# Cascading style sheet: -

**Cascading** refers to the way CSS applies one style on top of another.

**Style Sheets** control the look and feel of web documents.

## Selector types:

### Type selector:

p { color: red; font-size:130%; }

### id selector:

<style>#para1 { text-align: center; color: blue; } </style>

</head>

<body>

<p id="para1">Hello Javatpoint.com</p>

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

</body>

### class selector:

<!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>

### class seletor for specific element:

<!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> <!—only work for paragraph-->

</body> </html>

### class universal selector:

\* { color: green; font-size: 20px; }

### group selector:

h1,p{ color:blue; background-color:red; }

|  |  |
| --- | --- |
| .html | .css |
| <head>  <link rel="stylesheet" type="text/css" href="style.css">  </head> | p{color:blue} |

## Css comments

/\*..comments..\*/

# Css Properties:

## Display:

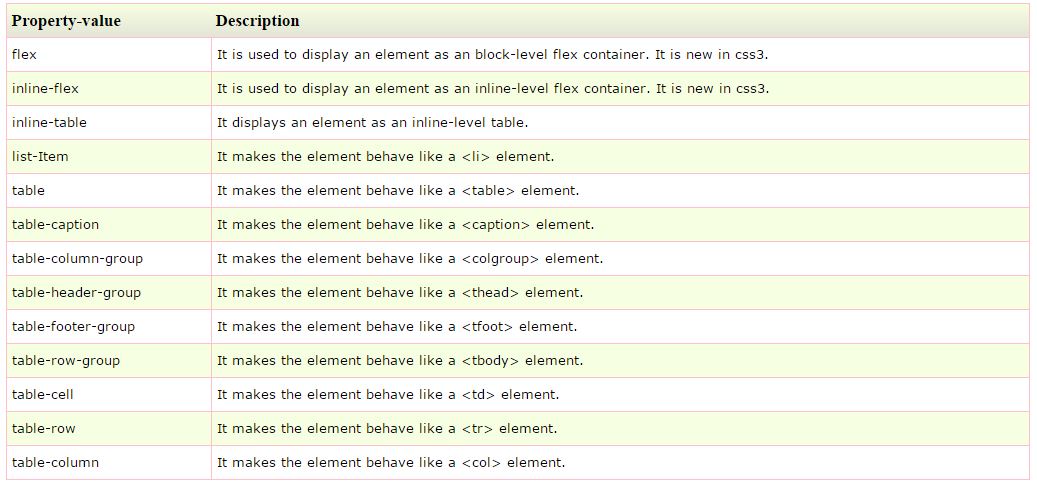
### 1.display: inline; /\*doesn’t consider </br>

### 2.display: inline-block; /\*same as above but can use width

### 3.display: block; /\*opposite to above and make double </br>

### 4.display: run-in; /\* These elements don't produce a specific box by themselves.

### 5.display: none; /\* doesn’t display element, removed from Dom



## Text:

### Font:

#### Font family: font-family:serif;

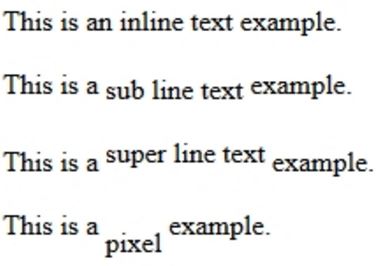
#### Font size: font-size:small; /\*xx-small,x-small, medium, large,x-large,xx-large,smaller,larger,100%\*/

#### Font style: font-style:italic; /\*oblique,normal\*/

#### Font variant: This property creates a small-caps effect. font-variant:small-caps; font-variant:normal;

#### Font weight: font-weight:bold; /\*bolder,lighter,numbers(100,200)\*

### Vertical align

vertical-align: text-top;

vertical-align: text-bottom;

vertical-align: text-middle;

vertical-align:baseline;

vertical-align:sub;

vertical-align:super;

vertica-align: -10px;

### Horizontal align

text-align:left;

text-align:right;

text-align:center;

### Text decoration:

text-decoration: none;

text-decoration: inherit;

text-decoration: overline;

text-decoration: underline;

text-decoration: line-through;

text-decoration: blink;

### Indenting Text:

text-indent: 60px;

### Shadowing text:

text-shadow: 5px 2px 4px grey 0.5em;

In the example above, we created a shadow using the following parameters:

5px – the X-coordinate

2px – the Y-coordinate

4px – the blur radius

grey – the color of the shadow

0.5em- blur effect

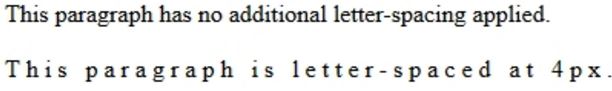
### Text transform:

text-transform: capitalize; /\* each word capitalized.

