

# Cascading Style Sheets: part 2

## CSS Selectors

Today we'll go into more depth with the CSS Selectors, creating some of our own rules (beyond Page Properties). The most common selectors are **Type**, **Class**, **ID** and **Pseudo-class**.

- **Type Selectors:** redefines all instances of a specific HTML tag (i.e. the p tag changes all paragraph tags)
  - this is the selector used most in Page Properties 

```
p {color: green;}
```
- **Class Selectors:** style any element with a class attribute that matches the selector (named the same)
  - a custom style applied to any element you select (in this case, there's no HTML tag named highlight, we'll apply that name) 

```
.highlight {color: green;}
```
- **ID Selectors:** redefines one specific item on your page
  - used for only one element, not used multiple times on a page 



```
#mytag {color: green;}
```
- **Pseudo-class Selectors:** styles an element when it's in a specific state
  - in this example, it will style any a tag (link tag) when the users mouse hovers over that link 

```
a:hover {color: green;}
```

## Type Selectors

```
p {color: green;}
```

Type Selectors are used to define existing HTML tags and, if in an external stylesheet, can effect that tag across an entire website



- in our *styles.css* file, **body**, **td**, **th** are type selectors (**td** & **th** are for table cells and table headers)
  - because these three selectors are grouped together, they share the attached list of rules
  - the **body** tag below has rules that apply to just it (and not **td** & **th**); therefore it has two sets of rules applied to it
- back in our *index.htm* file, in the **CSS Styles Panel**, click the **New CSS Rule** button 
- lets change the default style information for the Paragraph text, starting with the **line-height** (space between the lines of text)
  - because this is an existing tag (**p**) we need to have **Tag** selected, then select **p** from the **Tag drop-down menu**
  - make sure that you're defining this in the *style.css* external stylesheet and click **OK**
  - this brings up the **CSS Rule Definition dialog**
    - make sure that you have the **Type category** selected on the left, then type *18 pixels* in the **Line height** box
    - click on the **Block category** and change Text align to *justify*
    - now click **OK** and watch it make those changes
  - note that the **p** tag has been added to the **CSS Styles panel** (when you have All selected)
  - this has now been changed in all files linked to this stylesheet
- make another change by deleting the rule for **h1,h2,h3,h4,h5,h6** (type); the font reverts to the body tag's Verdana
  - now select the **h1** rule and edit it (click on the Edit Style button ) , change the font back to the **Georgia set**
- now select the **h2** rule in the CSS Styles panel & click on **Add Property**
  - select **Border-Left-Color** from the pop-up menu and set the color to match the type (it's easiest to sample it)
  - now Add a **Border-Left-Width** Property with a value of *11 pixels*

- now Add a **Border-Left-Style** with a value of *solid*
- Add **Padding-Left** with a value of *7 pixels* to give some separation
- Now you have something like a bullet, done through CSS. You also create this in the CSS Rule Definition dialog box

Now you can compare how the CSS Rules display in the Rule Definition dialog, the CSS Styles Panel, and the stylesheet file. All of these are making the changes in the stylesheet file, but are different ways of editing it. This will really help you learn CSS (make sure you save styles.css)

## ID Selectors

Use these when you want to style one specific element on a group of pages (i.e. the footer on every page)

