CSS Notes

Contents

[1 Introduction 1](#_Toc121915235)

[2 CSS Syntax 2](#_Toc121915236)

[3 Selectors 2](#_Toc121915237)

[3.1 Element & Class Selectors 3](#_Toc121915238)

[3.2 Anonymous Class Selector 4](#_Toc121915239)

[3.3 ID Selector 4](#_Toc121915240)

[3.4 Descendant Selector 5](#_Toc121915241)

[3.5 Child Selector 7](#_Toc121915242)

[3.6 Grouping Selectors 7](#_Toc121915243)

[3.7 Specificity – Determining which selector takes precedence. 8](#_Toc121915244)

[3.8 CSS Box Model 8](#_Toc121915245)

[4 Span 8](#_Toc121915246)

[5 Div 9](#_Toc121915247)

[6 Structuring a Page 11](#_Toc121915248)

[7 Common CSS Properties: 11](#_Toc121915249)

[8 Expectations 11](#_Toc121915250)

[Appendix 1 Resources 12](#_Toc121915251)

[Appendix 2 W3Schools CSs Tutorial 12](#_Toc121915252)

[Appendix 3 Using a Table to Structure a Page 13](#_Toc121915253)

# Introduction

In much earlier days of the internet, you could style html elements by specifying *attributes* inside the tags. For example:

<p><font size="2" color="#1c87c9">Blue text</font></p>

<p><font size="3" color="red">Red text, font size increased.</font></p>

<p><font face="arial" color="#8ebf42">Green text, typeface changed.</font></p>

However, now we use a different approach: Cascading Style Sheets (CSS). For example, now, HTML tags support a *style* attribute which is a string composed of property-value pairs, which are similar to the HTML attributes. For example:

<p style="font-size:16px; color:#1c87c9;">Blue text.</p>

<p style="font-size:125%; color:red;">Red text, font size increased.</p>

<p style="font-size:18px; color:#8ebf42; font-family:arial;">Green text, typeface changed.

A number of tags and attributes have been deprecated:

<https://www.w3docs.com/learn-html/deprecated-html-attributes.html>

<https://www.w3docs.com/learn-html/deprecated-html-tags.html>

CSS is the only way HTML should be styled. There are many more CSS properties compared to styling with HTML attributes.

There are three ways to use CSS. The example above is called *inline* (least used, and only sparingly)*.* The other two are are *internal* and *external*. I would guess that most sites use external, with some internal. We will talk later about when to use each technique. [Three Ways to Insert CSS](http://www.w3schools.com/CSS/css_howto.asp)

Most sites/apps you use have very sophisticated CSS styling. Consider Amazon: <https://www.amazon.com/> (Right-click, View page source, then scroll down and click on the CSS).

CSS Development for enterprise-level sites is beyond the scope of this course. However, this page overviews 7 large companies approach to CSS:

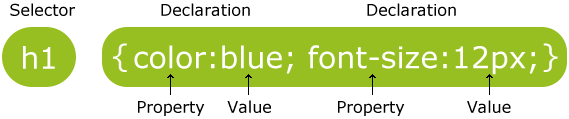
<https://www.webfx.com/blog/web-design/css-development-at-large-sites/>

This site provides analytics and visualizations for style sheets:

<https://cssstats.com/>

# CSS Syntax

CSS Syntax (Source: <http://www.w3schools.com/CSS/default.asp>) for using the *internal* or *external* approach is shown in the figure below. 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 property name and a value, separated by a semicolon.

We will study different types of selectors. To understand the more complex ones, you need to understand a few basic things about [The HTML Document Tree](http://web.simmons.edu/~grabiner/comm244/weekfour/document-tree.html)

# Selectors

A *selector* defines what elements are to be styled. There are a number of different types of selectors which we consider next. An excellent tutorial on selectors:

<http://css.maxdesign.com.au/selectutorial/index.htm>

## Element & Class Selectors

The first two selectors we consider are:

* Element Selector – Selects elements based on the HTML element name. Also called *type* selector.
* Class Selector – Selects specific HTML elements with a specific *class* attribute.

