# CSS

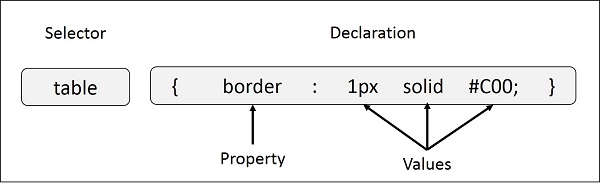
**Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**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

selector { property: value }



table{ border :1px solid #C00; }

## The Type Selectors

h1 {

color: #36CFFF;

}

## The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type

\* {

color: #000000;

}

## The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

ul em {

color: #000000;

}

## The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

.black {

color: #000000;

}

This rule renders the content in black for every element with class attribute set to *black* in our document. You can make it a bit more particular.

h1.black {

color: #000000;

}

This rule renders the content in black for only <h1> elements with class attribute set to *black*.

You can apply more than one class selectors to given element. Consider the following example −

<p class = "center bold">

This para will be styled by the classes *center* and *bold*.

</p>

## The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

#black {

color: #000000; }

This rule renders the content in black for every element with *id* attribute set to *black* in our document. You can make it a bit more particular. For example −

h1#black {

color: #000000;

}

This rule renders the content in black for only <h1> elements with *id* attribute set to *black*.

The true power of *id* selectors is when they are used as the foundation for descendant selectors,

#black h2 {

color: #000000;

}

## The Child Selectors

You have seen the descendant selectors. There is one more type of selector, which is very similar to descendants but have different functionality. Consider the following example −

body > p {

color: #000000;

}

This rule will render all the paragraphs in black if they are direct child of <body> element. Other paragraphs put inside other elements like <div> or <td> would not have any effect of this rule.

## The Attribute Selectors

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of *text* −

input[type = "text"] {

color: #000000;

}

The advantage to this method is that the <input type = "submit" /> element is unaffected, and the color applied only to the desired text fields.

There are following rules applied to attribute selector.

* **p[lang]** − Selects all paragraph elements with a *lang* attribute.
* **p[lang="fr"]** − Selects all paragraph elements whose *lang* attribute has a value of exactly "fr".
* **p[lang~="fr"]** − Selects all paragraph elements whose *lang* attribute contains the word "fr".
* **p[lang|="en"]** − Selects all paragraph elements whose *lang* attribute contains values that are exactly "en", or begin with "en-".

## Multiple Style Rules

You may need to define multiple style rules for a single element. You can define these rules to combine multiple properties and corresponding values into a single block as defined in the following example −

h1 {

color: #36C;

font-weight: normal;

letter-spacing: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

## Grouping Selectors

h1, h2, h3 {

color: #36C;

font-weight: normal;

letter-spacing: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

You can combine the various *id* selectors together as shown below −

#content, #footer, #supplement {

position: absolute;

left: 510px;

width: 200px;

}

CSS - Inclusion

## Inline CSS - The *style* Attribute

You can use *style* attribute of any HTML element to define style rules. These rules will be applied to that element only. Here is the generic syntax −

<element style = "...style rules....">

|  |  |  |
| --- | --- | --- |
| **ttribute** | **Value** | **Description** |
| style | style rules | The value of *style* attribute is a combination of style declarations separated by semicolon (;). |

<html>

<head>

</head>

<body>

<h1 style = "color:#36C;">

This is inline CSS

</h1>

</body>

</html>

## External CSS - The <link> Element

The <link> element can be used to include an external stylesheet file in your HTML document.

An external style sheet is a separate text file with **.css** extension. You define all the Style rules within this text file and then you can include this file in any HTML document using <link> element.

<head>

<link type = "text/css" href = "..." media = "..." />

</head>

Attributes associated with <style> elements are −

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Value** | **Description** |
| Type | text css | Specifies the style sheet language as a content-type (MIME type). This attribute is required. |
| Href | URL | Specifies the style sheet file having Style rules. This attribute is a required. |
|  |  |  |

h1, h2, h3 {

color: #36C;

font-weight: normal;

letter-spacing: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

Now you can include this file *mystyle.css* in any HTML document as follows −

<head>

<link type = "text/css" href = "mystyle.css" media = " all" />

</head>

## Imported CSS - @import Rule

@import is used to import an external stylesheet in a manner similar to the <link> element. Here is the generic syntax of @import rule.

<head>

<@import "URL";

</head>

Here URL is the URL of the style sheet file having style rules. You can use another syntax as well −

<head>

<@import url("URL");

</head>

Following is the example showing you how to import a style sheet file into HTML document −

<head>

@import "mystyle.css";

</head>

## CSS Rules Overriding

* Any inline style sheet takes highest priority. So, it will override any rule defined in <style>...</style> tags or rules defined in any external style sheet file.
* Any rule defined in <style>...</style> tags will override rules defined in any external style sheet file.
* Any rule defined in external style sheet file takes lowest priority, and rules defined in this file will be applied only when above two rules are not applicable.

## CSS Comments

Many times, you may need to put additional comments in your style sheet blocks. So, it is very easy to comment any part in style sheet. You can simple put your comments inside /\*.....this is a comment in style sheet.....\*/.

<!DOCTYPE html>

<html>

<head>

<style>

p {

color: red;

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

text-align: center;

}

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

</style>

</head>

<body>

<p>Hello World!</p>

</body>

</html>

# CSS - Backgrounds

## Set the Background Color

<html>

<head>

</head>

<body>

<p **style = "background-color:yellow;"**>

This text has a yellow background color.

</p>

</body>

</html>

## Set the Background Image

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-color: #cccccc;

}

</style>

</head>

<body>

<h1>Hello World!</h1>

</body>

<html>

Repeat the Background Image

The following example demonstrates how to repeat the background image if an image is small. You can use *no-repeat* value for *background-repeat* property if you don't want to repeat an image, in this case image will display only once.

By default *background-repeat* property will have *repeat* value.

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-repeat: repeat;

}

</style>

</head>

<body>

<p>Tutorials point</p>

</body>

</html>

The following example which demonstrates how to repeat the background image vertically.

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-repeat: repeat-y;

}

</style>

</head>

<body>

<p>Tutorials point</p>

</body>

</html>

The following example demonstrates how to repeat the background image horizontally.

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-repeat: repeat-x;

}

</style>

</head>

<body>

<p>Tutorials point</p>

</body>

</html>

## Set the Background Image Position

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-position:100px;

}

</style>

</head>

<body>

<p>Tutorials point</p>

</body>

</html>

The following example demonstrates how to set the background image position 100 pixels away from the left side and 200 pixels down from the top.

<html>

<head>

<style>

body {

background-image: url("/css/images/css.jpg");

background-position:100px 200px;

}

</style>

</head>

<body>

<p>Tutorials point</p>

</body>

</html>

## Set the Background Attachment

Background attachment determines whether a background image is fixed or scrolls with the rest of the page.

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url('/css/images/css.jpg');

background-repeat: no-repeat;

background-attachment: fixed;

}

</style>

</head>

<body>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

</body>

</html>

The following example demonstrates how to set the scrolling background image.

