

CSS(Cascading Style Sheets)

CSS stands for Cascading Style Sheets
Styles define **how to display** HTML
elements


CSS has various levels and profiles. Each level of CSS builds upon the last, typically adding new features and typically denoted as CSS₁, CSS₂, and CSS₃.

The first CSS specification to become an official W₃C Recommendation is CSS level 1, published in December 1996

CSS level 2 was developed by the W₃C and published as a Recommendation in May 1998. A superset of CSS₁, CSS₂ includes a number of new capabilities like absolute, relative, and fixed positioning of elements and z-index, the concept of media types etc.

CSS 3

CSS level 3 is currently under development. The W₃C maintains a CSS₃ progress report.

- 
- CSS Saves a Lot of Work!
 - CSS defines HOW HTML elements are to be displayed.
 - Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by editing one single file!

Understanding Style Rules

- A Style Rule is composed of two parts: a selector and a declaration.



- The Selector indicates the element to which the rule is applied.
- The Declaration determines the property values of a selector.

Understanding Style Rules

- The Property specifies a characteristic, such as color, font-family, position, and is followed by a colon (:).
- The Value expresses specification of a property, such as red for color, arial for font family, 12 pt for font-size, and is followed by a semicolon (;).

Property

P {color: red;}

Value

- CSS declarations always ends with a semicolon, and declaration groups are surrounded by curly brackets:

- `p {color:red;text-align:center;}`To make the CSS more readable, you can put one declaration on each line, like this:

- Example

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

### CSS Comments

- A CSS comment begins with `/*`, and ends with `*/`, like this:

- `/*This is a comment*/`

```
p
{
text-align:center;
/*This is another comment*/
color:black;
font-family:arial;
}
```

## ● The id and class Selectors

- In addition to setting a style for a HTML element, CSS allows you to specify your own selectors called "id" and "class".

## ● The id Selector

- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
- The style rule below will be applied to the element with id="para1":

- Example

- ```
#para1
{
text-align:center;
color:red;
}
```

The class Selector

- The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.
- This allows you to set a particular style for any HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a "."
- In the example below, all HTML elements with class="center" will be center-aligned:
- **Example**
- .center
- {
- text-align:center;
- }

- In the example below, all p elements with class="center" will be center-aligned:
- Example
- p.center {text-align:center;}

Three Ways to Insert CSS

External style sheet

Internal style sheet

Inline style

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section:

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

An external style sheet can be written in any text editor. The file should not contain any html tags. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

```
hr {color:red;}  
p {margin-left:20px;}  
body {background-image:url("images/back40.gif");}
```

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style.

```
<head>
<style type="text/css">
hr {color:red;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>
```

Inline Styles

```
<p style="color:red;margin-left:20px">This is a paragraph.</p>
```

Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these properties for the h3 selector:

```
h3  
{  
  color:red;  
  text-align:left;  
  font-size:8pt;  
}
```

And an internal style sheet has these properties for the h3 selector:

```
h3  
{  
  text-align:right;  
  font-size:20pt;  
}
```

If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

```
color:red;  
text-align:right;  
font-size:20pt;
```

The color is inherited from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

- More examples

- ```
body {background-color:#b0c4de;}
h1 {background-color:#6495ed;}
p {background-color:#e0ffff;}
div {background-color:#b0c4de;}
```

- ```
body {background-image:url('paper.gif');}
```

- ```
body
{
background-image:url('gradient2.png');
background-repeat:repeat-x;
}
```

- ```
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
}
```

-

- **body**

{

background-image:url('img_tree.png');

background-repeat:no-repeat;

background-position:top right;

}

in short it can be written as:

- **body {background:#ffffff url('img_tree.png') no-repeat top right;}**

```
body {color:blue;}
  h1 {color:#00ff00;}
  h2 {color:rgb(255,0,0);}
h1 {text-align:center;}
  p.date {text-align:right;}
  p.main {text-align:justify;}
```

A

```
{
text-decoration:none;
}
p.uppercase {text-transform:uppercase;}
  p.lowercase {text-transform:lowercase;}
  p.capitalize {text-transform:capitalize;}
p {
text-indent:50px;}
```

• Font Family

- The font family of a text is set with the font-family property.
- The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font.
- `p{font-family:"Times New Roman", Times, serif;}`

- `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;}`

- Set Font Size With Em
- To avoid the resizing problem with Internet Explorer, many developers use em instead of pixels.
- The em size unit is recommended by the W3C.
- 1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.
- The size can be calculated from pixels to em using this formula: $pixels/16=em$
- Example
- ```
h1 {font-size:2.5em;} /* 40px/16=2.5em */
h2 {font-size:1.875em;} /* 30px/16=1.875em */
p {font-size:0.875em;} /* 14px/16=0.875em */
```