- ID Selectors always start with a #
- often specified in a DIV tag to stylize the contained element
- you should never have more than one element on a page with the same ID(it might work, but in other instances it could crash browsers)
- open *index.htm* in our **Billboards** folder (this is a layout done with table cells instead of CSS regions)
- each cell in the table cell has its own purpose, so we're going to give them individual IDs and styles
- click anywhere in the table and then click on the <table> tag in the bottom left corner of the file window
- now in the **Properties Inspector** name the **Table ID** *tableLayout*
  - notice how the popup menu shows you all available ID Selectors (*selecting them can be good because they're case sensitive*)
  - it's very helpful to name your Selector with a descriptive name (i.e. tableLayout)
- we can now go to the **CSS Styles panel** and click on **New CSS Rule** 
  - we need to have **Advanced** selected & DW has already filled out the ID name for us
  - make sure that we attach this to our current stylesheet, *styles.css* and then click on **OK**
  - click on the **Background category** and select White as the **Background Color** (*#FFFFFF*)
  - set the **Background Image** to *tablebackground.jpg* (browse for it) & set **Repeat** to *no-repeat* and click **OK**
- lets create one more ID Selector rule, this time creating the rule first
  - go to the **CSS Styles panel**, click on **New Rule**  and name this one **#tdNavigation** (td is the tag for table cell)
  - make sure you're attaching it to the **current stylesheet** and click **OK**
  - lets set the **Background Color** to be the same as the images there (by sampling it)
  - go to the **Box category**, under Padding uncheck *Same For All* and set the **Left Padding** to *39* (this ensures that the images will all be flush left and indented) ...now click on **OK**
  - nothing has changed because we've not applied the rule yet
  - click anywhere in the cell with the navigation, click on the <td> tag (lower left corner)
  - now right-click on the td tag and select the ID name *tdNavigation*
  - now the background color has changed and the images sit in a nice, indented location


```
#footer {
    font-size: 10px;
    text-align: left;
}

<div id="footer">This is the site copyright</div>
```

## Class Selectors

Class Selectors are similar to ID Selectors. They are custom styles applied to elements with the class specified (instead of an ID). However, these can be applied to as many elements on a page as you need.

```
.smallprint {font-size: 10px;}  
<p class="smallprint">This is some small text</p>
```

- Class Selectors always start with a "."
- make sure that you're still in our *index.htm* file from the Billboards site
- go to the CSS Styles panel and click on **New Rule** 
  - select **Class** as the Selector Type and type **.smallprint** as the Name (*make sure we're attaching to our current stylesheet*) & click OK
  - in the **Type category**, set the **Size** to **10 pixels**, click **OK**
- now to apply the class, click on the last line of text ("Please let us know what you think") and in the Property Inspector click on the Style dropdown menu and select *smallprint*
- you can see how the class was added to the **p** tag (in the bottom note `<p.smallprint>` or in code `<p class="smallprint">`)
- let's also apply this class to the text in our footer
  - click on the footer text, then click on the DIV tag to the left of the p tag (lower-left corner) and select *smallprint* from the Class pulldown menu in the Properties Inspector
  - this type looks odd where it sits, so lets center it within the DIV
    - go create a **New Rule** in the **CSS Styles panel**
    - we need to do an **Advanced** tag named *div.smallprint*, since we want this applied to div tags that have smallprint applied to them (click on OK)
    - in the **Block category**, set **Text Align** to *center* (click OK)
    - now the type is centered
    - notice how smallprint acts differently on a `<p>` tag than on a `<div>` tag

## Pseudo Class Selectors

These describe the state of an element on your page and are tacked onto the end of other style names. They are only applied when that element is in a certain state.

```
a:link {color: 660000;}  
<a>This is a link</a>
```

The most common Pseudo Class Selectors are those on links (visited, active & hover) we created some of these in the Page Properties last time (on links, or <a> tags).

- take a look at the existing <a> tags (links) in our current stylesheet these are used in our text navigation in the footer
- lets do something a bit more advanced with these styles, just using CSS; open *index2* in our Billboards site
- in DW lets create **New CSS Rule** that is **Advanced** and name it **#tdNavigation a** because it's going to be <a> tags in the Navigation sidebar (*called a descendent selector; all a tags with the tdNavigation sidebar*)
  - in the **Type category** set the **Decoration** to *none* & change the **Color** to #474F49
  - in the **Block category** set **Display** to *block* (this makes the links stack up on top of each other by reserving space for each [nothing can sit to the Left or Right of each element]; as opposed to the default of *Inline*, all in a row)
  - in the **Box category** set the **Padding** to *8 pixels* (keep *Same* for all) and the **Margin** to *10 pixels* (*Padding is space within each element, Margin is space around each element*)
  - in the **Border category** uncheck *Same for all* for all three items set **Top** and **Bottom** each to *solid, 1pixel* and **Color** #CED2B6
- we also want to **create a hover state**, so create another **New CSS Rule** that is **Advanced** and named **#tdNavigation a:hover**
  - in the **Background category** set the **Background color** to #FBFCF9
  - save everything and preview in a browser