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| body { /\* Element selector \*/  font-family: Calibri,  sans-serif;  }  p { /\* Element selector \*/  margin-left: 35px;  margin-right: 35px;  }  p.important { /\* Class selector \*/  margin-left: 35px;  font-size: larger;  font-weight: bold;  } | <p class="important">This is a paragraph that has been styled with the "important" class</p>  <p>This is a regular paragraph with the default styling</p> | |
| **Resulting Web Page** | |
|  | |

If two selectors select the same HTML element, and if there is a conflict with the rules, then the more specific selector takes precedence. For example, the *margin-left* properties collide in the two rules on the left below when applied to the HTML in the first paragraph on the right. The class rule is more specific, so the left margin is set to 70 on the first paragraph. There is a good bit more to *specificity* considered in a later section.

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| p {  margin-left: 35px;  margin-right: 35px;  }  p.important {  margin-left: 70px;  font-size: larger;  font-weight: bold;  } | <p class="important">This is a paragraph that has been styled with the "important" class</p>  <p>This is a regular paragraph with the default styling</p> | |
| **Resulting Web Page** | |
|  | |

## Anonymous Class Selector

Anonymous Class Selector – Selects any HTML elements with a specific *class* attribute

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| .bigNBold {  font-size: larger;  font-weight: bold;  color: #3333FF  } | <p class="bigNBold">This is a paragraph that has been  styled with the "bigNBold" anonymous class</p>  <ol>  <li>Monday</li>  <li class="bigNBold">Tuesday</li>  <li>Wednesday</li>  </ol> | |
| **Resulting Web Page** | |
|  | |

## ID Selector

ID Selector – Use the *id* attribute of an HTML element to select a specific element. An *id* should be unique so this type of style would apply to exactly one element.

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| #para1 {  margin-left: 35px;  margin-right: 35px;  text-align: center;  color: red;  } | <p id="para1">Cascading Style Sheets (CSS) is ... </p>  <p>CSS is designed to enable the separation ...</p> | |
| **Resulting Web Page** | |
|  | |

## Descendant Selector

Descendant Selector – The descendant selector matches all elements that are descendants of a specified element. The idea with a descendant selector is that you can apply CSS rules to an element depending on what other elements it is nested inside. For instance, suppose that I want all bold tags, <strong> that are nested in a paragraph, <p> to be blue, but, otherwise, I want <strong> to just be a regular (default) bold. In this case, I can define a descendant (contextual) style. In the example below, notice that the style is not applied to the <li>.

Note: According to the HTML 5 specification, the <b> tag should be used as a LAST resort when no other tag is more appropriate. You should use <strong> instead.

1. Example 1

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| p strong {  color: blue;  } | <p>This is an <strong>exciting</strong> lecture.</p>  <ol>  <li>Monday</li>  <li><strong>Tuesday</strong></li>  <li>Wednesday  <p>Today is hump day and we are just trying to make it  to <strong>Friday</strong>!</p>  </li>  <li>Thursday</li>  </ol> |

|  |
| --- |
| **Resulting Web Page** |
|  |

1. Example 2 – Another example is shown below. There, we define a class style, ol.boxed, which puts a dashed line around an ordered list that is classed as "boxed". Next, we define a descendant selector for in list items <li> that appear inside a "boxed" ordered list, providing some space below the list item and a background color. In the HTML, we see that we have one "boxed" ordered list, and one default ordered list. The contextual style is only applied to the "boxed" ordered list.

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| ol.boxed {  border: medium dashed black;  }  ol.boxed li {  margin-bottom: 15px;  background-color: lightgray;  } | <ol class="boxed">  <li>Monday</li>  <li>Tuesday</li>  <li>Wednesday</li>  </ol>  <ol>  <li>Monday</li>  <li>Tuesday</li>  <li>Wednesday</li>  </ol> | |
| **Resulting Web Page** | |
|  | |

1. Example 3 – Another example is shown below, where we define a contextual style with a descendant selector for an ordered list that is inside an ordered list. In other words, the "outer" ordered list contains the list items: Monday, Tuesday (see HTML on right, below). There, we also see that each list item, has an ordered list defined inside it that shows the labels: a. School, b. Homework, etc. This inner ordered list, because of the contextual style, is presented with a background color, some padding, etc. See the [box model](http://www.w3.org/TR/CSS2/box.html) to learn about the difference between padding and margins.

|  |  |  |
| --- | --- | --- |
| **Style Sheet** | **HTML** | |
| ol ol {  padding-top: 10px;  padding-bottom: 10px;  list-style-type : lower-alpha;  background-color: palegoldenrod;  color : blue;  } | <ol>  <li>  Monday  <ol>  <li>School</li>  <li>Homework</li>  </ol>  </li>  <li>  Tuesday  <ol>  <li>Work</li>  <li>Feed Dog</li>  </ol>  </li>  </ol> | |
| **Resulting Web Page** | |
|  | |

## Child Selector

Child Selector – The child selector selects all elements that are the immediate children of a specified element.