text-transform: uppercase; /\*all word caps

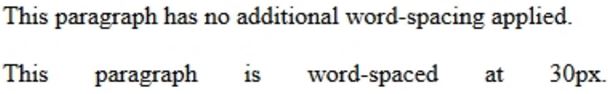
text-transform: lowercase; /\*all word lowercase

### Text spacing:

 letter-spacing: normal;

letter-spacing: 4px;

 letter-spacing: -1.5px;

word-spacing: normal;

word-spacing: 30px;

word-spacing: -5px

### Word wrap

break the long words and wrap onto the next line

word-wrap:normal;

word-wrap:break-word;

word-wrap:intial;

word-wrap:inherit;

### Whitespace:

white-space: normal;

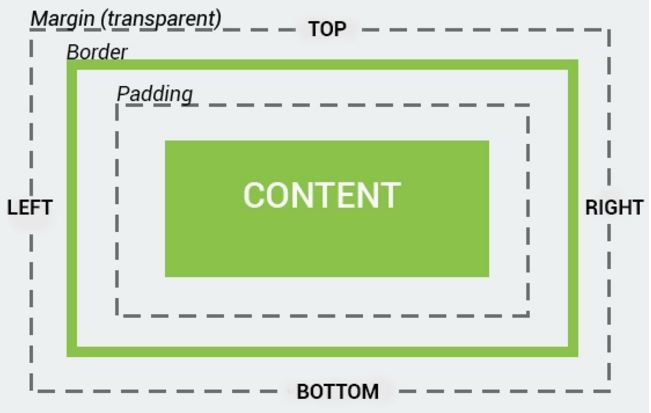
white-space: inherit;

white-space: nowrap; /\*text will continue on the same line until a <br /> tag is encountered.

white-space: pre; /\*text will only wrap on line breaks and white space

white-space: pre-line; /\* text will wrap where there is a break,extra white space ignored

white-space: pre-wrap; /\* text will wrap when necessary, and on line breaks



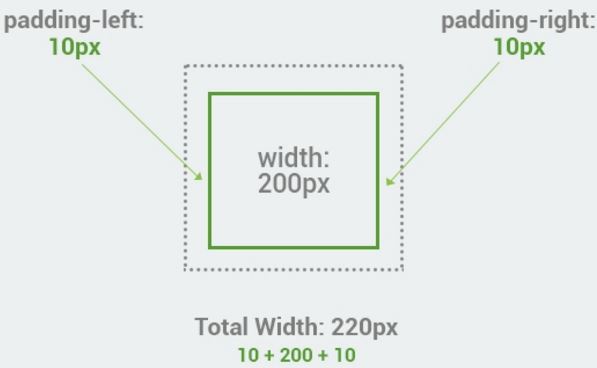
## The CSS Box Model

All HTML elements can be considered as boxes. The CSS box model represents the design and layout of the site. It consists of margins, borders, paddings, actual content.

The properties work in the same order: top, right, bottom, and left.

## Padding:

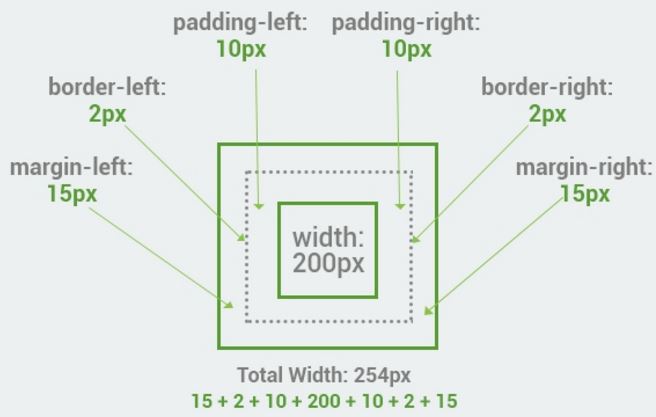
padding-top:50px; padding-bottom:100px; padding-right:20%; padding-left:40%;

text-align:centre;

## Width, Height:

width:auto;

width:100%;

width:150px;

height:100%;

height:150px;

Total element height = height + top padding + bottom padding + top border + bottom border + top margin + bottom margin

## Outline

outline:3px solid red;

outline-color: red;

outline-width: 4px;

outline-offset:3px; /\*distance between border and outline

## Visibility

visibility:hidden;

visibility:visible;

visibility:collapse;

## Counter

similar to variables. These are maintained by CSS and those values can be incremented

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.

|  |  |
| --- | --- |
| body {  counter-reset: section;  }  h2::before {  counter-increment: section;  content: "Section " counter(section) ": ";  }  </style></head>  <body>  <h3>ouput:</h>  <h1>Example of CSS Counters:</h1>  <h2>Java Tutorial</h2>  <h2>HTML Tutorial</h2>  <h2>CSS Tutorial</h2>  <h2>Oracle Tutorial</h2> |  |

