

CSS Basics

CSS is a language that describes the style of an HTML document.

CSS describes how HTML elements should be displayed.

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

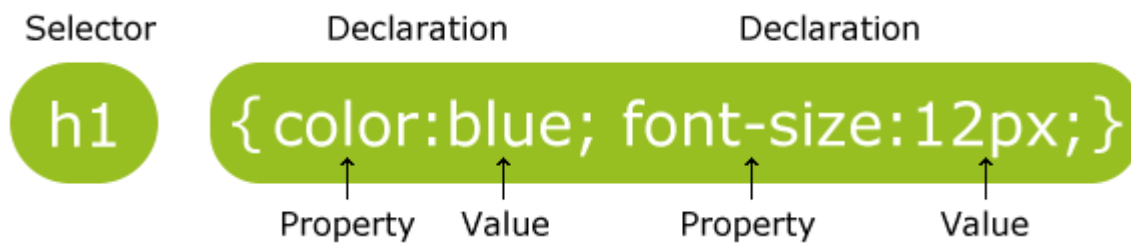
CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen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

Advantages of CSS

- **CSS saves time** – You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
- **Pages load faster** – If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.
- **Easy maintenance** – To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** – Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** – Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.
- **Offline Browsing** – CSS can store web applications locally with the help of an offline cache. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.
- **Platform Independence** – The Script offer consistent platform independence and can support latest browsers as well.

CSS Syntax

- A CSS rule-set consists of a selector and a declaration block:



- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a CSS property name and a value, separated by a colon.
- A CSS declaration always ends with a semicolon, and declaration blocks are surrounded by curly braces.

In the following example all <p> elements will be center-aligned, with a red text color:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: red;
  text-align: center;
}
</style>
</head>
<body>

<p>Hello World!</p>
<p>These paragraphs are styled with CSS.</p>

</body>
</html>
```

CSS Selectors

CSS selectors are used to "find" (or select) HTML elements based on their element name, id and class.

The element Selector

The element selector selects elements based on the element name.

You can select all <p> elements on a page like this (in this case, all <p> elements will be center-aligned, with a red text color):

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Every paragraph will be affected by the style.</p>
<p id="para1">Me too!</p>
<p>And me!</p>

</body>
</html>
```

The id Selector

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element should be unique within a page, so the id selector is used to select one unique element!
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

The style rule below will be applied to the HTML element with id="para1":

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
```

```

text-align: center;
color: red;
}
</style>
</head>
<body>

<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>

</body>
</html>

```

The class Selector

- The class selector selects elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the name of the class.

In the example below, all HTML elements with class="center" will be red and center-aligned:

Example:

```

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

<h1 class="center">Red and center-aligned heading</h1>
<p class="center">Red and center-aligned paragraph.</p>

</body>
</html>

```

You can also specify that only specific HTML elements should be affected by a class.

In the example below, only <p> elements with class="center" will be center-aligned:

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>

```

```

p.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>

</body>
</html>

```

HTML elements can also refer to more than one class.

In the example below, the <p> element will be styled according to class="center" and to class="large":

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
p.center {
  text-align: center;
  color: red;
}

p.large {
  font-size: 300%;
}
</style>
</head>
<body>

<h1 class="center">This heading will not be affected</h1>
<p class="center">This paragraph will be red and center-aligned.</p>
<p class="center large">This paragraph will be red, center-aligned, and in a large font-size.</p>

</body>
</html>

```

Grouping Selectors

If you have elements with the same style definitions, like this:

```
h1 {
  text-align: center;
  color: red;
}
h2 {
  text-align: center;
  color: red;
}
p {
  text-align: center;
  color: red;
}
```

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

In the example below we have grouped the selectors from the code above:

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code at a later date.

Comments are ignored by browsers.

A CSS comment starts with `/*` and ends with `*/`. Comments can also span multiple lines:

Example

```
p {
  color: red;
  /* This is a single-line comment */
  text-align: center;
}

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

Three Ways to Insert CSS (Types of CSS):

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

Inline Styles

- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

The example below shows how to change the color and the left margin of a <h1> element:

Example:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color: blue; margin-left: 30px ;"> This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Internal Style Sheet

