**IS 360  
CSS LAB 2  
EKEDAHL**

In this lab, you will work with Cascading Style Sheets. You will perform the following tasks:

* Work with CSS positioning.
* Continue to format text.

For all of the problems, I suggest that you use strict DTDs for all of the style sheets:  
  
<!DOCTYPE html >

<html xmlns="http://www.w3.org/1999/xhtml">  
</html>

**In this exercise, you will work with fixed positioning to display two overlapping boxes:**

1. Create an empty Web site.
2. Add a page named **Fixed.htm** to the site.
3. Using classes, declare two internal styles to describe two fixed-positioned boxes (position:fixed;). Name the two classes **FixedTop** and **FixedBottom**. Each box should be 250 pixels wide and 100 pixels tall. You will need to set the top and left properties so that the two boxes are offset as shown in the following figure. Try to modify the borders so that the boxes resemble the following. Set the colors as you see fit.
4. Create two <div> tags to use the styles you created above. The <div> tags should be declared as follows:

<div class="FixedTop">

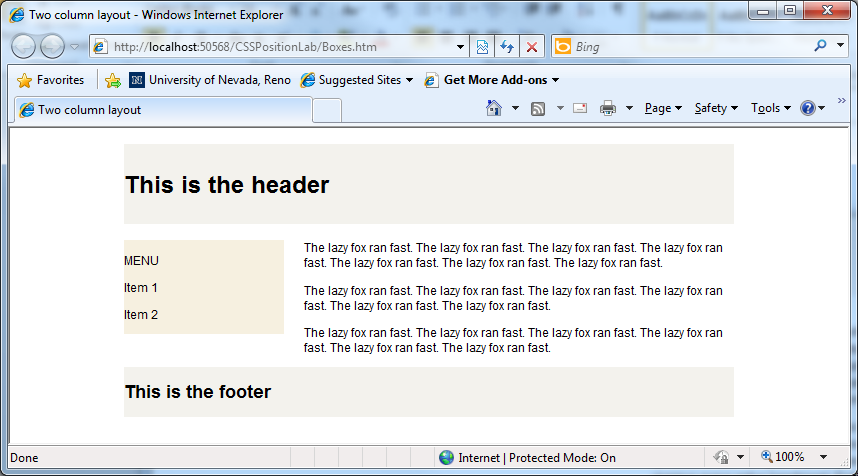
</div>

<div class="FixedBottom">

</div>

1. You might need to set the z-order so that the boxes overlay correctly.

**In this exercise, you will create a layout made up of a header, footer and a body made up of two columns. The page you are trying to create should look like the following image:**



Note that there are several ways to solve this problem. My suggested solution is but one of them. Note that in my solution, I used IDs but could have just as well used classes.

1. Create a blank HTML5 page named **boxes.htm.**
2. Create the following style for the body.

body

{

font: 76% arial, sans-serif;

}

1. Create styles for the <h1> and <h2> tags as follows. These tags will ultimately be used for the head in the header and footer. Note that by setting the two properties, the box is quite tall. Again, there are other ways to do this.

h1

{

height:80px;

line-height:80px;

}

h2

{

margin-top:10px;

height:50px;

line-height:50px;

}

1. Create a style named **wrapper**. This style will be applied to an outermost <div> tag and defines the width of the box outermost containing box. Make the width of the box 610 pixels.
2. Next, create the box that will display the header. This box should have a background color of #F3F2ED. Call this style **header.**
3. Now, create the HTML that will display the outer box and header as follows. Note that because the header appears inside of the wrapper, it’s width is constrained by the parent box.

<div class="wrapper">

<div class="header">

<h1>This is the header</h1>

</div>

</div>

1. Next, create the styles that will display the two columns. Name these styles **side-a** and **side-b**. The style named side-a should “float left”, have a width of 160 pixels, and a background color of #F6F0E0. Side-b should also float left and have a width of 420 pixels.
2. Now, modify the HTML you created to display the two side. It should look like the following:

<div class="wrapper">

<div class ="header">

<h1>This is the header</h1>

</div>

<div class ="container">

<div class ="side-a">

<p>MENU</p>

<p>Item 1</p>

<p>Item 2</p>

</div>

<div class ="side-b">

<p style="display:inline">The lazy fox ran fast.

The lazy fox ran fast. The lazy fox ran fast.

The lazy fox ran fast. The lazy fox ran fast.</p>

<p>The lazy fox ran fast. The lazy fox ran fast.