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| div {  padding: 25px 25px 25px 25px;  background-color: tan;  }  div > p {  padding: 5px 5px 5px 5px;  background-color: palegoldenrod;  } | <div>  <p>Paragraph 1 in the div.</p>  <p>Paragraph 2 in the div.</p>  <span>  <!-- not Child but Descendant -->  <p>Paragraph 3 in the div.</p>  </span>  </div>  <p>Paragraph 4. Not in a div.</p> |

|  |
| --- |
| **Resulting Web Page** |
|  |

## Grouping Selectors

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| h1, h2 {  text-align: center;  color: red;  }  h1 {  font-style: italic;  } | <h1>Cascading Style Sheets (CSS)</h1>  <h2>Selectors</h2>  <p>Selectors are used to select (find) the HTML element...</p> |

|  |
| --- |
| **Resulting Web Page** |
|  |

## Pseudo-Class Selector

A pseudo-class is used to define a special state of an element. For example, to set a bottom border on every third row in a table:

table { border-collapse: collapse; }

tr:nth-child(3n+3) { border-bottom: 2px solid ; }

The pseudo-class is highlighted. The border-collapse rule is necessary because most browsers render the table cell borders as “separate” meaning the bottom of one row has a border, while the top of the next row has its own border. This rule specifies to collapse the two borders into one.

<https://www.w3schools.com/css/css_pseudo_classes.asp>

<https://css-tricks.com/how-nth-child-works/>

## Specificity – Determining which selector takes precedence.

Specificity – Determining which selector takes precedence.

<https://css-tricks.com/specifics-on-css-specificity/>

<https://specifishity.com/>

<https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>

<https://www.w3schools.com/css/css_specificity.asp>

## CSS Box Model

“All HTML elements can be considered as boxes. The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content.”[[1]](#footnote-1)

<https://www.w3schools.com/CSS/css_boxmodel.asp>

# Span

Span – The HTML <span> tag is an inline tag, e.g. it doesn't break the line (unless forced to by the browser). We can use <span> to markup a word, or small group of words in a special way using a CSS class.

Example:

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| .keyword {  margin: 10px;  padding: 3px;  font-size: larger;  font-weight: bold;  border: 5px solid lime;  background-color: #cc66ff;  color: yellow;  } | <p>  This is a <span class="keyword">keyword</span>  in a paragraph  </p> |

|  |
| --- |
| **Resulting Web Page** |
|  |

# Div

Div – The HTML <div> tag is a block tag that surrounds a block of content and usually starts on a new line (unless absolute positioning is used, which we will not consider).

Example:

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| div{  margin-left:auto;  margin-right:auto;  padding:20px;  border:10px solid #0066FF;  background-color:#00FFFF;  margin-top: 30px;  letter-spacing: 2px;  font-weight:bold;  }  .ex1 {  width:75%;  }  .ex2 {  width:200px;  font-size:larger;  }  .empahsis {  font-style:italic;  color:red;  } | <div class="ex1">  “Who wants to die? Everything struggles to live. Look at that tree growing up there out of that grating. It gets no sun, and water only when it rains. It's growing out of sour earth. And it's strong because its hard struggle to live is making it strong. My children will be strong that way.”  ― <span class="empahsis">Betty Smith, A Tree Grows in Brooklyn</span>  </div>  <div class="ex2">  “If you need something from somebody always give that person a way to hand it to you.”  ― <span class="empahsis">Sue Monk Kidd, The Secret Life of Bees</span>  </div> |

|  |
| --- |
| **Resulting Web Page** |
|  |

Example – The example below has a table nested inside two nested <div>'s. To better understand padding and margins, take a look at the [box model](http://www.w3.org/TR/CSS2/box.html). Note that the <table> attribute "cellspacing" can be set in CSS by using the "border-spacing" property. Also, the <table> attribute "cellpadding" can be set in CSS by providing a style that overrides the <td> tag and sets the "padding" property (set as a contextual style below). Finally, centering the table can be a bit problematic. The technique I used below was to set two table properties in CSS: margin-left and margin-right to the value, "auto".

