value varies from 0 to 100. Lower value produces the greater opacity.

**CSS Overflow**

The **CSS overflow property** *specifies how to handle the content when it overflows* its block level container.

We know that every single element on a page is a rectangular box and the size, positioning and behavior of these boxes are controlled via CSS.

Let's take an example: If you don't set the height of the box, it will grow as large as the content. But if you set a specific height or width of the box and the content inside cannot fit then what will happen. The CSS overflow property is used to overcome this problem. It specifies whether to clip content, render scroll bars, or just display content.

**CSS Overflow property values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| visible | It specifies that overflow is not clipped. it renders outside the element's box.this is a default value. |
| hidden | It specifies that the overflow is clipped, and rest of the content will be invisible. |
| scroll | It specifies that the overflow is clipped, and a scroll bar is used to see the rest of the content. |
| auto | It specifies that if overflow is clipped, a scroll bar is needed to see the rest of the content. |
| inherit | It inherits the property from its parent element. |
| initial | It is used to set the property to its initial value. |

**CSS Overflow Example**

Let's see a simple CSS overflow property.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

div.scroll {

    background-color: #00ffff;

    width: 100px;

    height: 100px;

    overflow: scroll;

}

div.hidden {

    background-color: #00FF00;

    width: 100px;

    height: 170px;

    overflow: hidden;

}

**</style>**

**</head>**

**<body>**

**<p>**The overflow property specifies what to do if the content of an element exceeds thesize of the element's box.**</p>**

**<p>**overflow:scroll**</p>**

**<div** class="scroll"**>**You can use the overflow property when you want to have better control of the layout. The default value is visible.**</div>**

**<p>**overflow:hidden**</p>**

**<div** class="hidden"**>**You can use the overflow property when you want  to have better control of the layoutThe default value is visible.**</div>**

**</body>**

**</html>**

**Output:**

The overflow property specifies what to do if the content of an element exceeds the size of the element's box.

overflow:scroll

You can use the overflow property when you want to have better control of the layout. The default value is visible.

overflow:hidden

You can use the overflow property when you want to have better control of the layout. The default value is

**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**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    background-color: pink;

}

p.padding {

    padding-top: 50px;

    padding-right: 100px;

    padding-bottom: 150px;

    padding-left: 200px;

}

**</style>**

**</head>**

**<body>**

**<p>**This is a paragraph with no specified padding.**</p>**

**<p** class="padding"**>**This is a paragraph with specified paddings.**</p>**

**</body>**

**</html>**

**Output:**

This is a paragraph with no specified padding.

This is a paragraph with specified paddings.

**CSS Position**

The **CSS position property** is used *to set position for an element*. it is also used to place an element behind another and also useful for scripted animation effect.

You can position an element using the top, bottom, left and right properties. These properties can be used only after position property is set first. A position element's computed position property is relative, absolute, fixed or sticky.

Let's have a look at following CSS positioning:

1. CSS Static Positioning
2. CSS Fixed Positioning
3. CSS Relative Positioning
4. CSS Absolute Positioning

**1) CSS Static Positioning**

This is a by default position for HTML elements. It always positions an element according to the normal flow of the page. It is not affected by the top, bottom, left and right properties.

**2) CSS Fixed Positioning**

The fixed positioning property helps to put the text fixed on the browser. This fixed test is positioned relative to the browser window, and doesn't move even you scroll the window.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.pos\_fixed {

     position: fixed;

     top: 50px;

     right: 5px;

     color: blue;

}

**</style>**

**</head>**

**<body>**

**<p>**Some text...**</p>**

**<p>**Some text...**</p>**

**<p>**Some text...**</p>**

**<p>**........**</p>**

**<p>**.... ...**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........ **</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**........**</p>**

**<p>**Some text...**</p>**

**<p>**Some text...**</p>**

**<p>**Some text...**</p>**

**<p** class="pos\_fixed"**>**This is the fix positioned text.**</p>**

**</body>**

**</html>**

**Output:**

Some text... Some text... Some text... ........ .... ... ........ ........ ........ ........ ........ ........ ........ ........ ........ ........ ........ Some text... Some text... Some text...