<!DOCTYPE html>

<html>

<head>

<style>

body {

background-image: url('/css/images/css.jpg');

background-repeat: no-repeat;

background-attachment: fixed;

background-attachment:scroll;

}

</style>

</head>

<body>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

<p>The background-image is fixed. Try to scroll down the page.</p>

</body>

</html>

## Shorthand Property

You can use the *background* property to set all the background properties at once. For example −

<p **style = "background:url(/images/pattern1.gif) repeat fixed;"**>

This parapgraph has fixed repeated background image.

</p>

# CSS - Fonts

## Set the Font Family

<html>

<head>

</head>

<body>

<p **style = "font-family:georgia,garamond,serif;"**>

This text is rendered in either georgia, garamond, or the

default serif font depending on which font you have at your system.

</p>

</body>

</html>

## Set the Font Style

Possible values are *normal, italic and oblique*.

<html>

<head>

</head>

<body>

<p **style = "font-style:italic;"**>

This text will be rendered in italic style

</p>

</body>

</html>

## Set the Font Variant

Possible values are *normal and small-caps*.

<html>

<head>

</head>

<body>

<p **style = "font-variant:small-caps;"**>

This text will be rendered as small caps

</p>

</body>

</html>

## Set the Font Weight

The font-weight property provides the functionality to specify how bold a font is. Possible values could be *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*.

<html>

<head>

</head>

<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:500;"**>

This font is 500 weight.

</p>

</body>

</html>

## Set the Font Size

The font-size property is used to control the size of fonts. Possible values could be *xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, size in pixels or in %*.

<html>

<head>

</head>

<body>

<p **style = "font-size:20px;"**>

This font size is 20 pixels

</p>

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

This font size is small

</p>

<p **style = "font-size:large;"**>

This font size is large

</p>

</body>

</html>

## Set the Font Size Adjust

This property enables you to adjust the x-height to make fonts more legible. Possible value could be any number.

<html>

<head>

</head>

<body>

<p **style = "font-size-adjust:0.61;"**>

This text is using a font-size-adjust value.

</p>

</body>

</html>

## Set the Font Stretch

Possible values could be *normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded*.

<html>

<head>

</head>

<body>

<p **style = "font-stretch:ultra-expanded;"**>

If this doesn't appear to work, it is likely that your computer

doesn't have a <br>condensed or expanded version of the font being used.

</p>

</body>

</html>

## Shorthand Property

<html>

<head>

</head>

<body>

<p **style = "font:italic small-caps bold 15px georgia;"**>

Applying all the properties on the text at once.

</p>

</body>

</html>

# CSS - Text

* The **text-decoration** property is used to underline, overline, and strikethrough text.
* The **text-transform** property is used to capitalize text or convert text to uppercase or lowercase letters.
* The **white-space** property is used to control the flow and formatting of text.
* The **text-shadow** property is used to set the text shadow around a text.

## Set the Text Color

<html>

<head>

</head>

<body>

<p **style = "color:red;"**>

This text will be written in red.

</p>

</body>

</html>

## Set the Text Direction

Possible values are *ltr or rtl*.

<html>

<head>

</head>

<body>

<p **style = "direction:rtl;"**>

This text will be rendered from right to left

</p>

</body>

</html>

## Set the Space between Characters

Possible values are *normal or a number specifying space.*.

<html>

<head>

</head>

<body>

<p **style = "letter-spacing:5px;"**>

This text is having space between letters.

</p>

</body>

</html>

## Set the Space between Words

Possible values are *normal or a number specifying space*.

<html>

<head>

</head>

<body>

<p **style = "word-spacing:5px;"**>

This text is having space between words.

</p>

</body>

</html>

## Set the Text Indent

Possible values are *% or a number specifying indent space*.

<html>

<head>

</head>

<body>

<p **style = "text-indent:1cm;"**>

This text will have first line indented by 1cm and this line will remain at

its actual position this is done by CSS text-indent property.

</p>

</body>

</html>

## Set the Text Alignment

Possible values are *left, right, center, justify*.

<html>

<head>

</head>

<body>

<p **style = "text-align:right;"**>

This will be right aligned.

</p>

<p **style = "text-align:center;"**>

This will be center aligned.

</p>

<p **style = "text-align:left;"**>

This will be left aligned.

</p>

</body>

</html>

## Decorating the Text

Possible values are *none, underline, overline, line-through, blink*.

<html>

<head>

</head>

<body>

<p **style = "text-decoration:underline;"**>

This will be underlined

</p>

<p **style = "text-decoration:line-through;"**>

This will be striked through.

</p>

<p **style = "text-decoration:overline;"**>

This will have a over line.

</p>

<p **style = "text-decoration:blink;"**>

This text will have blinking effect

</p>

</body>

</html>

## Set the Text Cases

Possible values are *none, capitalize, uppercase, lowercase*.

<html>

<head>

</head>

<body>

<p **style = "text-transform:capitalize;"**>

This will be capitalized

</p>

<p **style = "text-transform:uppercase;"**>

This will be in uppercase

</p>

<p **style = "text-transform:lowercase;"**>

This will be in lowercase

</p>

</body>

</html>

## Set the White Space between Text

Possible values are *normal, pre, nowrap*.

<html>

<head>

</head>

<body>

<p **style = "white-space:pre;"**>

This text has a line break and the white-space pre setting

tells the browser to honor it just like the HTML pre tag.

</p>

</body>

</html>

# CSS - Using Images

## The Image Border Property

The *border* property of an image is used to set the width of an image border. This property can have a value in length or in %.

A width of zero pixels means no border.

<html>

<head>

</head>

<body>

<img style = "border:0px;" src = "/css/images/logo.png" />

<br />

<img style = "border:3px dashed red;" src = "/css/images/logo.png" />

</body>

</html>

## The Image Height Property

The *height* property of an image is used to set the height of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

<html>

<head>

</head>

<body>

<img style = "border:1px solid red; height:100px;" src = "/css/images/logo.png" />

<br />

<img style = "border:1px solid red; height:50%;" src = "/css/images/logo.png" />

</body>

</html>

## The Image Width Property

The *width* property of an image is used to set the width of an image. This property can have a value in length or in %. While giving value in %, it applies it in respect of the box in which an image is available.

<html>

<head>

</head>

<body>

<img style = "border:1px solid red; width:150px;" src = "/css/images/logo.png" />

<br />

<img style = "border:1px solid red; width:100%;" src = "/css/images/logo.png" />

</body>

</html>

# CSS - Links

Usually, all these properties are kept in the header part of the HTML document.

Remember a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective. Also, a:active MUST come after a:hover in the CSS definition as follows −

<style type = "text/css">

a:link {color: #000000}

a:visited {color: #006600}

a:hover {color: #FFCC00}

a:active {color: #FF00CC}

</style>

## Set the Color of Links

The following example demonstrates how to set the link color. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:link {color:#000000}

</style>

</head>

<body>

<a href = "">Link</a>

</body>

</html>

## Set the Color of Visited Links

Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:visited {color: #006600}

</style>

</head>

<body>

<a href = ""> link</a>

</body>

</html>

## Change the Color of Links when Mouse is Over

<html>

<head>

<style type = "text/css">

a:hover {color: #FFCC00}

</style>

</head>

<body>

<a href = "">Link</a>

</body>

</html>

## Change the Color of Active Links

<html>

<head>

<style type = "text/css">

a:active {color: #FF00CC}

</style>

</head>

<body>

<a href = "">Link</a>

</body>

</html>

# CSS - Tables

## The border-collapse Property

This property can have two values *collapse* and *separate*.

