

LESSON 7 – Identification and Grouping of Elements Using class and id

In order to apply a special style to a particular element or a particular group of elements you can use class and id to specify properties for selected elements.

I. Grouping elements with class

Assuming that you have two lists of links of different grapes used for white wine and red wine. The HTML code could look like this:

```
<p>Grapes for white wine:</p>
<ul>
<li><a href = "ri.html">Riesling</a></li>
<li><a href = "ch.html">Chardonnay</a></li>
<li><a href = "pb.html">Pinot Blanc</a></li>
</ul>

<p>Grapes for red wine:</p>
<ul>
<li><a href = "cs.html">Cabernet Sauvignon</a></li>
<li><a href = "me.html">Merlot</a></li>
<li><a href = "pn.html">Pinot Noir</a></li>
</ul>
```

If you want to make the white wine links to be yellow, the red wine links to be red and the rest of the existing links on the webpage to stay blue, divide the links into two categories. This is done by assigning a class to each link using the attribute class.

The code below specifies some classes in the example below.

```
<p>Grapes for white wine:</p>
<ul>
<li><a href = "ri.html" class = "whitewine">Riesling</a></li>
<li><a href = "ch.html" class = "whitewine">Chardonnay</a></li>
<li><a href = "pb.html" class = "whitewine">Pinot Blanc</a></li>
</ul>

<p>Grapes for red wine:</p>
<ul>
<li><a href = "cs.html" class = "redwine">Cabernet Sauvignon</a></li>
<li><a href = "me.html" class = "redwine">Merlot</a></li>
<li><a href = "pn.html" class = "redwine">Pinot Noir</a></li>
</ul>
```

You can define special properties for links belonging to whitewine and redwine respectively by doing this:

```
a{
```

```
color:blue;
}
a.whitewine{
color:#FFBB00;
}
a.redwine{
color:#800000;
}
```

II. Identification of element using id

In addition to grouping elements you might need to identify one unique element. This is done by using the attribute id.

What is special about the attribute id is that there cannot be two elements in the same document with the same id. Each id has to be unique. In other cases, you should use the class attribute instead. Now let us take a look at an example of a possible usage of id.

```
<h1>Chapter 1</h1>
...
<h2>Chapter 1.1</h2>
...
<h2>Chapter 1.2</h2>
...
<h1>Chapter 2</h1>
...
<h2>Chapter 2.1</h2>
...
<h2>Chapter 2.2</h2>
...
```

The above could be headings of any document split into chapters or paragraphs. It would be natural to assign an id to each chapter as follows:

```
<h1 id = "ch1">Chapter 1</h1>
...
<h2 id = "ch1-1">Chapter 1.1</h2>
...
<h2 id = "ch1-2">Chapter 1.2</h2>
...
<h1 id = "ch2">Chapter 2</h1>
...
<h2 id = "ch2-1">Chapter 2.1</h2>
...
<h2 id = "ch2-2">Chapter 2.2</h2>
...
```

If you want that the headline for chapter 1.2 must be in red, you can do this accordingly with CSS:

```
#c1-2{
color: red;
```

```
}
```

Copy the code and run to your browser.

LESSON 8 – Grouping of Elements using span and div

I. Grouping with

The element is what you could call a neutral element which does not add anything to the document itself. But with CSS, can be used to add visual features to specific parts of text in your documents.

An example of this could be this Benjamin Franklin quotation

```
<p>Early to bed and early to rise makes a man healthy, wealthy and wise.</p>
```

Assuming that Mr. Franklin sees as the benefits of not sleeping you day away emphasized in bed. For this purpose, we can mark the benefits with . Each span is then added a class, which can be defined in the style sheet as:

```
<p>Early to bed and early to rise makes a man <span class = "benefit">>healthy</span>, <span class = "benefit">>wealthy</span> and <span class = "benefit">>wise</span>.</p>
```

The CSS belonging to it:

```
span.benefit{  
color:red;  
}
```

Run the code to your browser.

II. Grouping with <div>

Whereas is used within a block-level element as seen in the previous example. <div> is used to group one or more block-level elements.

Aside from this difference, the grouping with <div> works in more or less the same way. Let us take a look at an example with two lists of U.S presidents divided into their political affiliations:

```
<div id = "democrats">  
<ul>  
<li>Franklin D. Roosevelt</li>  
<li>Harry S. Trauma</li>  
<li>John F. Kennedy</li>  
<li>Lyndon B. Johnson</li>  
<li>Jimmy Carter</li>  
<li>Bill Clinton</li>  
</ul>  
</div>  
  
<div id = "republicans">  
<ul>  
<li>Dwight D. Eisenhower</li>
```

```
<li>Richard Nixon</li>
<li>Gerald Ford </li>
<li>Ronald Reagan</li>
<li>George Bush</li>
<li>George W. Bush</li>
</ul>
</div>
```

In the style sheet, you can utilize the grouping in the exact same way as above.

```
#democrats{
background:blue;
}

#republicans{
background:red;
}
```

LESSON 9 – CSS Lists

With CSS you can customize the lists that can be made with HTML. However there are certain differences on how browsers display CSS Lists. Netscape browsers only let you add the list CSS to tags not just any tags.