|  |  |
| --- | --- |
| **Style Sheet** | **HTML** |
| div.outer {      padding: 15px;      margin-right: 15%;      margin-left: 15%;      background-color: #9999FF;  }  div.inner {      padding: 15px;      background-color: #6666FF;      font-weight: bold;      font-size: larger;      color: #FFFFFF;      text-align: center;  }  table.center {      margin-left:auto;      margin-right:auto;      border: thick solid #CCCC99;      border-spacing:10px;  }  table.center td {      border: medium solid #99CC99;      color: #CCCC99;      padding: 10px 10px 10px 10px;  } | <div class="outer">      <div class="inner">          <table class="center">              <tr>                  <td>Monday</td>                  <td>Tuesday</td>              </tr>              <tr>                  <td>School</td>                  <td>Work</td>              </tr>          </table>      </div>  </div> |

|  |
| --- |
| **Resulting Web Page** |
| result |

# Structuring a Page

The preferred way to provide structure for a page is to use div tags. An [example](http://www.w3schools.com/htmL/html_layout.asp) was shown in the w3schools tutorial. We consider this in a later topic. Another way is to use tables, and hide the gridlines and possibly merging rows or columns. An example is shown in an [Appendix](#UsingATableToStructureAPage). However, using tables for structure is definitely not preferred.

Discussion of Tables vs. Div’s:

<https://stackoverflow.com/questions/83073/why-not-use-tables-for-layout-in-html>

<https://stackoverflow.com/questions/6118403/why-is-using-divs-or-spans-tags-better-than-using-a-table-layout>

Using Div’s:

<https://tutorial.techaltum.com/div_based_layout.html>

# Common CSS Properties:

1. [background-color](http://www.w3schools.com/cssref/pr_background-color.asp)
2. [color](http://www.w3schools.com/cssref/pr_text_color.asp), [text-align](http://www.w3schools.com/cssref/pr_text_text-align.asp)
3. [font-family](http://www.w3schools.com/cssref/pr_font_font-family.asp), [font-style](http://www.w3schools.com/cssref/pr_font_font-style.asp), [font-size](http://www.w3schools.com/cssref/pr_font_font-size.asp), [font-weight](http://www.w3schools.com/cssref/pr_font_weight.asp)
4. [list-style-type](http://www.w3schools.com/cssref/pr_list-style-type.asp)
5. [Box Model](http://www.w3schools.com/CSS/css_boxmodel.asp) ([margin](http://www.w3schools.com/cssref/pr_margin.asp), [border](http://www.w3schools.com/cssref/pr_border.asp), [padding](http://www.w3schools.com/cssref/pr_padding.asp))
6. [width](http://www.w3schools.com/cssref/pr_dim_width.asp), [height](http://www.w3schools.com/cssref/pr_dim_height.asp)
7. [vertical-align](http://www.w3schools.com/cssref/pr_pos_vertical-align.asp)
8. [CSS Named Colors](https://www.w3.org/wiki/CSS/Properties/color/keywords)

# Expectations

These are expectations I have of you for testing:

1. Given a description and/or image of a web page, and a collection of styles, write the HTML and styles that create the page.
2. Know how to apply inline, internal, and external styles and their precedence.
3. Know these CSS properties: background-color, color, text-align, font-family, font-style, font-size, font-weight, list-style-type, Box Model (margin, padding, border), width, height, vertical-align
4. Write element, class, id, anonymous, and descendent selectors.