- An internal style sheet may be used if one single page has a unique style.
- Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
```

```

}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>

```

External Style Sheet

- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

Example:

```

<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>

```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.

Here is how the "mystyle.css" looks:

```

.....

body {
    background-color: lightblue;
}

```



```
h1 {
    color: navy;
    margin-left: 20px;
}
```

Multiple Style Sheets

If some properties have been defined for the same selector (element) in different style sheets, the value from the last read style sheet will be used.

Example

Assume that an external style sheet "mystyle.css" has the following style for the <h1> element:

```
h1 {
    color: navy;
}
```

Then, assume that an internal style sheet also has the following style for the <h1> element:

```
h1 {
    color: orange;
}
```

If the internal style is defined after the link to the external style sheet, the <h1> elements will be "orange":

See following example.

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
    color: orange;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
```

```
<p>The style of this document is a combination of an external stylesheet, and internal style</p>
```

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

However, if the internal style is defined before the link to the external style sheet, the `<h1>` elements will be "navy":

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
```

```
<h1>This is a heading</h1>
<p>The style of this document is a combination of an external stylesheet, and internal style</p>
```

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

Cascading Order

What style will be used when there is more than one style specified for an HTML element?

Generally speaking we can say that all the styles will "cascade" into a new "virtual" style sheet by the following rules, where number one has the highest priority:

1. Inline style (inside an HTML element)
2. External and internal style sheets (in the head section)
3. Browser default

So, an inline style (inside a specific HTML element) has the highest priority, which means that it will override a style defined inside the `<head>` tag, or in an external style sheet, or a browser default value.

CSS Properties:

Text Properties:

- color: blue;
- text-align: center/left/right/justify;
- text-decoration: overline/line-through/underline;
- text-transform: uppercase/lowercase/capitalize;
- text-indent: 10px/-10px;
- letter-spacing: 5px/-5px;
- line-height: 0.8;
- direction: rtl;
- word-spacing: 10px;
- text-shadow: 3px 2px red;

Example:

```
html>
<head>
<style type="text/css">

p.color {
  color: blue;
}

p.align {
  text-align: right;
}

p.decoration {
  text-decoration: line-through;
}

p.indent {
  text-indent: 50px;
}

p.lspace {
  letter-spacing: 5px;
}
```

```
p.shadow {
  text-shadow: 3px 2px red;
}
```

```
p.height {
  line-height: 5.5;
}
```

```
p.uppercase {
  text-transform: uppercase;
}
```

```
p.lowercase {
  text-transform: lowercase;
}
```

```
p.capitalize {
  text-transform: capitalize;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="color">This is some text in blue color.</p>
```

```
<p class="align">This is some text aligned right.</p>
```

```
<p class="decoration">This is some text line through</p>
```

```
<p class="indent">This is some text with 50px indent</p>
```

```
<p class="lspace">This is some text with line spacing</p>
```

```
<p class="shadow">This is some shadow text</p>
```

```
<p class="height">This is some text with line height</p>
```

```
<p class="uppercase">This is some uppercase text.</p>
```

```
<p class="lowercase">This is some lowercase text.</p>
```

```
<p class="capitalize">This is some capitalized text.</p>
```

```
</body>
```

```
</html>
```

Font Properties:

1. font-family: arial;
2. font-style: normal/italic/oblique;
3. font-size: 30px;
4. font-weight: normal/bold;
5. font-variant: normal/small-caps;

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
p.fmly {
    font-family: arial;
}

p.fmly1 {
    font-family: symbol;
}

p.normal {
    font-style: normal;
}

p.italic {
    font-style: italic;
}

p.oblique {
    font-style: oblique;
}

h1 {
    font-size: 40px;
}

h2 {
    font-size: 30px;
}

p {
    font-size: 14px;
}

p.normal {
    font-weight: normal;
}

p.light {
    font-weight: lighter;
}

p.thick {
    font-weight: bold;
}

p.thicker {
    font-weight: 900;
}

p.normal {
    font-variant: normal;
}

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