While Internet Explorer's support of CSS with relation to lists is only fully supported for browsers on the Windows platform.

In cases where the browsers does not support it, it just shows as a normal list.

I. List Properties

Property	Values
list-style-type	disc circle square decimal lower-roman upper-roman lower-alpha upper-alpha none
list-style-image	none url(<url>)
list-style-position	Outside inside
list-style	<list-style-type> <list-style-position> <list-style-image>

II. Defining styles for links

As mentioned in the above table, there are four unique selectors with respect to lists. The fourth selector, list-style, is an overall selector that let you define all list related styles at once.

The three basic selectors are:

- list-style-type
defines the look of the bullets used in your list.
- list-style-image
lets you use a custom graphic for bullets.
- list-style-position
often the text in a list is longer than one line
 - ✓ **list-style-position: outer** lets the second line align with the first line. That is the bullet is to the left of both lines.
 - ✓ **list-style-position: inner** lets the second line align with the bullet.

III. Assigning several properties at once

Instead of using different selectors for each list-style you can specify them all at once using the list-style property. Below is an example:

```
<html>
<head>
<style type = "text/css">
L1.list1 {list-style: circle outside; color: green; }
L1.list2 {list-style: square inside; color: blue; }
.blacktext{ color: black; }
</style>
</head>

<body>
<ul>
<li class = "list1"><span class = "blacktext">This is one black line</span></li>
<li class = "list1">This is another line that is much longer than the first. But it isn't a black line since we
did not specify a style for the text that goes here other than the style we defined for the list.</li>
</ul>
<br>
<br>
<ul>
<li class = "list2"><span class = "blacktext">This is one black line</span></li>
<li class = "list2">This is another line that is much longer than the first. But it isn't a black line since we
did not specify a style for the text that goes here other than the style we defined for the list.</li>
</ul>
</body>
</html>
```

Run the code above to your browser.

ExerciseCSS4

I. Objectives. At the end of the exercise, the students are expected to :

1. Apply basic CSS formatting that they have learned from the previous activities
 2. Apply the use of CSS classes/id's, span and div
 3. Apply CSS list selectors and properties and
 4. Use EXTERNAL style sheet
- II. Concepts to be applied
1. External CSS
 2. CSS basic formatting tags
 3. CSS classes/id's, span and div
 4. CSS list selectors and properties
- III. Instructions
1. Open Exercise2b.html. Save it as ExerciseCSS4.html inside CSS folder.
 2. Create a file called ListCSS.css and save it inside CSS folder.
 3. In the external stylesheet create a way to modify ExerciseCSS4.html such that
 - a. The header styles are defined in CSS
 - b. The list type in HTML code will be replaced with the equivalent CSS code
 - c. The font color of the lists will have their own unique color. Define 6 different colors and assign each to each list.
 - d. Each list type and color shall be defined by using CSS classes/id's and span / div

Winter and Snow Camping Safety for Scout Leaders

Topics Covered

Session 1 - The Fun Stuff - In town, Saturday morning

1. Goals
2. Introductory remarks and outline of this course
3. Round Robin
 - o Clothing
 - o Sleeping materials:
 - Sleeping bags,
 - Sleeping pads,
 - other sleeping gear
 - o Tents and other camp
4. Trip planning
5. Winter driving

Session 2 - Serious Concerns - In town, Saturday afternoon

1. Further Administrative Details
2. Overview of problems unique to winter and snow activities
3. Round Robin
 - o Minimum Impact Camping and Sanitation
 - o Snow travel and intro to
 - Skis
 - Snowshoes
 - sleds
 - o Weather - some considerations on how winter differs from summer trips
 - o Avalanche dangers
4. Navigating - Special considerations in land navigation in winter conditions
5. Dealing with Winter Emergencies
6. Other Problems to Consider and Wrap-up of session