<html>

<head>

<style type = "text/css">

table.one {border-collapse:collapse;}

table.two {border-collapse:separate;}

td.a {

border-style:dotted;

border-width:3px;

border-color:#000000;

padding: 10px;

}

td.b {

border-style:solid;

border-width:3px;

border-color:#333333;

padding:10px;

}

</style>

</head>

<body>

<table class = "one">

<caption>Collapse Border Example</caption>

<tr><td class = "a"> Cell A Collapse Example</td></tr>

<tr><td class = "b"> Cell B Collapse Example</td></tr>

</table>

<br />

<table class = "two">

<caption>Separate Border Example</caption>

<tr><td class = "a"> Cell A Separate Example</td></tr>

<tr><td class = "b"> Cell B Separate Example</td></tr>

</table>

</body>

</html>

## The border-spacing Property

The border-spacing property specifies the distance that separates adjacent cells'. borders. It can take either one or two values; these should be units of length.

If you provide one value, it will applies to both vertical and horizontal borders. Or you can specify two values, in which case, the first refers to the horizontal spacing and the second to the vertical spacing −

<style type="text/css">

/\* If you provide one value \*/

table.example {border-spacing:10px;}

/\* This is how you can provide two values \*/

table.example {border-spacing:10px; 15px;}

</style>

<html>

<head>

<style type = "text/css">

table.one {

border-collapse:separate;

width:400px;

border-spacing:10px;

}

table.two {

border-collapse:separate;

width:400px;

border-spacing:10px 50px;

}

</style>

</head>

<body>

<table class = "one" border = "1">

<caption>Separate Border Example with border-spacing</caption>

<tr><td> Cell A Collapse Example</td></tr>

<tr><td> Cell B Collapse Example</td></tr>

</table>

<br />

<table class = "two" border = "1">

<caption>Separate Border Example with border-spacing</caption>

<tr><td> Cell A Separate Example</td></tr>

<tr><td> Cell B Separate Example</td></tr>

</table>

</body>

</html>

## The caption-side Property

The caption-side property allows you to specify where the content of a <caption> element should be placed in relationship to the table. The table that follows lists the possible values.

This property can have one of the four values *top, bottom, left*or *right*.

<html>

<head>

<style type = "text/css">

caption.top {caption-side:top}

caption.bottom {caption-side:bottom}

caption.left {caption-side:left}

caption.right {caption-side:right}

</style>

</head>

<body>

<table style = "width:400px; border:1px solid black;">

<caption class = "top">

This caption will appear at the top

</caption>

<tr><td > Cell A</td></tr>

<tr><td > Cell B</td></tr>

</table>

<br />

<table style = "width:400px; border:1px solid black;">

<caption class = "bottom">

This caption will appear at the bottom

</caption>

<tr><td > Cell A</td></tr>

<tr><td > Cell B</td></tr>

</table>

<br />

<table style = "width:400px; border:1px solid black;">

<caption class = "left">

This caption will appear at the left

</caption>

<tr><td > Cell A</td></tr>

<tr><td > Cell B</td></tr>

</table>

<br />

<table style = "width:400px; border:1px solid black;">

<caption class = "right">

This caption will appear at the right

</caption>

<tr><td > Cell A</td></tr>

<tr><td > Cell B</td></tr>

</table>

</body>

</html>

## The empty-cells Property

The empty-cells property indicates whether a cell without any content should have a border displayed.

This property can have one of the three values - *show, hide* or *inherit*.

Here is the empty-cells property used to hide borders of empty cells in the <table> element.

<html>

<head>

<style type = "text/css">

table.empty {

width:350px;

border-collapse:separate;

empty-cells:hide;

}

td.empty {

padding:5px;

border-style:solid;

border-width:1px;

border-color:#999999;

}

</style>

</head>

<body>

<table class = "empty">

<tr>

<th></th>

<th>Title one</th>

<th>Title two</th>

</tr>

<tr>

<th>Row Title</th>

<td class = "empty">value</td>

<td class = "empty">value</td>

</tr>

<tr>

<th>Row Title</th>

<td class = "empty">value</td>

<td class = "empty"></td>

</tr>

</table>

</body>

</html>

# CSS - Borders

The border-color property allows you to change the color of the border surrounding an element. You can individually change the color of the bottom, left, top and right sides of an element's border using the properties −

* **border-bottom-color** changes the color of bottom border.
* **border-top-color** changes the color of top border.
* **border-left-color** changes the color of left border.
* **border-right-color** changes the color of right border.

<html>

<head>

<style type = "text/css">

p.example1 {

border:1px solid;

border-bottom-color:#009900; /\* Green \*/

border-top-color:#FF0000; /\* Red \*/

border-left-color:#330000; /\* Black \*/

border-right-color:#0000CC; /\* Blue \*/

}

p.example2 {

border:1px solid;

border-color:#009900; /\* Green \*/

}

</style>

</head>

<body>

<p class = "example1">

This example is showing all borders in different colors.

</p>

<p class = "example2">

This example is showing all borders in green color only.

</p>

</body>

</html>

The border-style Property

The border-style property allows you to select one of the following styles of border −

* **none** − No border. (Equivalent of border-width:0;)
* **solid** − Border is a single solid line.
* **dotted** − Border is a series of dots.
* **dashed** − Border is a series of short lines.
* **double** − Border is two solid lines.
* **groove** − Border looks as though it is carved into the page.
* **ridge** − Border looks the opposite of groove.
* **inset** − Border makes the box look like it is embedded in the page.
* **outset** − Border makes the box look like it is coming out of the canvas.
* **hidden** − Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties −

* **border-bottom-style** changes the style of bottom border.
* **border-top-style** changes the style of top border.
* **border-left-style** changes the style of left border.
* **border-right-style** changes the style of right border.

<html>

<head>

</head>

<body>

<p style = "border-width:4px; border-style:none;">

This is a border with none width.

</p>

<p style = "border-width:4px; border-style:solid;">

This is a solid border.

</p>

<p style = "border-width:4px; border-style:dashed;">

This is a dashed border.

</p>

<p style = "border-width:4px; border-style:double;">

This is a double border.

</p>

<p style = "border-width:4px; border-style:groove;">

This is a groove border.

</p>

<p style = "border-width:4px; border-style:ridge">

This is a ridge border.

</p>

<p style = "border-width:4px; border-style:inset;">

This is a inset border.

</p>

<p style = "border-width:4px; border-style:outset;">

This is a outset border.

</p>

<p style = "border-width:4px; border-style:hidden;">

This is a hidden border.

</p>

<p style = "border-width:4px;

border-top-style:solid;

border-bottom-style:dashed;

border-left-style:groove;

border-right-style:double;">

This is a a border with four different styles.

</p>

</body>

</html>

## The border-width Property

The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to *thin, medium or thick.*

You can individually change the width of the bottom, top, left, and right borders of an element using the following properties −

* **border-bottom-width** changes the width of bottom border.
* **border-top-width** changes the width of top border.
* **border-left-width** changes the width of left border.
* **border-right-width** changes the width of right border.