```

</style>
</head>
<body>

<h1>CSS font-family</h1>
<p class="fmly">This is a paragraph</p>
<p class="fmly1">This is a paragraph</p>
<p class="normal">This is a paragraph in normal style.</p>
<p class="italic">This is a paragraph in italic style.</p>
<p class="oblique">This is a paragraph in oblique style.</p>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
<p class="normal">this is normal text</p>
<p class="small">Capitalised Small text.</p>

</body>
</html>

```

Border Properties:

1. border-style: dotted/dashed/solid/double/groove/ridge/inset/outset/none/hidden;
2. border-width: 5px;
border-width: 2px 10px 4px 20px;
3. border-color: red;
Border-color: red green blue yellow;

Note: four values indicates - first for top border, second for right border, third for bottom border, fourth for left border.

Example:

```

<!DOCTYPE html>
<html>
  <head>
    <style>
      p.one {
        border-style: solid;
        border-color: red;
        margin: 10px 100px 10px 10px;
      }

      p.two {
        border-style: solid;
        border-color: green;

```

```

    }

    p.three {
        border-style: solid;
        border-color: red green blue yellow;
    }
</style>
</head>
<body>

<h2>The border-color Property</h2>
<p>This property specifies the color of the four borders:</p>

<p class="one">A solid red border</p>
<p class="two">A solid green border</p>
<p class="three">A solid multicolor border</p>
</body>
</html>

```

Background Properties:

- background-color: red;
- background-image: url("image_path");
- background-repeat: repeat-x/repeat-y/round/no-repeat;
- background-attachment: fixed/scroll/local;
- background-size: 20px 30px;
- background-position: right top/right bottom/left top/left bottom;
or
30px 20px;

Margin Properties:

- margin-top: 100px;
- margin-bottom: 100px;
- margin-right: 150px;
- margin-left: 80px;

Link Properties:

- a:link - unvisited link
- a:hover - when mouse is on it
- a:active - when mouse click on it
- a:visited - after mouse clicked

Example:

```

<!DOCTYPE html>
<html>
<head>

```

```

<style>
/* unvisited link */
a:link {
    color: red;
}

/* mouse over link */
a:hover {
    color: green;
    text-decoration:none;
}

/* selected link */
a:active {
    color: blue;
}

/* visited link */
a:visited {
    color: brown;
}

</style>
</head>
<body>

<a href="text_css.html" target="_blank">This is a link</a>

</body>
</html>

```

Layouts:

HTML Layout Elements

Websites often display content in multiple columns (like a magazine or newspaper).

HTML5 offers new semantic elements that define the different parts of a web page:



- <header> - Defines a header for a document or a section
- <nav> - Defines a container for navigation links
- <section> - Defines a section in a document
- <article> - Defines an independent self-contained article
- <aside> - Defines content aside from the content (like a sidebar)
- <footer> - Defines a footer for a document or a section
- <details> - Defines additional details
- <summary> - Defines a heading for the <details> element

Example:

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
  height: 100%;
}
div.container {

  width:100%;
  border: 1px solid red;
}

header, footer {
  padding: 1em;
  color: white;
  background-color: black;
  text-align: center;
}

nav {
  float: left;
  max-width: 10px;
  margin: 0;
  padding: 1em;
}

nav ul {
  list-style-type: none;
  padding: 0;
}

nav ul a {
  text-decoration: none;
}

article {
  margin-left: 170px;
  border-left: 1px solid gray;

  padding: 1em;
  overflow: hidden;
}
</style>
</head>
<body>

<div class="container">

<header>
  <h1>LBRCE, Mylavaram</h1>
</header>

<nav>
  <ul>

```

```

<li><a href="layout.html" target="blank">CSE</a></li>
<li><a href="#">ECE</a></li>
<li><a href="#">EEE</a></li>
</ul>
</nav>

<article id="content">
  <h1>CSE</h1>
  <p>welcome to cse dept.</p>
  <p>this is located at block1</p>
</article>

<footer>Copyright &copy; CSE solutions</footer>

</div>

</body>
</html>

```

Output:

LBRCE, Mylavaram	
<div> <div>CSE</div> <div>ECE</div> <div>EEE</div> </div>	<div>CSE</div> <div>welcome to cse dept.</div> <div>this is located at block1</div>
Copyright © CSE solutions	