IV. Checklist for the instructor

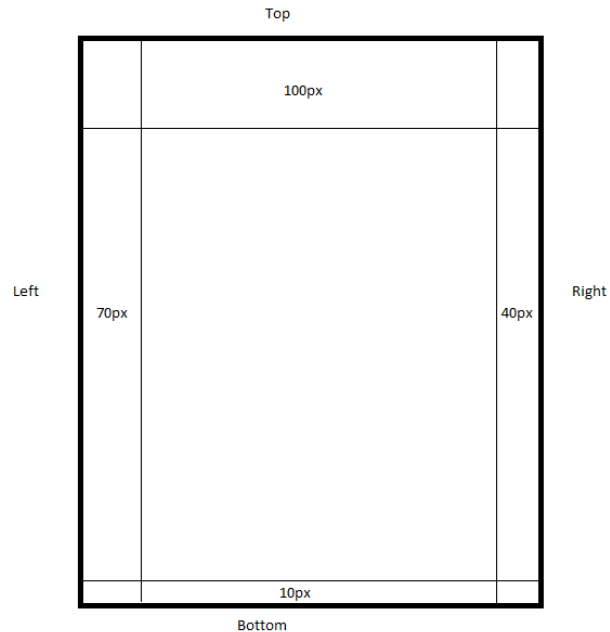
Activity	Yes (2 points)	No (0 point)
A. External Stylesheet		
1. Did the student create external CSS file named ListCSS.css?		
2. Does the external stylesheet contain CSS styles for: <ul style="list-style-type: none">○ Headers○ List type○ Ids/classes		
B. HTML File		
1. Is the external style sheet linked to the HTML file?		
2. Are the colors for each list type reflected? <ul style="list-style-type: none">○ Are there 6 different colors?○ Are the lists properly colored?		
3. Is the file saved as ExerciseCSS4A.html and save it to CSS folder?		

LESSON 10 – Margin and Padding

Setting the margin in an element

An element has four sides: right, left, top, bottom. The margin is the distance from each side to the neighboring element (or the borders of the document).

As the first example, you will look at how you define margins for the document itself i.e. for the element <body>. The illustration below shows how we want the margins in our pages to be.



The CSS code for this would look as follow:

```
body{  
margin-top:100px;  
margin-right:40px;  
margin-left:70px;  
margin-bottom:10px;  
}
```

Or you could do this

```
body{  
margin: 100px 40px 10px 70px;  
}
```

Setting the padding in an element

Padding can also be understood as “filling”. This makes sense as padding does not affect the distance of the element to other elements but only defines the inner distance between the border and the content of the element.

The usage of padding can be illustrated by looking at a simple example where all headlines have background colors.

```
h1{  
background:yellow;  
}  
h2{  
background:orange;
```



```
}
```

By defining padding for the headlines, you change how much filling there will be around the text in each headline.

```
h1{
background:yellow;
padding:20px 20px 20px 80px;
}
h2{
background:orange;
padding-left:120px;
padding-right:50px;
padding-top:20px;
padding-bottom:10px;
}
```

ExerciseCSS5a and Exercise CSS5b

- I. Objectives. At the end of the exercise, the students are expected to:
 1. Apply basic CSS formatting that they have learned from the previous activities.
 2. Apply the use of CSS margins and
 3. Use EXTERNAL style sheet
- II. Concepts to be applies
 1. EXTERNAL CSS
 2. CSS basic formatting tags
 3. CSS margins
- III. Instructions
 1. Open 2 HTML/CSS activities. Save each as ExerciseCSS5a.html and ExerciseCSS5b.html. Save the file in the CSS folder.
 2. Create an external style sheet called MarginCSS.css in CSS folder and set the margins as follow:
 - Top: 100 pixels
 - Right: 40 pixels
 - Bottom: 10 pixels
 - Left: 70 pixels
 3. Using MarginCSS.css modify the margins of ExerciseCSS5a.html and Exercise5b.html
- IV. Checklist for the instructor

Activity	Yes (2 points)	No (0 point)
A. External Stylesheet		
1. Did the student create an external CSS file named MarginCSS.css?		
2. Does the external stylesheet contain CSS styles for: <ul style="list-style-type: none">○ Top: 100 pixels		

<ul style="list-style-type: none"> ○ Right: 40 pixels ○ Bottom: 10 pixels ○ Left: 70 pixels 		
B. ExerciseCSS5a.html and ExerciseCSS5b.html		
1. Is the external style sheet linked to the two HTML file?		
2. Are the margins reflected in the two HTML file?		
3. Are the files saved as ExerciseCSS5a.html and ExerciseCSS5b.html? Does the file located in the CSS folder?		

LESSON 11 – CSS Layers

With CSS, it is possible to work with layers; pieces of HTML that are placed on top of the regular page with pixel precision.

I. Layer Basics

Copy the code below and run to your browser.

LAYER 1 ON TOP:

```
<div style = "position: absolute; font-size:50px; z-index:2;">LAYER 1</div>
<div style = "position:relative; top:-50; left:5; color:red; font-size:80px; z-index:1">Layer 2</div>
```

LAYER 2 ON TOP:

```
<div style = "position: absolute; font-size:50px; z-index:3;">LAYER 1</div>
<div style = "position:relative; top:-50; left:5; color:red; font-size:80px; z-index:4">Layer 2</div>
```

To create a layer all you need to do is assign the position attribute to your style. The position can be either **absolute** or **relative**.