<html>

<head>

</head>

<body>

<p style = "border-width:4px; border-style:solid;">

This is a solid border whose width is 4px.

</p>

<p style = "border-width:4pt; border-style:solid;">

This is a solid border whose width is 4pt.

</p>

<p style = "border-width:thin; border-style:solid;">

This is a solid border whose width is thin.

</p>

<p style = "border-width:medium; border-style:solid;">

This is a solid border whose width is medium;

</p>

<p style = "border-width:thick; border-style:solid;">

This is a solid border whose width is thick.

</p>

<p style = "border-bottom-width:4px;border-top-width:10px;

border-left-width: 2px;border-right-width:15px;border-style:solid;">

This is a a border with four different width.

</p>

</body>

</html>

## Border Properties Using Shorthand

The border property allows you to specify color, style, and width of lines in one property −

The following example shows how to use all the three properties into a single property. This is the most frequently used property to set border around any element.

<html>

<head>

</head>

<body>

<p style = "border:4px solid red;">

This example is showing shorthand property for border.

</p>

</body>

</html>

# CSS - Margins

 The **margin-right** specifies the right margin of an element.

Now, we will see how to use these properties with examples.

## The Margin Property

The margin property allows you set all of the properties for the four margins in one declaration.

<html>

<head>

</head>

<body>

<p style = "margin: 15px; border:1px solid black;">

all four margins will be 15px

</p>

<p style = "margin:10px 2%; border:1px solid black;">

top and bottom margin will be 10px, left and right margin will be 2%

of the total width of the document.

</p>

<p style = "margin: 10px 2% -10px; border:1px solid black;">

top margin will be 10px, left and right margin will be 2% of the

total width of the document, bottom margin will be -10px

</p>

<p style = "margin: 10px 2% -10px auto; border:1px solid black;">

top margin will be 10px, right margin will be 2% of the total

width of the document, bottom margin will be -10px, left margin

will be set by the browser

</p>

</body>

</html>

## The margin-bottom Property

The margin-bottom property allows you set bottom margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-bottom: 15px; border:1px solid black;">

This is a paragraph with a specified bottom margin

</p>

<p style = "margin-bottom: 5%; border:1px solid black;">

This is another paragraph with a specified bottom margin in percent

</p>

</body>

</html>

## The margin-top Property

The margin-top property allows you set top margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-top: 15px; border:1px solid black;">

This is a paragraph with a specified top margin

</p>

<p style = "margin-top: 5%; border:1px solid black;">

This is another paragraph with a specified top margin in percent

</p>

</body>

</html>

## The margin-left Property

The margin-left property allows you set left margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-left: 15px; border:1px solid black;">

This is a paragraph with a specified left margin

</p>

<p style = "margin-left: 5%; border:1px solid black;">

This is another paragraph with a specified top margin in percent

</p>

</body>

</html>

## The margin-right Property

The margin-right property allows you set right margin of an element. It can have a value in length, % or auto.

<html>

<head>

</head>

<body>

<p style = "margin-right: 15px; border:1px solid black;">

This is a paragraph with a specified right margin

</p>

<p style = "margin-right: 5%; border:1px solid black;">

This is another paragraph with a specified right margin in percent

</p>

</body>

</html>

# CSS - Lists

## The list-style-type Property

The *list-style-type* property allows you to control the shape or style of bullet point (also known as a marker) in the case of unordered lists and the style of numbering characters in ordered lists.

Here are the values which can be used for an unordered list −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **none** NA |
| 2 | **disc (default)** A filled-in circle |
| 3 | **circle** An empty circle |
| 4 | **square** A filled-in square |

Here are the values, which can be used for an ordered list −

|  |  |  |
| --- | --- | --- |
| **Value** | **Description** | **Example** |
| decimal | Number | 1,2,3,4,5 |
| decimal-leading-zero | 0 before the number | 01, 02, 03, 04, 05 |
| lower-alpha | Lowercase alphanumeric characters | a, b, c, d, e |
| upper-alpha | Uppercase alphanumeric characters | A, B, C, D, E |
| lower-roman | Lowercase Roman numerals | i, ii, iii, iv, v |
| upper-roman | Uppercase Roman numerals | I, II, III, IV, V |
| lower-greek | The marker is lower-greek | alpha, beta, gamma |
| lower-latin | The marker is lower-latin | a, b, c, d, e |
| upper-latin | The marker is upper-latin | A, B, C, D, E |
| hebrew | The marker is traditional Hebrew numbering |  |
| armenian | The marker is traditional Armenian numbering |  |
| georgian | The marker is traditional Georgian numbering |  |
| cjk-ideographic | The marker is plain ideographic numbers |  |
| hiragana | The marker is hiragana | a, i, u, e, o, ka, ki |
| katakana | The marker is katakana | A, I, U, E, O, KA, KI |
| hiragana-iroha | The marker is hiragana-iroha | i, ro, ha, ni, ho, he, to |
| katakana-iroha | The marker is katakana-iroha | I, RO, HA, NI, HO, HE, TO |

<html>

<head>

</head>

<body>

<ul style = "list-style-type:circle;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ul style = "list-style-type:square;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style-type:decimal;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-alpha;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-roman;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

## The list-style-position Property

The *list-style-position* property indicates whether the marker should appear inside or outside of the box containing the bullet points. It can have one the two values −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **none** NA |
| 2 | **inside** If the text goes onto a second line, the text will wrap underneath the marker. It will also appear indented to where the text would have started if the list had a value of outside. |
| 3 | **outside** If the text goes onto a second line, the text will be aligned with the start of the first line (to the right of the bullet). |

<html>

<head>

</head>

<body>

<ul style = "list-style-type:circle; list-stlye-position:outside;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ul style = "list-style-type:square;list-style-position:inside;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style-type:decimal;list-stlye-position:outside;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-alpha;list-style-position:inside;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

## The list-style-image Property

The *list-style-image* allows you to specify an image so that you can use your own bullet style. The syntax is similar to the background-image property with the letters url starting the value of the property followed by the URL in brackets. If it does not find the given image then default bullets are used.

<html>

<head>

</head>

<body>

<ul>

<li style = "list-style-image: url(/images/bullet.gif);">Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol>

<li style = "list-style-image: url(/images/bullet.gif);">Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

## The list-style Property

The *list-style* allows you to specify all the list properties into a single expression. These properties can appear in any order.

<html>

<head>

</head>

<body>

<ul style = "list-style: inside square;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style: outside upper-alpha;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

## The marker-offset Property

The *marker-offset* property allows you to specify the distance between the marker and the text relating to that marker. Its value should be a length as shown in the following example −

Unfortunately, this property is not supported in IE 6 or Netscape 7.

<html>

<head>

</head>

<body>

<ul style = "list-style: inside square; marker-offset:2em;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style: outside upper-alpha; marker-offset:2cm;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

# CSS - Paddings

## The padding-bottom Property

The *padding-bottom* property sets the bottom padding (space) of an element. This can take a value in terms of length of %.

<html>

<head>

</head>

<body>

<p style = "padding-bottom: 15px; border:1px solid black;">

This is a paragraph with a specified bottom padding

</p>

<p style = "padding-bottom: 5%; border:1px solid black;">

This is another paragraph with a specified bottom padding in percent

</p>

</body>

</html>

## The padding-top Property

The *padding-top* property sets the top padding (space) of an element. This can take a value in terms of length of %.