Appendix

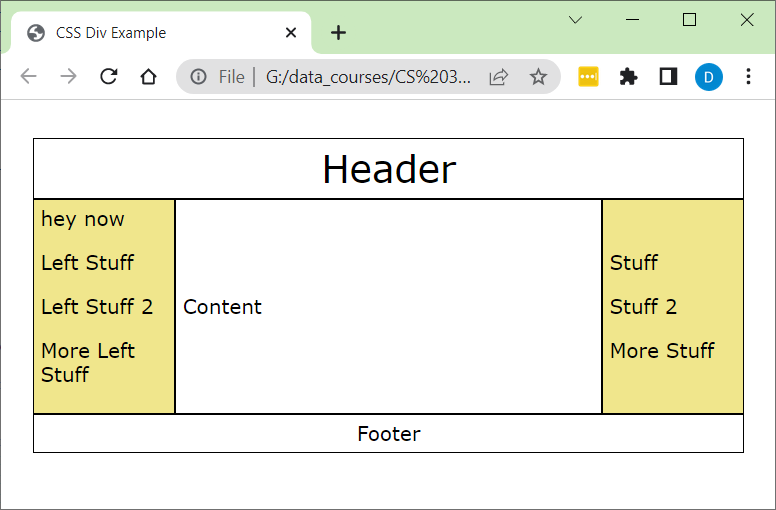
1. Resources
2. [All CSS 3 Properties](http://www.w3schools.com/cssref/default.asp)
3. [All CSS 3 Selectors](http://www.w3schools.com/cssref/css_selectors.asp)
4. Tutorials: [CSSBasics](http://www.cssbasics.com/), [HTML.net](http://www.html.net/tutorials/css/), [w3schools](http://www.w3schools.com/css/default.asp)
5. W3: [Box Model](https://www.w3schools.com/css/css_boxmodel.asp), [CSS 3 Selectors - Specification](http://www.w3.org/TR/css3-selectors/), [Visual Formatting Model](https://www.w3.org/TR/CSS2/visuren.html), [Visual Formatting Details](https://www.w3.org/Style/css2-updates/css2/visudet.html), [contents](http://dev.w3.org/csswg/css2/visudet.html).
6. Color tools: [Creative Bloq](https://www.creativebloq.com/advice/the-best-colour-tools-for-web-designers), web color picker, [Visibone](http://www.visibone.com/colorlab/big.html)
7. CSS Tips and Tricks: [w3.org](https://www.w3.org/Style/Examples/007/), [CSS Tricks](https://css-tricks.com/),
8. [CSSZenGarden](http://csszengarden.com/) - Identical pages, but all styled differently by graphic artists. Seminal page at one time, links are dead mostly.
9. W3Schools CSS Tutorial

W3Schools has a [CSS tutorial](https://www.w3schools.com/css/default.asp) and we covered the highlighted chapters below.

|  |  |
| --- | --- |
| [CSS Tutorial](http://www.w3schools.com/css/default.asp)   1. CSS HOME – Scan 2. CSS Introduction – Read carefully 3. CSS Syntax – Read carefully 4. CSS How To – Read carefully 5. CSS Colors – Scan 6. CSS Backgrounds – First topic only, “Background Color” 7. CSS Borders – Scan 8. CSS Margins – Scan 9. CSS Padding – Scan 10. CSS Height/Width – Scan 11. CSS Box Model – Read carefully 12. CSS Outline – Omit 13. CSS Text – First topic (“Text Color”) and last topic (“Text Indentation”) only. 14. CSS Fonts – Read carefully 15. CSS Icons - Omit 16. CSS Links – Omit 17. CSS Lists – Scan first topic only (“Different List Item Markers”) 18. CSS Tables – Scan | 1. CSS Display – Omit 2. CSS Max-width - Omit 3. CSS Position – Omit 4. CSS Overflow - Omit 5. CSS Float – Omit 6. CSS Inline-block - Omit 7. CSS Align – Omit 8. CSS Combinators – First two topics only (“Descendant Selector” and “Child Selector”) 9. CSS Pseudo-class – Omit 10. CSS Pseudo-element – Omit 11. CSS Opacity – Omit 12. CSS Navigation Bar – Omit 13. CSS Dropdowns – Omit 14. CSS Image Gallery – Omit 15. CSS Image Sprites – Omit 16. CSS Attr Selectors – Omit 17. CSS Forms – Omit 18. CSS Counters – Omit 19. CSS Website Layout – Omit 20. CSS Units - Scan 21. CSS Specificity – Omit |

1. Using a Table to Structure a Page

Consider this page made with a table:



And the corresponding HTML:

<div>

<table cellspacing="0">

<tr>

<td colspan="3" class="header">Header</td>

</tr>

<tr>

<td class="leftSideBar">hey now

<p>Left Stuff</p>

<p>Left Stuff 2</p>

<p>More Left Stuff</p>

</td>

<td class="content">Content

</td>

<td class="rightSideBar">

<p>Stuff</p>

<p>Stuff 2</p>

<p>More Stuff</p>

</td>

</tr>

<tr>

<td colspan="3" class="footer">Footer</td>

</tr>

</table>

</div>

1. <https://www.w3schools.com/CSS/css_boxmodel.asp> [↑](#footnote-ref-1)