**3) CSS Relative Positioning**

The relative positioning property is used to set the element relative to its normal position.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h2.pos\_left {

    position: relative;

    left: -30px;

}

h2.pos\_right {

    position: relative;

    left: 30px;

}

**</style>**

**</head>**

**<body>**

**<h2>**This is a heading with no position**</h2>**

**<h2** class="pos\_left"**>**This heading is positioned left according to its normal position**</h2>**

**<h2** class="pos\_right"**>**This heading is positioned right according to its normal position**</h2>**

**<p>**The style "left:-30px" subtracts 30 pixels from the element's original

left position. **</p>**

**<p>**The style "left:30px" adds 30 pixels to the element's original left position.**</p>**

**</body>**

**</html>**

Output:

## This is a heading with no position

## This heading is positioned left according to its normal position

## This heading is positioned right according to its normal position

The style "left:-30px" subtracts 30 pixels from the element's original left position.

The style "left:30px" adds 30 pixels to the element's original left position.

**4) CSS Absolute Positioning**

The absolute positioning is used to position an element relative to the first parent element that has a position other than static. If no such element is found, the containing block is HTML.

With the absolute positioning, you can place an element anywhere on a page.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h2 {

    position: absolute;

    left: 150px;

    top: 250px;

}

**</style>**

**</head>**

**<body>**

**<h2>**This heading has an absolute position**</h2>**

**<p>** The heading below is placed 150px from the left and 250px from the top of the page.**</p>**

**</body>**

**</html>**

**Output:**

The heading below is placed 150px from the left and 250px from the top of the page.

## This heading has an absolute position

**All CSS Position Properties**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **property** | **description** | **values** |
| 1) | bottom | It is used to set the bottom margin edge for a positioned box. | auto, length, %, inherit |
| 2) | clip | It is used to clip an absolutely positioned element. | shape, auto, inherit |
| 3) | cursor | It is used to specify the type of cursors to be displayed. | url, auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait, help |
| 4) | left | It sets a left margin edge for a positioned box. | auto, length, %, inherit |
| 5) | overflow | This property is used to define what happens if content overflow an element's box. | auto, hidden, scroll, visible, inherit |
| 6) | position | It is used to specify the type of positioning for an element. | absolute, fixed, relative, static, inherit |
| 7) | right | It is used to set a right margin edge for a positioned box. | auto, length, %, inherit |
| 8) | top | It is used to set a top margin edge for a positioned box. | auto, length, %, inherit |
| 9) | z-index | It is used to set stack order of an element. | number, auto, inherit |

**CSS Vertical Align**

The CSS vertical align property is used to define the vertical alignment of an inline or table-cell box. It is the one of the self-explanatory property of CSS. It is not very easy property for beginners.

**What it does**

1. It is applied to inline or inline-block elements.
2. It affects the alignment of the element, not its content. (except table cells)
3. When it applied to the table cells, it affect the cell contents, not the cell itself.

**CSS Vertical Align Values**

|  |  |
| --- | --- |
| **value** | **description** |
| baseline | It aligns the baseline of element with the baseline of parent element. This is a default value. |
| length | It is used to increase or decrease length of the element by the specified length. negative values are also allowed. |
| % | It is used to increase or decrease the element in a percent of the "line-height" property. negative values are allowed. |
| sub | It aligns the element as if it was subscript. |
| super | It aligns the element as if it was superscript. |
| top | It aligns the top of the element with the top of the tallest element on the line. |
| bottom | It aligns the bottom of the element with the lowest element on the line. |
| text-top | the top of the element is aligned with the top of the parent element's font. |
| middle | the element is placed in the middle of the parent element. |
| text-bottom | the bottom of the element is aligned with the bottom of the parent element's font. |
| initial | It sets this property to Its default value. |
| inherit | inherits this property from Its parent element. |

**CSS Vertical Align Example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

img.top {

    vertical-align: text-top;

}

img.bottom {

    vertical-align: text-bottom;

}

**</style>**

**</head>**

**<body>**

**<p><img** src="good-morning.jpg"alt="Good Morning Friends"**/>** This is an image with a default alignment.**</p>**

**<p><img** src="good-morning.jpg" class="top" alt="Good Morning Friends"**/>**This is an image with a text-top alignment. **</p>**

**<p><img** src="good-morning.jpg"class="bottom" alt="Good Morning Friends"**/>**This is an image with a text-bottom  alignment. **</p>**

**</body>**

**</html>**

**Output:**

 This is an image with a default alignment.

 This is an image with a text-top alignment.

 This is an image with a text-bottom alignment.

**CSS White Space**

The **CSS white space property** is used to specify how to display the content within an element. It is used *to handle the white spaces inside an element*.