<html>

<head>

</head>

<body>

<p style = "padding-top: 15px; border:1px solid black;">

This is a paragraph with a specified top padding

</p>

<p style = "padding-top: 5%; border:1px solid black;">

This is another paragraph with a specified top padding in percent

</p>

</body>

</html>

## The padding-left Property

The *padding-left* property sets the left padding (space) of an element. This can take a value in terms of length of %.

<html>

<head>

</head>

<body>

<p style = "padding-left: 15px; border:1px solid black;">

This is a paragraph with a specified left padding

</p>

<p style = "padding-left: 15%; border:1px solid black;">

This is another paragraph with a specified left padding in percent

</p>

</body>

</html>

## The padding-right Property

The *padding-right* property sets the right padding (space) of an element. This can take a value in terms of length of %.

<html>

<head>

</head>

<body>

<p style = "padding-right: 15px; border:1px solid black;">

This is a paragraph with a specified right padding

</p>

<p style = "padding-right: 5%; border:1px solid black;">

This is another paragraph with a specified right padding in percent

</p>

</body>

</html>

## The Padding Property

The *padding* property sets the left, right, top and bottom padding (space) of an element. This can take a value in terms of length of %.

<html>

<head>

</head>

<body>

<p style = "padding: 15px; border:1px solid black;">

all four padding will be 15px

</p>

<p style = "padding:10px 2%; border:1px solid black;">

top and bottom padding will be 10px, left and right

padding will be 2% of the total width of the document.

</p>

<p style = "padding: 10px 2% 10px; border:1px solid black;">

top padding will be 10px, left and right padding will

be 2% of the total width of the document, bottom padding will be 10px

</p>

<p style = "padding: 10px 2% 10px 10px; border:1px solid black;">

top padding will be 10px, right padding will be 2% of

the total width of the document, bottom padding and top padding will be 10px

</p>

</body>

</html>

# CSS - Cursors

|  |  |
| --- | --- |
| Sr.No. | Value & Description |
| 1 | **auto** Shape of the cursor depends on the context area it is over. For example an I over text, a hand over a link, and so on... |
| 2 | **crosshair** A crosshair or plus sign |
| 3 | **default** An arrow |
| 4 | **pointer** A pointing hand (in IE 4 this value is hand) |
| 5 | **move** The I bar |
| 6 | e-resize The cursor indicates that an edge of a box is to be moved right (east) |
| 7 | ne-resize The cursor indicates that an edge of a box is to be moved up and right (north/east) |
| 8 | nw-resize The cursor indicates that an edge of a box is to be moved up and left (north/west) |
| 9 | n-resize The cursor indicates that an edge of a box is to be moved up (north) |
| 10 | se-resize The cursor indicates that an edge of a box is to be moved down and right (south/east) |
| 11 | sw-resize The cursor indicates that an edge of a box is to be moved down and left (south/west) |
| 12 | s-resize The cursor indicates that an edge of a box is to be moved down (south) |
| 13 | w-resize The cursor indicates that an edge of a box is to be moved left (west) |
| 14 | text The I bar |
| 15 | wait An hour glass |
| 16 | Help A question mark or balloon, ideal for use over help buttons |
| 17 | <url> The source of a cursor image file |

<html>

<head>

</head>

<body>

<p>Move the mouse over the words to see the cursor change:</p>

<div style = "cursor:auto">Auto</div>

<div style = "cursor:crosshair">Crosshair</div>

<div style = "cursor:default">Default</div>

<div style = "cursor:pointer">Pointer</div>

<div style = "cursor:move">Move</div>

<div style = "cursor:e-resize">e-resize</div>

<div style = "cursor:ne-resize">ne-resize</div>

<div style = "cursor:nw-resize">nw-resize</div>

<div style = "cursor:n-resize">n-resize</div>

<div style = "cursor:se-resize">se-resize</div>

<div style = "cursor:sw-resize">sw-resize</div>

<div style = "cursor:s-resize">s-resize</div>

<div style = "cursor:w-resize">w-resize</div>

<div style = "cursor:text">text</div>

<div style = "cursor:wait">wait</div>

<div style = "cursor:help">help</div>

</body>

</html>

# CSS - Outlines

* The **outline-width** property is used to set the width of the outline.
* The **outline-style** property is used to set the line style for the outline.
* The **outline-color** property is used to set the color of the outline.
* The **outline** property is used to set all the above three properties in a single statement.

## The outline-width Property

The *outline-width* property specifies the width of the outline to be added to the box. Its value should be a length or one of the values *thin, medium, or thick,*just like the border-width attribute.

A width of zero pixels means no outline.

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;">

This text is having thin outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:solid;">

This text is having thick outline.

</p>

<br />

<p style = "outline-width:5px; outline-style:solid;">

This text is having 5x outline.

</p>

</body>

</html>

## The outline-style Property

The *outline-style* property specifies the style for the line (solid, dotted, or dashed) that goes around an element. It can take one of the following values −

* **none** − No border. (Equivalent of outline-width:0;)
* **solid** − Outline is a single solid line.
* **dotted** − Outline is a series of dots.
* **dashed** − Outline is a series of short lines.
* **double** − Outline is two solid lines.
* **groove** − Outline looks as though it is carved into the page.
* **ridge** − Outline looks the opposite of groove.
* **inset** − Outline makes the box look like it is embedded in the page.
* **outset** − Outline makes the box look like it is coming out of the canvas.
* **hidden** − Same as none.

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;">

This text is having thin solid outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:dashed;">

This text is having thick dashed outline.

</p>

<br />

<p style = "outline-width:5px;outline-style:dotted;">

This text is having 5x dotted outline.

</p>

</body>

</html>

## The outline-color Property

The *outline-color* property allows you to specify the color of the outline. Its value should either be a color name, a hex color, or an RGB value, as with the color and border-color properties.

<html>

<head>

</head>

<body>

<p style = "outline-width:thin; outline-style:solid;outline-color:red">

This text is having thin solid red outline.

</p>

<br />

<p style = "outline-width:thick; outline-style:dashed;outline-color:#009900">

This text is having thick dashed green outline.

</p>

<br />

<p style = "outline-width:5px;outline-style:dotted;outline-color:rgb(13,33,232)">

This text is having 5x dotted blue outline.

</p>

</body>

</html>

## The outline Property

The *outline* property is a shorthand property that allows you to specify values for any of the three properties discussed previously in any order but in a single statement.

<html>

<head>

</head>

<body>

<p style = "outline:thin solid red;">

This text is having thin solid red outline.

</p>

<br />

<p style = "outline:thick dashed #009900;">

This text is having thick dashed green outline.

</p>

<br />

<p style = "outline:5px dotted rgb(13,33,232);">

This text is having 5x dotted blue outline.

</p>

</body>

</html>

# CSS - Dimension

The **max-width** property is used to set the maximum width that a box can be.

The **min-width** property is used to set the minimum width that a box can be.

## The Height and Width Properties

The *height* and *width* properties allow you to set the height and width for boxes. They can take values of a length, a percentage, or the keyword auto.