### /

## Float:

It is a positioning property. It is used to push an element to the left or right, allowing other element to wrap around it.

float:left; /\*also wer can use right, top, bottom,inherit,none\*/

## Line height:

It is used to define the minimal height of line boxes within the element. It sets the differences between two lines of your content. line-height: 70%;

## Margin

p { background-color: pink; }

p.ex { margin-top: 50px; margin-bottom: 50px; margin-right: 100px; margin-left: 100px; }

margin:100px; /\*all sides\*/

margin:50px 100px; /\*top bottom, left right\*/

margin:50px 100px 150px; /\*top, left right, bottom\*/

margin: 50px 100px 150px 200px; /\*top,right, bottom,left\*/

## Opacity:

opacity:0; /\*100%opacity\*/

opacity:0.5; /\*50% opacity\*/

opacity:1; /\* 0% opacity\*/

## CSS overflow:

overflow:visible; /\* extra text will visible outside of the box

overflow: scroll; /\*scroll bar will present in box

overflow: hidden; /\*no scroll bar, not visible outside box, the box will extend upto text

overflow:auto; /\*scroll bar occur if needed

overflow:inherit; /\*get properties from parents

overflow:intial; /\*get intial assinged values

## Position

CSS Static Positioning- default html elements

CSS Fixed Positioning- position: fixed; top: 50px; right: 5px; color: blue;

CSS Relative Positioning- used to set the element relative to its normal position. position: relative; left: -30px;

CSS Absolute Positioning- position: absolute; left: 150px; top: 250px;

## Mouse cursor:

<span style="cursor:help;">

Do you need help?

</span>

p { cursor:pointer;} /\*default

**CSS3**

CSS3 is completely backwards-compatible with earlier CSS versions.

Some of the most significant new features are:

Border radius - allows us to create rounded corners for elements.

Border images - allows us to specify an image as the border around an element.

Multiple backgrounds - applies multiple backgrounds to elements.

background-image: url(csslogo.png), url(csscode.jpg);

Animations and effects, and much more!

Box Shadow

With the box-shadow property, you can attach one or more shadows to an element by specifying values for color, size, blur, and offset.

box-shadow: 10px 10px #888888;

Gradients

CSS3 gradients allow us to set the background color of the element to a gradient. Two types of gradients are available: Linear and Radial.

Lineaar gradient: background:-moz-linear-gradient(DeepSkyBlue, Black);

Radial: background: radial-gradient(position, shape or size, color-stops);

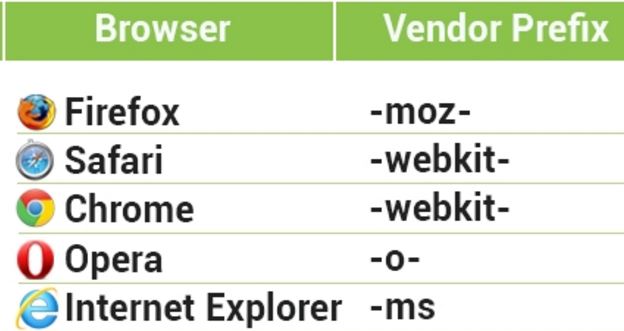
Transforms allow you to rotate, scale, move, and skew elements.

Another popular feature is Transitions which allows you to animate from one CSS property value to another. You can combine it with transforms and animate the element's position, rotation, or scale.

The property attracting the most attention is Animations.

CSS Animations have their own specifications, and they allow you to create keyframes, set duration, easing, and more

CSS Vendor Prefixes

-webkit-border-radius: 24px;

Border radius(rounder corners)

border-radius: 20px;

Working with Pseudo Elements

p::first-line {

color: #589432;

}

::first-line - the first line of the text in a selector

::first-letter - the first letter of the text in a selector

::selection - selects the portion of an element that is selected by a user

::before - inserts some content before an element

::after - inserts some content after an element

CSS3 Transitions

CSS3 transitions allow us to change from one property value to another over a given duration.

transition-property - specifies the property to be transitioned

transition-duration - specifies the duration over which transitions should occur

transition-timing-function - specifies how the pace of the transition changes over its duration

transition-delay - specifies a delay (in seconds) for the transition effect

transition: width 3s;

transition-timing-function: cubic-bezier(0,0,1,1);

Transforms

transform: rotate(10deg);

transform: skew(30deg);

transform-origin: 25% 75%;

transform:translate(100px, 50px);

transform: scale(0.7, 0.7);

transform: rotate(45deg) translate(100px);

Animation:

div {

animation-name: colorchange;

animation-duration: 3s;

animation-timing-function: ease-in;

animation-delay: 1s;

animation-iteration-count: infinite;

animation-direction: reverse;

}

3D tranforms:

div.X {

transform: rotateX(150deg);

}

div.Y {

transform: rotateY(150deg);

}

div.Z {

transform: rotateZ(150deg);

}