

Cascading Style Sheets (CSS)

Introduction

- Styles are rules that modify the behavior of tags in your document.
- A style sheet is a definitions or specifications that tell the browser how to display the various elements on a page.
- Cascading Style Sheets (CSS) form the presentation layer of the user interface.
 - Structure (HTML)
 - Behavior (Client-Side Scripting, e.g Javascript)
 - **Presentation (CSS)**
- Tells the browser agent *how* the element is to be presented to the user.

Cont...

- CSS removes the presentation attributes from the structure allowing
- reusability,
- ease of maintainability,
- interchangeable presentation layer.

The Cascade

- The power of CSS is found in the “cascade” which is the combination of the browser’s default styles, external style sheets, internal (embedded) style sheet and inline styles
- The cascade sets priorities on the individual styles which effects inheritance.

CSS Inheritance

- Allows elements to “inherit” styles from parent elements.
- Helpful in reducing the amount of CSS to set styles for child elements.
- Unless a more specific style is set on a child element, the element looks to the parent element for its styles.
- Each style has a numeric specificity value that is given based on its selector.

Using Style Sheets

- External Style Sheet

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

- Also a "media" descriptor (screen, tv, print, handheld, etc)
- Preferred method.

- Embedded Styles (internal)

```
<style type="text/css">
body {}
</style>
```

- Inline Styles

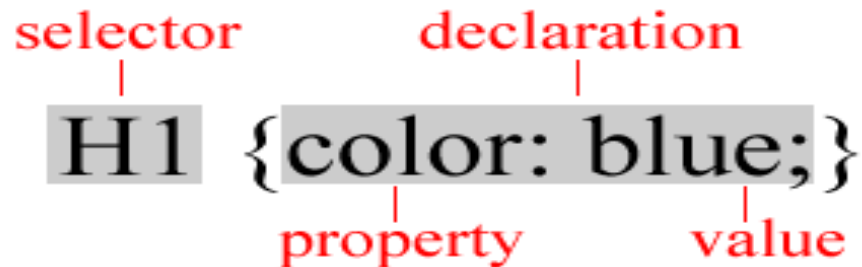
```
<p style="font-size: 12px">Lorem ipsum</p>
```

CSS syntax

- Every CSS statement must have a **selector** and a **declaration**.
- The declaration is one or more properties separated by semicolons.
- Each property has a property name followed by a colon and then the value for that property.
- Sometimes a property can take a number of values. The values in the list should be separated by a comma and a space.
- White space can be used to make your style sheet easier to read and write.

CSS Syntax

```
selector/element  
{  
  property: value;  
}
```



A diagram illustrating the components of a CSS rule. The text `H1 {color: blue;}` is shown. The `H1` is highlighted with a grey background and has a red line pointing to the label `selector` above it. The entire rule `{color: blue;}` is highlighted with a grey background and has a red line pointing to the label `declaration` above it. Within the declaration, `color:` is highlighted with a grey background and has a red line pointing to the label `property` below it. `blue;` is highlighted with a grey background and has a red line pointing to the label `value` below it.

The selector can either be a grouping of elements, an identifier, class, or single HTML element (body, div, h, p, I etc)

Type (Element) Selector

Specify the style(s) for a single HTML element.

```
body {  
margin: 2cm 2cm 2cm 2cm;//top, right, bottom, left  
border-top: 1px solid #ff0;//color, style, width  
}
```

Grouping Elements

Allows you to specify a single style for multiple elements at one time.

```
h1, h2, h3, h4, h5, h6 {  
    font-family: "Trebuchet MS",  
    sans-serif; // arial, sans-serif  
}
```

The Class Selector

```
<p class="intro">This is my introduction  
text</p>
```

```
.intro {  
  font: 12px verdana, sans-serif;  
  margin: 10px;  
}
```

The Identifier Selector

```
<p id="intro"> This is my introduction text</p>
```

```
#intro {  
    border-bottom: 2px dashed #fff;  
}
```

CSS Selectors

- Identifier or class? What's the difference?
 - An identifier is specified only once on a page and has a higher inheritance specificity than a class.
 - A class is reusable as many times as needed in a page.

Advanced CSS Selectors

- Descendant Selector

```
body h1 { }  
#navigation p { }
```

- Adjacent Sibling Selectors

```
p.intro + span { }
```

- Child Selectors

```
div ol > p { }
```

- Universal Selector

```
* { }
```

- Attribute Selectors

```
div[href="http://home.org"]
```

- Pseudo-Class Selectors

```
a:active { }  
#nav:hover { }
```

CSS Fonts

- Font-family
- Font-weight
- Font-style
- Font-size

CSS Units & Colors

- Units
 - %
 - in
 - cm
 - mm
 - **em**
 - **px**
 - pt
- Colors
 - color name (red, etc)
 - rgb(x,x,x)
 - **#rrggbb (HEX)**

CSS Layout

- Margin
- Padding
- Border
- Z-index
- Positioning
- Width
- Height
- Float
- Text-align
- Vertical-align

CSS Text

- Text-indent
- Text-align
- Text-decoration
- Letter-spacing
- Text-transform
- Word-spacing
- White-space

CSS Background

- Background-color
- Background-image
- Background-position
- Background-repeat

CSS Lists

- List-style
- List-style-image
- List-style-position
- List-style-type

CSS Shorthand

- Consolidates many styles into a single declaration.

```
font-family: verdana, sans-serif;  
font-weight: bold;  
font-size: 12px;
```

→ `font: bold 12px verdana, sans-serif;`

```
padding-top: 5px;  
padding-right: 8px;  
padding-bottom: 5px;  
padding-left: 10px;
```

→ `padding: 5px 8px 5px 10px;`

Advantages

Reduced Bandwidth Costs

- One style sheet called and cached
- **faster download**

- **Higher Search Engine Rankings**

- Cleaner code is easier for search engines to index
- Greater density of indexable content

- **Separate text/function from appearance**

- **Allow externally-defined appearance specifications that can apply to whole site**

- **Increase portability**