**CSS White Space values**

There are many white space values that can be used to display the content inside an element.

|  |  |
| --- | --- |
| **Value** | **Description** |
| normal | This is a default value. in this value, text is wrapped when necessary. sequences of white space will collapse into a single whitespace. |
| nowrap | Sequences of white space will collapse into a single whitespace. in this value, text will never wrap to the next line and only break when <br> tag is used. |
| pre | Whitespace is preserved by the browser. it is act like html <pre> tag. text will only wrap on line breaks. |
| pre-line | Sequences of white space will collapse into a single whitespace. texts are wrapped when necessary, and on line break. |
| pre-wrap | Whitespace is preserved by the browser. texts are wrapped when necessary, and on line break. |
| initial | It sets this property to its default value. |
| inherit | It inherits this property from its parent element. |

**CSS White Space Example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p {

    white-space: nowrap;

}

**</style>**

**</head>**

**<body>**

**<p>**

Write some text..Write some text..Write some text..

Write some text..Write some text..Write some text..

Write some text..Write some text..Write some text..

Write some text..Write some text..Write some text..

Write some text..Write some text..Write some text..

**</p>**

**</body>**

**</html>**

**Output:**

Write some text..Write some text..Write some text.. Write some text..Write some text..Write some text.. Write some text..Write some text..Write some text.. Write some text..Write some text..Write some text.. Write some text..Write some text..Write some text..

# CSS Width

The **CSS width property** is used to set the width of the content area of an element.

It does not include padding borders or margins. It sets width of the area inside the padding, border, and margin of the element.

## CSS width values

|  |  |
| --- | --- |
| **Value** | **Description** |
| auto | It is a default value. it is used to calculate the width. |
| length | It is used to define the width in px, cm etc. |
| % | It defines the width of the containing block in %. |
| initial | It is used to set the property to its default value. |
| inherit | It is used to inherit the property from its parent element. |

## CSS Width Example: width in px

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

img.normal {

    width: auto;

}

img.big {

    width: 150px;

}

p.ex {

    height: 150px;

    width: 150px;

}

**</style>**

**</head>**

**<body>**

**<img** class="normal" src="good-morning.jpg" width="95" height="84"**><br>**

**<img** class="big" src="good-morning.jpg" width="95" height="84"**>**

**<p** class="ex"**>**The height and width of this paragraph is 150px.**</p>**

**<p>**This is a paragraph.**</p>**

**</body>**

**</html>**

Output:





The height and width of this paragraph is 150px.

This is a paragraph.

## CSS Width Example: width in %

The percent width is a measurement unit for the containing block. It is great for images.

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

img.normal {

    width: auto;

}

img.big {

    width: 50%;

}

img.small {

    width: 10%;

}

**</style>**

**</head>**

**<body>**

**<img** class="normal" src="good-morning.jpg" width="95" height="84"**> <br>**

**<img** class="big" src="good-morning.jpg" width="95"height="84"**> <br>**

**<img** class="small" src="good-morning.jpg" width="95" height="84"**>**

**</body>**

**</html>**

**Output:**







#### Note: You can also use the "min-width" and "max-width" property to control the size of image.

**CSS Word Wrap**

**CSS word wrap property** is used *to break the long words and wrap onto the next line*. This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box.

**CSS Word Wrap Values**

|  |  |
| --- | --- |
| **Value** | **Description** |
| normal | This property is used to break words only at allowed break points. |
| break-word | It is used to break unbreakable words. |
| initial | It is used to set this property to its default value. |
| inherit | It inherits this property from its parent element. |

**CSS Word Wrap Example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

p.test {

    width: 11em;

    background-color: #00ffff;

    border: 1px solid #000000;

    padding:10px;

    word-wrap: break-word;

}

**</style>**

**</head>**

**<body>**

**<p** class="test"**>** In this paragraph, there is a very long word:

 iamsooooooooooooooooooooooooooooooolongggggggggggggggg.The long word will break and wrap to the next line.**</p>**

**</body>**

**</html>**

**CSS Outline**

CSS outline is just like CSS border property. It facilitates you to draw an extra border around an element to get visual attention.

It is as easy as to apply as a border.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style** type="text/css"**>**

.box {

        background-color: #eee;

        outline: 3px solid red;

        border: 3px solid lightgreen;

        padding: 5px 10px;

}

**</style>**

**<body>**

**<div** class="box"**>**Welcome to Zerozilla**</div>**

**</body>**

**</html>**

**Difference between Border and Outline**

At first glance, border and outline look similar, but there are some very important differences between them:

* It is not possible to apply a different outline width, style and color for the four sides of an element while in border; you can apply the provided value for all four sides of an element.
* The border is a part of element's dimension while the outline is not the part of element's dimension. Means, it doesn't matter how thick an outline you apply to the element, the dimensions of it won't change.

