



Client Side Technologies

css
(Cascade Style Sheets)

What is CSS?

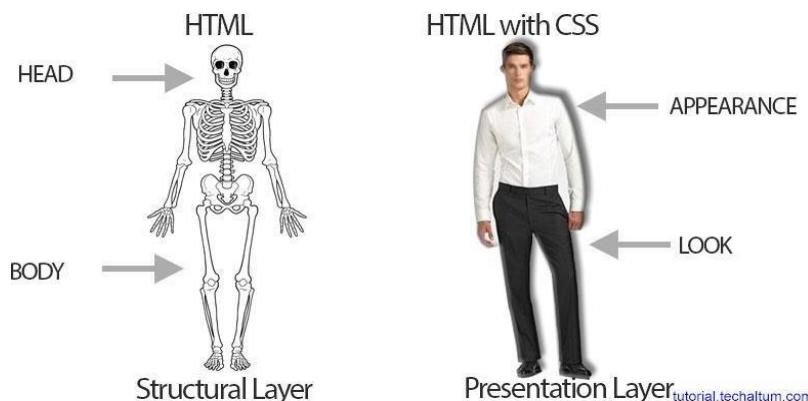
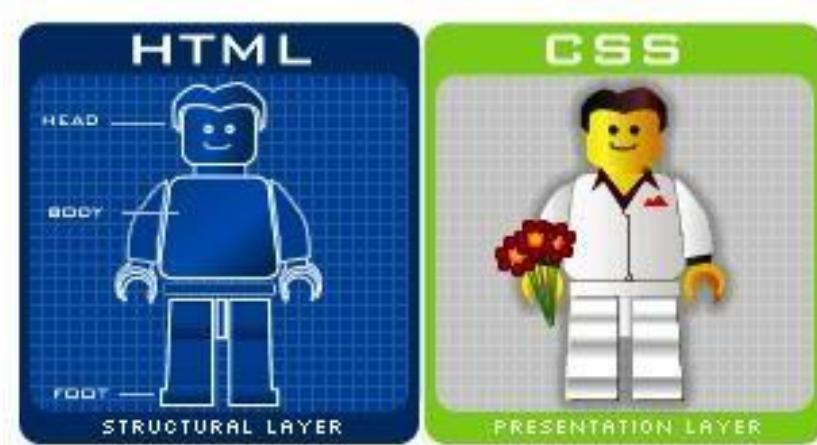


- CSS stands for **Cascading Style Sheets**.
- CSS was developed by the W3C.
- CSS is a stylesheet language used to describe the **presentation** of a document written in a markup language.
- Its most common application is to style web pages written in HTML, XHTML and any kind of XML document.
- Styles define how to display HTML elements (font face, size, color, alignment, ...etc)
- Styles are normally stored in Style Sheets
- The term cascading derives from the fact that multiple style sheets can be applied to the same Web page.

Why use CSS?



- The Separation of Structure and Presentation
- Managing Style at Large Sites
- Improved performance
- Decreased production work
- Rich design and layout



CSS Versions



- Cascading Style Sheets 1 (CSS1)
- Cascading Style Sheets 2 (CSS2 & CSS 2.1)
- Cascading Style Sheets 3(CSS3).

How to Link CSS?



□ CSS can be linked to an HTML document as:

- Embedding a style tag <style>
- Linking to an external stylesheet file
- Importing a stylesheet
- Inline style

Inline style

- ❑ Inline style loses many of the advantages of style sheets by mixing content with presentation.
- ❑ Example:

```
<P STYLE="color: red; font-family: 'Ariel' ">  
    This paragraph is styled in red with the Arial font, if available.  
</P>
```

Embedding a style tag



- An internal/embedded style sheet should be used when a single document has a unique style.
- You define internal styles in the head section by using the `<style>` tag
- An embedded (internal) style sheet should be used when a single document has a unique style.

```
<head>
  <style>
    H1 { color: blue }
    H2 { color: red}
  </style>
</head>
```