The position itself is defined with the **top** and **left** properties.

Finally, which layer is on top is defined with the z-index attributes.

II. Relative Versus Absolute Positioning

You can either position your layer calculated from the upper left corner (absolute) or calculate from the position where the layer itself is inserted (relative).

- position: absolute

If you define the position to be absolute it will be calculated from the upper left corner of the page, unless the layer is defined inside another layer, in which case it will be calculated from the upper left corner of the parent layer.

- position: relative

If you define the position to be relative it will be relative to the position of the tag that carries the style.

That is, if you add a relatively positioned layer in the middle of the page, then the position will be calculated from that exact spot in the middle of your page where it was added.

A. Defining the position

While the position property indicates the out spring of our coordinates system, the left and top properties defines the exact position of our layer.

You can enter both positive and negative values for these properties, thus it is possible to place content higher up and further to the left on the page than the logical position in the HTML code where the layer itself is defined.

In other words, at the bottom of your HTML code you can enter the code for a layer that is positioned at the top of the resulting page.

Both left and top properties can be dynamically changed with JavaScript.

This means that it is possible to move things around on the screen even after the page has finished loading.

In fact this technique can be (and has been) used to create entire games. Other uses might be menus that pop out when a mouse-over is detected on a link. The possibilities are endless.

B. Position in the stack – The z-index

Picture a game of 52 cards. If the ace of spades was at the bottom we'd say it had z-index:1. If the jack of spades was at the top we'd say z-index:52.

Try looking at the code example at the top of this page again, and see how we used the z-index to put LAYER 1 on top in the first example, while we had LAYER 2 on top in the second example. Very interesting possibilities arise from the fact that the z-index can be dynamically changed with JavaScript.

You could create several "pages" on top of each other, all on the same page. When the user clicks a link it will simply move the layer with the desired info on top rather than load a new page. The techniques to create effects like that goes beyond the scope of pure CSS however, so for now we will just refer to DHTML (Dynamic HTML) a mix between JavaScript and CSS for further explorations into that area.

III. Visible Versus Hidden Layers

4. A final property is the visibility property that will allow you to create invisible layers. Why would anyone want to create an invisible layer? Well, imagine the possibilities it gives for adding pop-up menus and other cool effects on your pages.
5. With dynamic HTML it is possible to change the visibility of a layer according to certain events. The most common use of this is to create menus that pop out (like the sub menus in the START menu on windows). The trick behind these menus is to create all submenus as invisible layers. Then, when a mouse-over is detected on a link the according layer becomes visible. (Sounds pretty easy, actually is pretty easy except when tried on Netscape browsers that seem to have only a vague idea of the logic behind CSS layers).
6. Valid values for the visibility property are: **visible** and **hidden**.
7. This example shows how to create a hidden and visible layer.

`<div style = "position:relative; visibility:hidden;">Hidden HELLO!!!</div>`

```
<div style = "position:relative; visibility:visible;">Visible HELLO!!!</div>
```

IV. Uses of Layers

Layers can be used to create the following:

- Flying elements / banners on the page
- Games where you move an object around
- Menus that pop out when triggered
- Menus that become visible when triggered

ExerciseCSS6

V. Objectives. At the end of the exercise, the students are expected to:

1. Apply basic CSS formatting that they have learned from the previous activities.
2. Create columns and boxes using div
3. Format columns and boxes using classes and ids and div properties
4. User internal style sheet

VI. Concepts to be applies

1. INTERNAL CSS
2. CSS basic formatting tags
3. DIV properties especially float and clear
4. CSS classes and IDs
5. CSS border properties
6. Concepts in designing

VII. Instructions

1. Open your ExerciseCSS3.html. Save this as ExerciseCSS6.html in your CSS folder.
2. Remove <p> and <pre> tags in the HTML code
3. Using <div> create 3 columns of the article that are horizontally aligned with each other.
 - a. Divide the article into 3 sections to do this.
 - b. Place the formatting tags as internal CSS
4. Each column should be enclosed in a box. Each box should have its own border style and background color.

VIII. Checklist for the instructor

Activity	Yes (2 points)	No (0 point)
A. Internal Stylesheet		
1. Did the student put the formatting tags inside the HTML Code?		
2. Did the student use classes or Ids to format the boxes?		
B. Article		
1. Are there 3 column boxes?		

2. Are the boxes horizontally aligned along each other?		
3. Is each box formatted in terms of <ul style="list-style-type: none">○ Border style○ Background color		