The lazy fox ran fast. The lazy fox ran fast.

</p>

<p>The lazy fox ran fast. The lazy fox ran fast.

The lazy fox ran fast. The lazy fox ran fast.

</p>

</div>

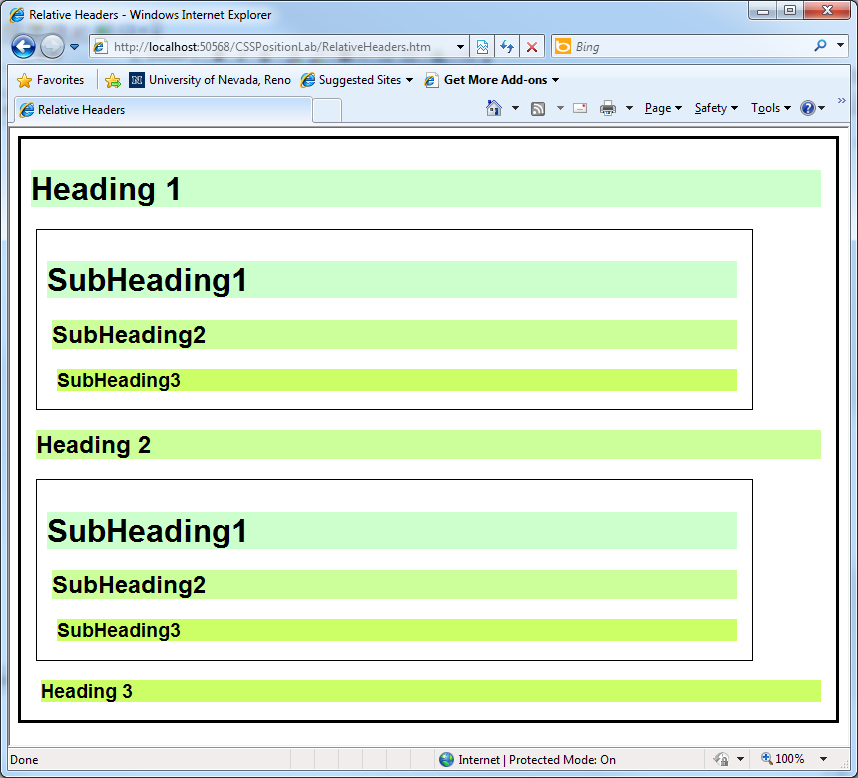
</div>  
 </div>

1. Now, it’s up to you to create the footer. Clear the float with the following: clear:both;

**In this exercise, you will work with scrolling windows:**

1. Visit the following URL: <http://w3schools.com/css/pr_pos_overflow.asp>.
2. Using the information from this Web page, create a blank page named **scroll.htm**
3. Create a class named **scroll** that is 200 pixels tall and 400 pixels wide. The content should scroll when it will not fit in the box (**overflow:scroll**)
4. Create a paragraph and some content to test your work.

**In this exercise, you will work with relative positioning. The purpose to is create nested boxes, each having the same indenting. The following figure shows the completed output.**



1. Create a blank XHTML page named **RelativeHeaders.htm**.
2. Using relative positioning, create styles for the <h1>, <h2>, and <h3> tags as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | <h1> | <h2> | <h3> |
| position | relative | relative | relative |
| background-color | #ccffcc | #ccff99 | #ccff66 |
| margin-left | 10px | 15px | 20px |

1. Next, create two styles named **wrapper** and **level2Wrapper** such that level2Wrapper is positioned relative to its container (which will be wrapper). Set the margins and padding such that the output appears similar to the output shown in the preceding figure. The width of the level2Wrapper should be 700px and the width of the wrapper should be 800px
2. Use the following HTML to test your work. Note that because the headings and <div> tags are relatively positioned, they behave differently based the container.

<body>

<div class="wrapper">

<h1>Heading 1</h1>

<div class="level2Wrapper">

<h1>SubHeading1</h1>

<h2>SubHeading2</h2>

<h3>SubHeading3</h3>

</div>

<h2>Heading 2</h2>

<div class="level2Wrapper">

<h1>SubHeading1</h1>

<h2>SubHeading2</h2>

<h3>SubHeading3</h3>

</div>

<h3>Heading 3</h3>

</div>

<!—Pay attention to how these boxes are positioned differently.

<h1>Foo</h1>

<div class="level2Wrapper">

<h1>SubHeading1</h1>

<h2>SubHeading2</h2>

<h3>SubHeading3</h3>

</div>

</body>>