H1 header with blue color

H2 header with red color

Linking to an external style sheet file

- ❑ 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" href="style.css"/>
</head>
```

Importing a style sheet



- ❑ Importing allows you to import one style sheet into another.
- ❑ This is slightly different than the link scenario, because you can import style sheets inside a linked style sheet.
- ❑ But if you include an @import in the head of your HTML document, it is written:

```
<STYLE>
```

```
    @import url("styles1.css");
    @import url("style2.css");
    p {color: yellow }
```

```
</STYLE>
```

Cascading Order



- Styles will be applied to HTML in the following order:
 - Browser default
 - External style sheet
 - Internal style sheet
 - Inline style

- When styles conflict, the “nearest” (most recently applied) style wins

Cascading Order - Example



- **External Style sheet**

```
H3  
{  
    color: red;  
    text-align: left;  
    font-size: 8pt  
}
```

- **Internal Style sheet**

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

- **Resultant attributes**

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

CSS Syntax



- The CSS syntax rule is made up of three parts:
 - selector
 - property
 - value
- **selector** is the tag to be affected
- **property** and **value** describe the appearance of that tag
- Style rules are formed as follows:

selector {property: value}

p {font-family: sans serif}

CSS Comments



```
<STYLE TYPE="text/css">
p {
    color: red;
    /* This is a single-line comment */
    text-align: center;
}

/* This is
a multi-line
comment */
</STYLE>
```

Selector

□ Several types of selectors are defined for use when implementing Style Sheets:

- Type Selector
- Class Selector
- ID Selector
- Descendant/Contextual Selector
- Child Selector
- Adjacent sibling selectors
- Attribute selectors

Universal Selector



- The universal (*) selector selects all elements.

- Example:

```
* {  
    background-color: yellow;  
}
```

- The * selector can also select all elements inside another element

```
div * {  
    background-color: yellow;  
}
```

Type Selector



- The **STYLE** attribute can be added to any HTML element.
- Example:

```
H1 {color: blue;}
```

- It selects an element of the HTML document: **P**, **H1**, **BODY**, etc.

Attribute Selector



- Allows you to specify rules that match attributes defined in the source document.
- Syntax :
 - Match when the element sets the "att" attribute, whatever the value of the attribute.
`element[att] { property:value; }`
 - Match when the element's "att" attribute value is exactly "val".
`element [att = "val"] {property: value; }`

Attribute Selector



□ Example:

- Selects “input” element that has the attribute type with value of “button”:

```
Input [type="button"] {background-color: blue;}
```

- Selects any element that has the attribute type with value of “button”:

```
[type="button"] {background-color: blue;}
```

- Selects all elements with a name attribute containing the word "flower"

```
[name~=flower]{background-color: blue;}
```

- Selects every <a> element whose href attribute value begins with "https"

```
a[href^=http]{font-size: 12;}
```

- Selects every <a> element whose href attribute value ends with ".pdf"

```
a[href$=.pdf]{font-size: 16;}
```

IDs

- The ID attribute is used to define a unique style for an element.

- Example:

- In the CSS

```
#id1 {color: red}
```

- In the HTML

```
<div id="id1" >  
    This is the div with the id.  
</div>
```

Classes



- ❑ Classes allow you to define a style which can be applied to multiple elements on your page.
- ❑ Example (1):

- Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

- In the CSS

```
p.righttxt {text-align: right}  
p.centertxt {text-align: center}
```

- In the HTML

```
<p class="righttxt">  
    This paragraph will be right-aligned.  
</p>  
<p class="centertxt">  
    This paragraph will be center-aligned.  
</p>
```

Classes (Cont.)



□ Example (2):

- To apply more than one class per given element:
→ In the CSS

```
p.boldtxt { font-weight: bold}  
p.largetxt { font-size: xx-large}
```

→ In the HTML

```
<p class="boldtxt largetxt">
```

→ This paragraph will be Bold & very large.</p>

- The paragraph above will be styled by the class “bold” AND the class “large”.

Classes (Cont.)



□ Example (3):

→ In the CSS