The outline property is a shorthand property. It can be divided into outline-width, outline-style and outline-color properties. It facilitates you to use any of the property alone, if you need it.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style** type="text/css"**>**

.box {

        background-color: #eee;

        border: 3px solid Lightgreen;

        padding: 5px 10px;

        outline-width: 3px;

        outline-style: solid;

        outline-color: red;

}

**</style>**

**<body>**

**<div** class="box"**>**Welcome to Zerozilla**</div>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=css-outline2)

In the above example, you can see the three outline properties:

**Outline-width:**It is similar to margin and padding. It can be either an absolute value or a relative value or one of the predefined outline width values i.e. thin, medium or thick. It is preferred to use either an absolute value or a relative value because different browsers interpret differently while using predefined outline width values like thin, medium or thick.

**Outline-color:**It specifies the color that you use for the outline. It supports all the colors available in HTML and CSS.

**Outline-style:**In the above example, we have used only solid outline style while there are a lot of outline style i.e. hidden, dotted, dashed, solid, double, groove, ridge, inset and outset.

Let's take an example to demonstrate the usage of different outline-styles.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style** type="text/css"**>**

.box {

        outline-color: red;

        outline-width: 4px;

        margin: 10px;

        float: left;

        border: 2px solid lightgreen;

        padding: 5px 10px;

}

**</style>**

**<body>**

**<div** class="box" style="outline-style: dashed;"**>** This is dashed outline.**</div>**

**<div** class="box" style="outline-style: dotted;"**>**This is dotted outline. **</div>**

**<div** class="box" style="outline-style: double;"**>**This is double outline. **</div>**

**<div** class="box" style="outline-style: groove;"**>**This is groove outline. **</div>**

**<div** class="box" style="outline-style: inset;"**>**This is inset outline. **</div>**

**<div** class="box" style="outline-style: outset;"**>**This is outset outline. **</div>**

**<div** class="box" style="outline-style: ridge;"**>**This is ridge outline. **</div>**

**<div** class="box" style="outline-style: solid;"**>**This is solid outline. **</div>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=css-outline3)

**Outline offset**

The outline offset is used to create a distance between outline and border.

It takes a CSS length unit and the empty space between the border and the outline will be transparent and then it takes the background color of the parent element. So you can see a visible difference between outline and border.

Let's take an example to see the difference between outline and border.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<style** type="text/css"**>**

.box {

        background-color: #eee;

        outline: 3px solid red;

        outline-offset: 6px;

        border: 3px solid Lightgreen;

        padding: 5px 10px;

}

**</style>**

**<body>**

**<div** class="box"**>**Welcome to Zerozilla**</div>**

**</body>**

**</html>**

**CSS Visibility**

The CSS visibility property is used to specify whether an element is visible or not.

**Note:** An invisible element also take up the space on the page. By using display property you can create invisible elements that don't take up space.

**Syntax:**

visibility: visible|hidden|collapse|initial|inherit;

**CSS Visibility Parameters**

**visible:**It is the by default value. It specifies that the element is visible.

**hidden:**It specifies that the element is invisible (but still takes up space).

**collapse:**It is used only for table elements. It is used to remove a row or column, but it does not affect the table layout.

The space taken up by the row or column will be available for other content.

If collapse is used on other elements, it renders as "hidden"

**initial:**It is used to set this property to its default value.

**inherit:**It is used to inherit this property from its parent element.

**CSS Visibility Example**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

h1.visible {

    visibility: visible

}

h1.hidden {

    visibility: hidden

}

**</style>**

**</head>**

**<body>**

**<h1** class="visible"**>**I am visible**</h1>**

**<h1** class="hidden"**>**I am invisible**</h1>**

**<p><strong>**Note:**</strong>** An invisible element also take up the space on the page.By using display property you can create invisible elements that don't  take space. **</p>**

**</body>**

**</html>**

**JavaScript Syntax:**

object.style.visibility="hidden"

**See the JavaScript example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

#myDIV {

    margin: auto;

    width: 400px;

    height: 200px;

    background-color: lightpink;

    border: 1px solid black;

}

**</style>**

**</head>**

**<body>**

**<p>**Click the "Try it" button to set the visibility property and hide the div element.**</p>**

**<button** onclick="myFunction()"**>**Try it**</button>**

**<div** id="myDIV"**>**This is my DIV element.**</div>**

**<p><strong>**Note:**</strong>** An invisible element also take up the space on the page. **</p>**

**<script>**

function myFunction() {

    document.getElementById("myDIV").style.visibility = "hidden";

}

**</script>**

**</body>**

**</html>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=css-visibility2)

# CSS Counters

CSS counters are similar to variables. These are maintained by CSS and those values can be incremented by CSS rules to track how many times they are used.