<html>

<head>

</head>

<body>

<p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;">

This paragraph is 400pixels wide and 100 pixels high

</p>

</body>

</html>

## The line-height Property

The *line-height* property allows you to increase the space between lines of text. The value of the line-height property can be a number, a length, or a percentage.

<html>

<head>

</head>

<body>

<p style = "width:400px; height:100px; border:1px solid red; padding:5px; margin:10px; line-height:30px;">

This paragraph is 400pixels wide and 100 pixels high and here line height is 30pixels.

This paragraph is 400 pixels wide and 100 pixels high and here line height is 30pixels.

</p>

</body>

</html>

## The max-height Property

The *max-height* property allows you to specify maximum height of a box. The value of the max-height property can be a number, a length, or a percentage.

<html>

<head>

</head>

<body>

<p style = "width:400px; max-height:10px; border:1px solid red; padding:5px; margin:10px;">

This paragraph is 400px wide and max height is 10px

This paragraph is 400px wide and max height is 10px

This paragraph is 400px wide and max height is 10px

This paragraph is 400px wide and max height is 10px

</p>

<br>

<br>

<br>

<img alt = "logo" src = "/css/images/logo.png" width = "195" height = "84" />

</body>

</html>

## The min-height Property

The *min-height* property allows you to specify minimum height of a box. The value of the min-height property can be a number, a length, or a percentage.

<html>

<head>

</head>

<body>

<p style = "width:400px; min-height:200px; border:1px solid red; padding:5px; margin:10px;">

This paragraph is 400px wide and min height is 200px

This paragraph is 400px wide and min height is 200px

This paragraph is 400px wide and min height is 200px

This paragraph is 400px wide and min height is 200px

</p>

<img alt = "logo" src = "/css/images/logo.png" width = "95" height = "84" />

</body>

</html>

## The max-width Property

The *max-width* property allows you to specify maximum width of a box. The value of the max-width property can be a number, a length, or a percentage.

<html>

<head>

</head>

<body>

<p style = "max-width:100px; height:200px; border:1px solid red; padding:5px; margin:10px;">

This paragraph is 200px high and max width is 100px

This paragraph is 200px high and max width is 100px

This paragraph is 200px high and max width is 100px

This paragraph is 200px high and max width is 100px

This paragraph is 200px high and max width is 100px

</p>

<img alt = "logo" src = "/images/css.gif" width = "95" height = "84" />

</body>

</html>

## The min-width Property

The *min-width* property allows you to specify minimum width of a box. The value of the min-width property can be a number, a length, or a percentage.

<html>

<head>

</head>

<body>

<p style = "min-width:400px; height:100px; border:1px solid red; padding:5px; margin:10px;">

This paragraph is 100px high and min width is 400px

This paragraph is 100px high and min width is 400px

</p>

<img alt = "logo" src = "/css/images/css.gif" width = "95" height = "84" />

</body>

</html>

# CSS – Scrollbars

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **visible** Allows the content to overflow the borders of its containing element. |
| 2 | **hidden** he content of the nested element is simply cut off at the border of the containing element and no scrollbars is visible. |
| 3 | **scroll** The size of the containing element does not change, but the scrollbars are added to allow the user to scroll to see the content. |
| 4 | **auto** The purpose is the same as scroll, but the scrollbar will be shown only if the content does overflow. |

<html>

<head>

<style type = "text/css">

.scroll {

display:block;

border: 1px solid red;

padding:5px;

margin-top:5px;

width:300px;

height:50px;

overflow:scroll;

}

.auto {

display:block;

border: 1px solid red;

padding:5px;

margin-top:5px;

width:300px;

height:50px;

overflow:auto;

}

</style>

</head>

<body>

<p>Example of scroll value:</p>

<div class = "scroll">

I am going to keep lot of content here just to show you how

scrollbars works if there is an overflow in an element box.

This provides your horizontal as well as vertical scrollbars.

</div>

<br />

<p>Example of auto value:</p>

<div class = "auto">

I am going to keep lot of content here just to show you how

scrollbars works if there is an overflow in an element box.

This provides your horizontal as well as vertical scrollbars.

</div>

</body>

</html>

CSS - Visibility

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **visible** The box and its contents are shown to the user. |
| 2 | **hidden** The box and its content are made invisible, although they still affect the layout of the page. |
| 3 | **collapse** This is for use only with dynamic table columns and row effects. |

<html>

<head>

</head>

<body>

<p>

This paragraph should be visible in normal way.

</p>

<p style = "visibility:hidden;">

This paragraph should not be visible.

</p>

</body>

</html>

# CSS - Positioning

Move Up - Use a negative value for *top*.

Move Down - Use a positive value for *top*.

<html>

<head>

</head>

<body>

<div style = "**position:relative; left:80px; top:2px;** background-color:yellow;">

This div has relative positioning.

</div>

</body>

</html>

## Absolute Positioning

An element with **position: absolute** is positioned at the specified coordinates relative to your screen top-left corner.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

* Move Left - Use a negative value for *left*.
* Move Right - Use a positive value for *left*.
* Move Up - Use a negative value for *top*.
* Move Down - Use a positive value for *top*.

**NOTE** − You can use *bottom* or *right* values as well in the same way as top and left.

<html>

<head>

</head>

<body>

<div style = "**position:absolute; left:80px; top:20px;** background-color:yellow;">

This div has absolute positioning.

</div>

</body>

</html>

## Fixed Positioning

Fixed positioning allows you to fix the position of an element to a particular spot on the page, regardless of scrolling. Specified coordinates will be relative to the browser window.

You can use two values *top* and *left* along with the *position* property to move an HTML element anywhere in the HTML document.

* Move Left - Use a negative value for *left*.
* Move Right - Use a positive value for *left*.
* Move Up - Use a negative value for *top*.
* Move Down - Use a positive value for *top*.

**NOTE** − You can use *bottom* or *right* values as well in the same way as *top*and *left*.

<html>

<head>

</head>

<body>

<div style = "**position:fixed; left:80px; top:20px;** background-color:yellow;">

This div has fixed positioning.

</div>

</body>

</html>

# CSS - Layers

CSS gives you opportunity to create layers of various divisions. The CSS layers refer to applying the z-index property to elements that overlap with each other.

The z-index property is used along with the position property to create an effect of layers. You can specify which element should come on top and which element should come at bottom.

A z-index property can help you to create more complex webpage layouts. Following is the example which shows how to create layers in CSS.

<html>

<head>

</head>

<body>

<div style = "background-color:red;

width:300px;

height:100px;

position:relative;

top:10px;

left:80px;

z-index:2">

</div>

<div style = "background-color:yellow;

width:300px;

height:100px;

position:relative;

top:-60px;

left:35px;

z-index:1;">

</div>

<div style = "background-color:green;

width:300px;

height:100px;

position:relative;

top:-220px;

left:120px;

z-index:3;">

</div>

</body>

</html>

# CSS - Pseudo Classes

The most commonly used pseudo-classes are as follows −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **:link** Use this class to add special style to an unvisited link. |
| 2 | **:visited** Use this class to add special style to a visited link. |
| 3 | **:hover** Use this class to add special style to an element when you mouse over it. |
| 4 | **:active** Use this class to add special style to an active element. |
| 5 | **:focus** Use this class to add special style to an element while the element has focus. |
| 6 | **:first-child** Use this class to add special style to an element that is the first child of some other element. |
| 7 | **:lang** Use this class to specify a language to use in a specified element. |

While defining pseudo-classes in a <style>...</style> block, following points should be noted −

* a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.
* a:active MUST come after a:hover in the CSS definition in order to be effective.
* Pseudo-class names are not case-sensitive.
* Pseudo-class are different from CSS classes but they can be combined.