## ● Styling Links

- Links can be style with any CSS property (e.g. color, font-family, background-color).
- Special for links are that they can be styled differently depending on what state they are in.
- The four links states are:
- a:link - a normal, unvisited link
- a:visited - a link the user has visited
- a:hover - a link when the user mouses over it
- a:active - a link the moment it is clicked
- Example
- ```
a:link {color:#FF0000;} /* unvisited link */  
a:visited {color:#00FF00;} /* visited link */  
a:hover {color:#FF00FF;} /* mouse over link */  
a:active {color:#0000FF;} /* selected link */
```
- ```
a:link {text-decoration:none;}
a:visited {text-decoration:none;}
a:hover {text-decoration:underline;}
a:active {text-decoration:underline;}

a:link {background-color:#B2FF99;}
a:visited {background-color:#FFFF85;}
a:hover {background-color:#FF704D;}
a:active {background-color:#FF704D;}
```

- `ul.a {list-style-type: circle;}`  
`ul.b {list-style-type: square;}`

`ol.c {list-style-type: upper-roman;}`

`ol.d {list-style-type: lower-alpha;}`

- `ul`  
`{`  
`list-style-image: url('sqpurple.jpg');`  
`}`

- table, th, td  
{  
border: 1px solid black;  
}

- table, th, td  
{  
border: 1px solid black;  
}

```
td
{
height:50px;
vertical-align:bottom;
}
td
{
padding:15px;
}
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
```

- ```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html>  
<head>  
<style type="text/css">  
div.ex  
{  
width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;  
}  
</style>  
</head>
```

- `<style>`
- `.contentBox`
- `{ display:block; border-width: 1px; border-style: solid; border-color: 000; padding:5px; margin-top:5px; width:200px; height:50px; overflow:scroll`
- `}`
- `</style>`
- `<div class="contentBox">` Why do I call them "CSS Scrollbars"?
- `</div>`



- ```
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
```

- `<html>`
- `<head>`
- `<style type="text/css">`
- `p`
- `{`
- `background-color:yellow;`
- `}`
- `p.padding`
- `{`
- `padding-top:25px;`
- `padding-bottom:25px;`
- `padding-right:50px;`
- `padding-left:50px;`
- `}`
- `</style>`
- `</head>`
  
- `<body>`
- `<p>This is a paragraph with no specified padding.</p>`
- `<p class="padding">This is a paragraph with specified paddings.</p>`
- `</body>`
  
- `</html>`



- `<html>`
- `<head>`
- `<style type="text/css">`
- `h1,h2,p`
- `{`
- `color:green;`
- `}`
- `</style>`
- `</head>`
  
- `<body>`
- `<h1>Hello World!</h1>`
- `<h2>Smaller heading!</h2>`
- `<p>This is a paragraph.</p>`
- `</body>`
- `</html>`

- `<html>`
- `<head>`
- `<style type="text/css">`
- `p`
- `{`
- `color:blue;`
- `text-align:center;`
- `}`
- `.marked p`
- `{`
- `color:white;`
- `}`
- `</style>`
- `</head>`
  
- `<body>`
- `<p>This is a blue, center-aligned paragraph.</p>`
- `<div class="marked">`
- `<p>This p element should not be blue.</p>`
- `</div>`
- `</body>`
- `</html>`

```
<html>
<head>
<style type="text/css">
h1.hidden {visibility:hidden;}
</style>
</head>

<body>
<h1>This is a visible heading</h1>
<h1 class="hidden">This is a hidden heading</h1>
<p>Notice that the hidden heading still takes up space.</p>
</body>
</html>
```

Output:

**This is a visible heading**

Notice that the hidden heading still takes up space.

- `<html>`
- `<head>`
- `<style type="text/css">`
- `img`
- `{`
- `position:absolute;`
- `left:0px;`
- `top:0px;`
- `z-index:-1;`
- `}`
- `</style>`
- `</head>`
  
- `<body>`
- `<h1>This is a heading</h1>`
- ``
- `<p style="color:red">Because the image has a z-index of -1, it will be placed behind the text.</p>`
- `</body>`
- `</html>`

## Output:

**This is a heading**

Because the image has a z-index of -1, it will be placed behind the text

## • Advantages of CSS

### • **CSS saves time**

When most of us first learn HTML, we get taught to set the font face, size, colour, style etc every time it occurs on a page. This means we find ourselves typing (or copying & pasting) the same thing over and over again. With CSS, you only have to specify these details once for any element. CSS will automatically apply the specified styles whenever that element occurs.

### • **Pages load faster**

Less code means faster download times.

### • **Easy maintenance**

To change the style of an element, you only have to make an edit in one place.

### • **Superior styles to HTML**

CSS has a much wider array of attributes than HTML.



- **Disadvantages of CSS:**

- **Browser compatibility**

Browsers have varying levels of compliance with Style Sheets. This means that some Style Sheet features are supported and some aren't. To confuse things more, some browser manufacturers decide to come up with their own proprietary tags.