CSS counters facilitate simple CSS based incrementing and display of a number for generated content.

## CSS Counter Properties

Following is a list of properties that are used with CSS counter:

* **counter-reset:** It is used to create or reset a counter.
* **counter-increment:** It is used to increment the counter value.
* **content:** It is used to insert generated content.
* **counter() or counters() function:** It is used to add the value of a counter to an element.

#### Note:Before using a CSS counter, it must be created with counter-reset.

## CSS Counter Example

Let's take an example to create a counter for a page and increment the counter value for each next element.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    counter-reset: section;

}

h2::before {

    counter-increment: section;

    content: "Section " counter(section) ": ";

}

**</style>**

**</head>**

**<body>**

**<h1>**Example of CSS Counters:**</h1>**

**<h2>**Java Tutorial**</h2>**

**<h2>**HTML Tutorial**</h2>**

**<h2>**CSS Tutorial**</h2>**

**<h2>**Oracle Tutorial**</h2>**

**<p><strong>**Note:**</strong>** IE8 supports these properties only if a !DOCTYPE is specified.**</p>**

**</body>**

**</html>**

#### Note: In the above example you can see that a counter is created for the page in the body selector and it increments the counter value for each <h2> element and adds "Section <value of the counter>:" to the beginning of each <h2> element.

## CSS Nesting Counters

You can also create counters within the counter. It is called nesting of a counter. Let's take an example to demonstrate nesting counter.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

body {

    counter-reset: section;

}

h1 {

    counter-reset: subsection;

}

h1::before {

    counter-increment: section;

    content: "Section " counter(section) ". ";

}

h2::before {

    counter-increment: subsection;

    content: counter(section) "." counter(subsection) " ";

}

**</style>**

**</head>**

**<body>**

**<h1>**Java tutorials:**</h1>**

**<h2>**Core Java tutorial**</h2>**

**<h2>**Servlet tutorial**</h2>**

**<h2>**JSP tutorial**</h2>**

**<h2>**Spring tutorial**</h2>**

**<h2>**Hibernate tutorial**</h2>**

**<h1>**Web technology tutorials:**</h1>**

**<h2>**HTML tutorial**</h2>**

**<h2>**CSS tutorial**</h2>**

**<h2>**jQuery tutorial**</h2>**

**<h2>**Bootstrap tutorial**</h2>**

**<h1>**Database tutorials:**</h1>**

**<h2>**SQL tutorial**</h2>**

**<h2>**MySQL tutorial**</h2>**

**<h2>**PL/SQL tutorial**</h2>**

**<h2>**Oracle tutorial**</h2>**

**<p><strong>**Note:**</strong>** IE8 supports these properties only if a !DOCTYPE is specified.**</p>**

**</body>**

**</html>**

**Note:**In the above example you can see that a counter is created for the section and another nesting counter named subsection is created within section.

## Different level of nesting counters

You can create outlined lists by using nesting counters. It facilitates you to insert a string between different levels of nested counters.

**See this example:**

<!DOCTYPE html**>**

**<html>**

**<head>**

**<style>**

ol {

    counter-reset: section;

    list-style-type: none;

}

li::before {

    counter-increment: section;

    content: counters(section,".") " ";

}

**</style>**

**</head>**

**<body>**

**<h2>**Different level of Nesting counters**</h2>**

**<ol>**

**<li>**item**</li>**

**<li>**item

**<ol>**

**<li>**item**</li>**

**<li>**item**</li>**

**<li>**item

**<ol>**

**<li>**item**</li>**

**<li>**item**</li>**

**<li>**item**</li>**

**</ol>**

**</li>**

**<li>**item**</li>**

**</ol>**

**</li>**

**<li>**item**</li>**

**<li>**item**</li>**

**</ol>**

**<p><b>**Note:**</b>** IE8 supports these properties only if a !DOCTYPE is specified.**</p>**

**</body>**

**</html>**