```
p { font-size: 20px /* apply to all p */  
    .c1{color:red}  
    .c2{color:blue}  
    .c3{ font-weight: bold}
```

→ In the HTML

```
<p>  
    This paragraph will be font size 20.  
</p>  
<p class="c1">  
    This paragraph will be font size 20, and color red.  
</p>  
<p class="c1 c3">  
    This paragraph will be font size 20, and color red, and Bold  
</p>
```

Classes (Cont.)



□ Example (4):

- To apply one class over more than one different HTML element:

→ In the CSS

```
.bold { font-weight: bold }
```

→ In the HTML

```
<p class="bold">  
    This paragraph will be Bold.  
</p>  
<SPAN class="bold">  
    This SPAN will be Bold too.  
</SPAN>
```

- Both the paragraph & the span elements will be styled by the class "bold".

Descendant/Contextual Selector

- Used when we want selectors to match an element that is the descendant (inside) of another element in the document tree (In any level).

```
<H1>
```

This headline is
very
important

```
</H1>
```

- Example:

```
H1 { color: red; }  
span { color: green; }  
H1 span{ color: blue; }
```

This headline is very important

Child Selector



- ❑ The child selector selects all elements that are the **immediate children** of a specified element.
- ❑ A child selector is made up of two or more selectors separated by ">".
- ❑ Example:
 - The following rule sets the style of all P elements that are children of div [that the div is their parent] (Applies only to direct children):

```
div > p {background-color: yellow;}
```

```
<div>
  <p>Paragraph 1 in the div.</p> <!-- Direct child, applies-->
  <p>Paragraph 2 in the div.</p> <!-- Direct child, applies-->
  <span><p>Paragraph 3 in the div.</p></span> <!-- not Child but Descendant -->
</div>
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
```

Adjacent Sibling Selector



- Adjacent sibling selectors have the following syntax: E1 + E2, where E2 is the subject of the selector.
- The selector matches if E1 and E2 share the same parent in the document tree and E1 **immediately precedes** E2.
- Example:
 - The following rule changes the color of an H2 that there's an H2 immediately follows it:

```
H1+H2{color:red ;}
```

```
<body>
  <h1>text</h1>
  <h2> text</h2> will appear in red
</body>
```

element1~element2 Selector



- The element1~element2 selector matches occurrences of element2 that are preceded by element1
- Both elements **must have the same parent**, but element2 does not have to be immediately preceded by element1.
- Example:
 - The following rule changes the color of all H2 that preceded by H2 with the same parent :

```
H1 ~ H2 {color:red }
```

```
<body>
  <h1>text</h1>
  <p>paragraph</p>
  <h2> text</h2> will appear in red
</body>
```

Grouping selector



- ❑ Grouping selectors is done by separating each selector with a comma:

```
H1 { font-family: sans-serif }  
H2 { font-family: sans-serif }  
H3 { font-family: sans-serif }
```

- is equivalent to:

```
H1, H2, H3 { font-family: sans-serif }
```

Pseudo Classes selector



- CSS pseudo-classes are used to add special effects to some selectors.
- A pseudo-class is similar to a class in HTML, but it's not specified explicitly in the markup.
- Syntax:

```
selector:pseudo-class {property:value;}
```

```
selector.class:pseudo-class {property:value;}
```

- Example:

Anchor Pseudo-classes:

```
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.menu:active {color:#0000FF;} /* selected link */
```

CSS Pseudo Classes (cont.)



□ More Example:

Selector	example	Description
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:not()	.class1:not(p)	Selects every element that is not a <p> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:focus	input: focus	Selects the input element which has focus.

Pseudo Elements selector



- ❑ Pseudo-elements match virtual elements that don't exist explicitly in the document tree.

- ❑ In CSS1 and CSS2, pseudo-elements start with a colon (:). In CSS3, pseudo-elements start with a double colon (::), which differentiates them from pseudo-classes.

- ❑ A CSS pseudo-element is used to style specified parts of an element.

Pseudo elements selector (cont.)

□ Examples:

Selector	Example	Example description
::after	p::after	Insert content after every <p> element Example: p::after {content: " - Remember this";} http://www.w3schools.com/cssref/tryit.asp?filename=trycss_sel_after_style
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user http://www.w3schools.com/cssref/tryit.asp?filename=trycss3_selection

Style Precedence in CSS: Specificity, Inheritance



□ Factors that controls which CSS rule applies to a given html element:

○ Specificity Calculations

- Calculate selectors in the CSS rule, knowing that some selectors has more priority than others.
- Importance trumps specificity, When you mark a css property with !important you're overriding specificity rules

○ Inheritance

- Elements inherit styles from their parent container.
- If you set the body tag to use color: red then the text for all elements inside the body will also be red unless otherwise specified.
- Not all CSS properties are inherited, though. For example margins and paddings are non-inherited properties.

Style Precedence in CSS: Specificity, Inheritance - !important



- !important statement can be used to add weight to a declaration.
- !important statement is placed at the end of the declaration, just before the semicolon, and after the value, its invalid if it's located anywhere else.
- It's not a good practice, because it's disrupting the normal flow of the CSS rules.
- Use it when it's very necessary to use, and after all other avenues have been exhausted.
- Examples for when you may need to use it:
 1. You have a global CSS file that sets visual aspects of your site globally.
 2. You use inline styles on elements themselves which is a very bad practice

Style Precedence in CSS: Specificity, Inheritance - !important



□ Example:

- In the below code sample, the element with the id of “example” will have text sized at 14px, due to the addition of !important.
- Without the use of !important, the second block has more specificity (#container followed by #example instead of just #example).

```
#container #example {  
    font-size: 10px;}  
  
#example {  
    font-size: 14px !important;}
```

Style Precedence in CSS: Specificity, Inheritance

○ More about !important and Style Precedence :

- <http://www.vanseodesign.com/css/css-specificity-inheritance-cascade/>
- <http://www.sitepoint.com/web-foundations/cascade/>
- [http://www.w3.org/TR/CSS2/cascade.html/](http://www.w3.org/TR/CSS2/cascade.html)
- <https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>
- http://css.maxdesign.com.au/selectutorial/advanced_cascade.htm
- <http://css-tricks.com/specifcson-css-specificity>
- <http://www.smashingmagazine.com/2010/11/02/the-important-css-declaration-how-and-when-to-use-it/>
- <http://www.sitepoint.com/web-foundations/specifcity/>

Vendor Extension Prefix



Vendor Extension Prefixes

Prefix	Organization
-moz-	Mozilla Foundation
-ms-	Microsoft
-o-	Opera Software
-webkit-	Safari and Chrome

CSS measurement Units



- Physical Measurements
 - **inches (in)**
 - **points (pt)**
- Screen Measurements
 - **pixels (px)**
- Relative Measurements
 - **%**
 - **em**
- **1em = 12pt = 16px = 100%.**

CSS reference



❑ CSS tutorial:

- <http://www.w3schools.com/css/default.asp>
- <http://css-tricks.com>
- <http://www.sitepoint.com>
- <http://css.maxdesign.com.au/selectutorial>

❑ CSS 3 tutorial:

- <http://www.w3schools.com/css3/default.asp>
- <http://www.css3.info/>

❑ CSS Selector reference:

- http://www.w3schools.com/cssref/css_selectors.asp

❑ CSS Properties reference:

- <http://www.w3schools.com/cssref/default.asp>

Self Study



- ❑ CSS cascading and Specificity.
- ❑ CSS3 New properties.
- ❑ CSS3 new properties for HTML5.
- ❑ CSS3 Transition, transformation, and animation.



A cartoon-style question mark character with a smiling face, black outline, and a small black dot at the bottom. It is positioned above the text 'script' and '</script>'.

<script> **</script>**

**<script>document.writeln("Thank
You!")</script>**