## The :link pseudo-class

The following example demonstrates how to use the *:link* class to set the link color. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:link {color:#000000}

</style>

</head>

<body>

<a href = "">Black Link</a>

</body>

</html>

## The :visited pseudo-class

The following is the example which demonstrates how to use the *:visited* class to set the color of visited links. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:visited {color: #006600}

</style>

</head>

<body>

<a href = "">Click this link</a>

</body>

</html>

## The :hover pseudo-class

The following example demonstrates how to use the *:hover* class to change the color of links when we bring a mouse pointer over that link. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:hover {color: #FFCC00}

</style>

</head>

<body>

<a href = "">Bring Mouse Here</a>

</body>

</html>

## The :active pseudo-class

The following example demonstrates how to use the *:active* class to change the color of active links. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:active {color: #FF00CC}

</style>

</head>

<body>

<a href = "">Click This Link</a>

</body>

</html>

## The :focus pseudo-class

The following example demonstrates how to use the *:focus* class to change the color of focused links. Possible values could be any color name in any valid format.

<html>

<head>

<style type = "text/css">

a:focus {color: #0000FF}

</style>

</head>

<body>

<a href = "">Click this Link</a>

</body>

</html>

## The :first-child pseudo-class

The *:first-child* pseudo-class matches a specified element that is the first child of another element and adds special style to that element that is the first child of some other element.

To make :first-child work in IE <!DOCTYPE> must be declared at the top of document.

For example, to indent the first paragraph of all <div> elements, you could use this definition −

<html>

<head>

<style type = "text/css">

div > p:first-child {

text-indent: 25px;

}

</style>

</head>

<body>

<div>

<p>First paragraph in div. This paragraph will be indented</p>

<p>Second paragraph in div. This paragraph will not be indented</p>

</div>

<p>But it will not match the paragraph in this HTML:</p>

<div>

<h3>Heading</h3>

<p>The first paragraph inside the div. This paragraph will not be effected.</p>

</div>

</body>

</html>

## The :lang pseudo-class

The language pseudo-class *:lang*, allows constructing selectors based on the language setting for specific tags.

This class is useful in documents that must appeal to multiple languages that have different conventions for certain language constructs. For example, the French language typically uses angle brackets (< and >) for quoting purposes, while the English language uses quote marks (' and ').

In a document that needs to address this difference, you can use the :lang pseudo-class to change the quote marks appropriately. The following code changes the <blockquote> tag appropriately for the language being used −

<html>

<head>

<style type = "text/css">

/\* Two levels of quotes for two languages\*/

:lang(en) { quotes: '"' '"' "'" "'"; }

:lang(fr) { quotes: "<<" ">>" "<" ">"; }

</style>

</head>

<body>

<p>...<q lang = "fr">A quote in a paragraph</q>...</p>

</body>

</html>

The :lang selectors will apply to all the elements in the document. However, not all elements make use of the quotes property, so the effect will be transparent for most elements.

# CSS - Pseudo Elements

The most commonly used pseudo-elements are as follows −

|  |  |
| --- | --- |
| **Sr.No.** | **Value & Description** |
| 1 | **:first-line** Use this element to add special styles to the first line of the text in a selector. |
| 2 | **:first-letter** Use this element to add special style to the first letter of the text in a selector. |
| 3 | **:before** Use this element to insert some content before an element. |
| 4 | **:after** Use this element to insert some content after an element. |

## The :first-line pseudo-element

The following example demonstrates how to use the *:first-line* element to add special effects to the first line of elements in the document.

<html>

<head>

<style type = "text/css">

p:first-line { text-decoration: underline; }

p.noline:first-line { text-decoration: none; }

</style>

</head>

<body>

<p class = "noline">

This line would not have any underline because this belongs to nline class.

</p>

<p>

The first line of this paragraph will be underlined as defined in the

CSS rule above. Rest of the lines in this paragraph will remain normal.

This example shows how to use :first-line pseduo element to give effect

to the first line of any HTML element.

</p>

</body>

</html>

## The :first-letter pseudo-element

The following example demonstrates how to use the *:first-letter* element to add special effects to the first letter of elements in the document.

<html>

<head>

<style type = "text/css">

p:first-letter { font-size: 5em; }

p.normal:first-letter { font-size: 10px; }

</style>

</head>

<body>

<p class = "normal">

First character of this paragraph will be normal and will have font size 10 px;

</p>

<p>

The first character of this paragraph will be 5em big as defined in the

CSS rule above. Rest of the characters in this paragraph will remain

normal. This example shows how to use :first-letter pseduo element

to give effect to the first characters of any HTML element.

</p>

</body>

</html>

## The :before pseudo-element

The following example demonstrates how to use the *:before* element to add some content before any element.

<html>

<head>

<style type = "text/css">

p:before {

content: url(/images/bullet.gif)

}

</style>

</head>

<body>

<p> This line will be preceded by a bullet.</p>

<p> This line will be preceded by a bullet.</p>

<p> This line will be preceded by a bullet.</p>

</body>

</html>

## The :after pseudo-element

The following example demonstrates how to use the *:after* element to add some content after any element.

<html>

<head>

<style type = "text/css">

p:after {

content: url(/images/bullet.gif)

}

</style>

</head>

<body>

<p> This line will be succeeded by a bullet.</p>

<p> This line will be succeeded by a bullet.</p>

<p> This line will be succeeded by a bullet.</p>

</body>

</html>

# CSS - @ Rules

## The @import rule

The @import rule allows you to import styles from another style sheet. It should appear right at the start of the style sheet before any of the rules, and its value is a URL.

<style type = "text/css">

<!--

@import "mystyle.css";

or

@import url("mystyle.css");

.......other CSS rules .....

-->

</style>

The significance of the @import rule is that it allows you to develop your style sheets with a modular approach. You can create various style sheets and then include them wherever you need them.

## The @charset Rule

If you are writing your document using a character set other than ASCII or ISO-8859-1 you might want to set the @charset rule at the top of your style sheet to indicate what character set the style sheet is written in.

The @charset rule must be written right at the beginning of the style sheet without even a space before it. The value is held in quotes and should be one of the standard character-sets. For example −

<style type = "text/css">

<!--

@charset "iso-8859-1"

.......other CSS rules .....

-->

</style>

## The @font-face Rule

The @font-face rule is used to exhaustively describe a font face for use in a document. @font-face may also be used to define the location of a font for download, although this may run into implementation-specific limits.

In general, @font-face is extremely complicated, and its use is not recommended for any except those who are expert in font metrics.

<style type = "text/css">

<!--

@font-face {

font-family: "Scarborough Light";

src: url("http://www.font.site/s/scarbo-lt");

}

@font-face {

font-family: Santiago;

src: local ("Santiago"),

url("http://www.font.site/s/santiago.tt")

format("truetype");

unicode-range: U+??,U+100-220;

font-size: all;

font-family: sans-serif;

}

-->

</style>

## The !important Rule

Cascading Style Sheets cascade. It means that the styles are applied in the same order as they are read by the browser. The first style is applied and then the second and so on.

The !important rule provides a way to make your CSS cascade. It also includes the rules that are to be applied always. A rule having a !important property will always be applied, no matter where that rule appears in the CSS document.

For example, in the following style sheet, the paragraph text will be black, even though the first style property applied is red:

<style type = "text/css">

<!--

p { color: #ff0000; }

p { color: #000000; }

-->

</style>

So, if you wanted to make sure that a property always applied, you would add the !important property to the tag. So, to make the paragraph text always red, you should write it as follows −

<html>

<head>

<style type = "text/css">

p { color: #ff0000 !important; }

p { color: #000000; }

</style>

</head>

<body>

<p>Tutorialspoint.com</p>

</body>

</html>

Here you have made *p { color: #ff0000 !important; }* mandatory, now this rule will always apply even you have defined another rule *p { color: #000000; }*

# CSS Paged Media - @page Rule

The CSS2 defines a "page box", a box of finite dimensions in which content is rendered. The page box is a rectangular region that contains two areas −

* **The page area** − The page area includes the boxes laid out on that page. The edges of the page area act as the initial containing block for layout that occurs between page breaks.
* **The margin area** − It surrounds the page area.

You can specify the dimensions, orientation, margins, etc., of a page box within an @page rule. The dimensions of the page box are set with the 'size' property. The dimensions of the page area are the dimensions of the page box minus the margin area.

For example, the following @page rule sets the page box size to 8.5 × 11 inches and creates '2cm' margin on all sides between the page box edge and the page area −

<style type = "text/css">

<!--

@page { size:8.5in 11in; margin: 2cm }

-->

</style>

You can use the *margin, margin-top, margin-bottom, margin-left, and margin-right* properties within the @page rule to set margins for your page.

Finally, the *marks* property is used within the @page rule to create crop and registration marks outside the page box on the target sheet. By default, no marks are printed. You may use one or both of the *crop* and *cross* keywords to create crop marks and registration marks, respectively, on the target print page.

Setting Page Size

The *size* property specifies the size and orientation of a page box. There are four values which can be used for page size −

* **auto** − The page box will be set to the size and orientation of the target sheet.
* **landscape** − Overrides the target's orientation. The page box is the same size as the target, and the longer sides are horizontal.
* **portrait** − Overrides the target's orientation. The page box is the same size as the target, and the shorter sides are horizontal.
* **length** − Length values for the 'size' property create an absolute page box. If only one length value is specified, it sets both the width and height of the page box. Percentage values are not allowed for the 'size' property.

In the following example, the outer edges of the page box will align with the target. The percentage value on the 'margin' property is relative to the target size so if the target sheet dimensions are 21.0cm × 29.7cm (i.e., A4), the margins are 2.10cm and 2.97cm.

<style type = "text/css">

<!--

@page {

size: auto; /\* auto is the initial value \*/

margin: 10%;

}

-->

</style>

The following example sets the width of the page box to be 8.5 inches and the height to be 11 inches. The page box in this example requires a target sheet size of 8.5" × 11" or larger.

<style type = "text/css">

<!--

@page {

size: 8.5in 11in; /\* width height \*/

}

-->

</style>

Once you create a named page layout, you can use it in your document by adding the page property to a style that is later applied to an element in your document. For example, this style renders all the tables in your document on landscape pages −

<style type = "text/css">

<!--

@page { size : portrait }

@page rotated { size : landscape }

table { page : rotated }

-->

</style>

Due to the above rule, while printing, if the browser encounters a <table> element in your document and the current page layout is the default portrait layout, it starts a new page and prints the table on a landscape page.

Left, Right, and First pages

When printing double-sided documents, the page boxes on left and right pages should be different. It can be expressed through two CSS pseudo-classes as follows −

<style type = "text/css">

<!--

@page :left {

margin-left: 4cm;

margin-right: 3cm;

}

@page :right {

margin-left: 3cm;

margin-right: 4cm;

}

-->

</style>

You can specify the style for the first page of a document with the :first pseudo-class −

<style type = "text/css">

<!--

@page { margin: 2cm } /\* All margins set to 2cm \*/

@page :first {

margin-top: 10cm /\* Top margin on first page 10cm \*/

}

-->

</style>

Controlling Pagination

Unless you specify otherwise, page breaks occur only when the page format changes or when the content overflows the current page box. To otherwise force or suppress page breaks, use the *page-break-before, page-break-after,*and *page-break-inside* properties.

Both the *page-break-before* and *page-break-after* accept the *auto, always, avoid, left,* and *right* keywords.

The keyword *auto* is the default, it lets the browser generate page breaks as needed. The keyword *always* forces a page break before or after the element, while *avoid* suppresses a page break immediately before or after the element. The *left* and *right* keywords force one or two page breaks, so that the element is rendered on a left-hand or right-hand page.

Using pagination properties is quite straightforward. Suppose your document has level-1 headers start new chapters with level-2 headers to denote sections. You'd like each chapter to start on a new, right-hand page, but you don't want section headers to be split across a page break from the subsequent content. You can achieve this using following rule −

<style type = "text/css">

<!--

h1 { page-break-before : right }

h2 { page-break-after : avoid }

-->

</style>

Use only the *auto* and *avoid* values with the *page-break-inside* property. If you prefer that your tables not be broken across pages if possible, you would write the rule −

<style type = "text/css">

<!--

table { page-break-inside : avoid }

-->

</style>

# CSS Printing - @media Rule

<style type = "text/css">

<!--

@media screen {

p.bodyText {font-family:verdana, arial, sans-serif;}

}

@media print {

p.bodyText {font-family:georgia, times, serif;}

}

@media screen, print {

p.bodyText {font-size:10pt}

}

-->

</style>

If you are defining your style sheet in a separate file, then you can also use the media attribute when linking to an external style sheet −

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

# CSS - Layouts

* CSS is pivotal to the future of Web documents and will be supported by most browsers.
* CSS is more exact than tables, allowing your document to be viewed as you intended, regardless of the browser window.
* Keeping track of nested tables can be a real pain. CSS rules tend to be well organized, easily read, and easily changed.

CSS also provides *table-layout* property to make your tables load much faster. Following is an example −

<table style = "**table-layout:fixed**;width:600px;">

<tr height = "30">

<td width = "150">CSS table layout cell 1</td>

<td width = "200">CSS table layout cell 2</td>

<td width = "250">CSS table layout cell 3</td>

</tr>

</table>

## Sample Column Layout

<style style = "text/css">

body {

margin:9px 9px 0 9px;

padding:0;

background:#FFF;

}

#level0 {background:#FC0;}

#level1 {

margin-left:143px;

padding-left:9px;

background:#FFF;

}

#level2 {background:#FFF3AC;}

#level3 {

margin-right:143px;

padding-right:9px;

background:#FFF;

}

#main {background:#CCC;}

</style>

<body>

<div id = "level0">

<div id = "level1">

<div id = "level2">

<div id = "level3">

<div id = "main">

Final Content goes here...

</div>

</div>

</div>

</div>

</div>

</body>