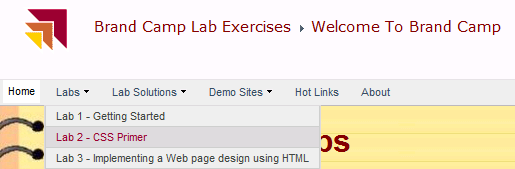
## CSS Primer for SharePoint Sites

**Lab Time**: 45 minutes

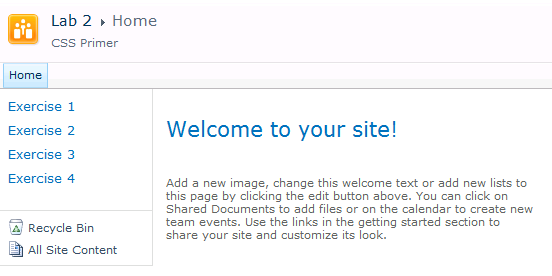
### Exercise 1: Working with SharePoint Designer

In this exercise you will get experience customizing the CSS rules behind pages using the SharePoint Designer.

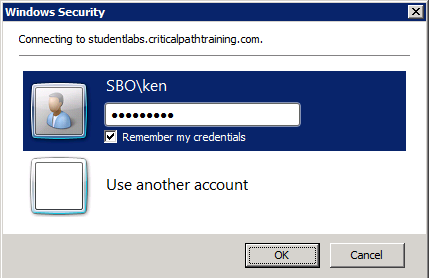
1. Using the browser, navigate to the Collaboration Site Collection located at **http://[[COLLAB-SITE]]**. Next, navigate to the child site for **Lab 2** by using the global navigation menu of the top-level site.



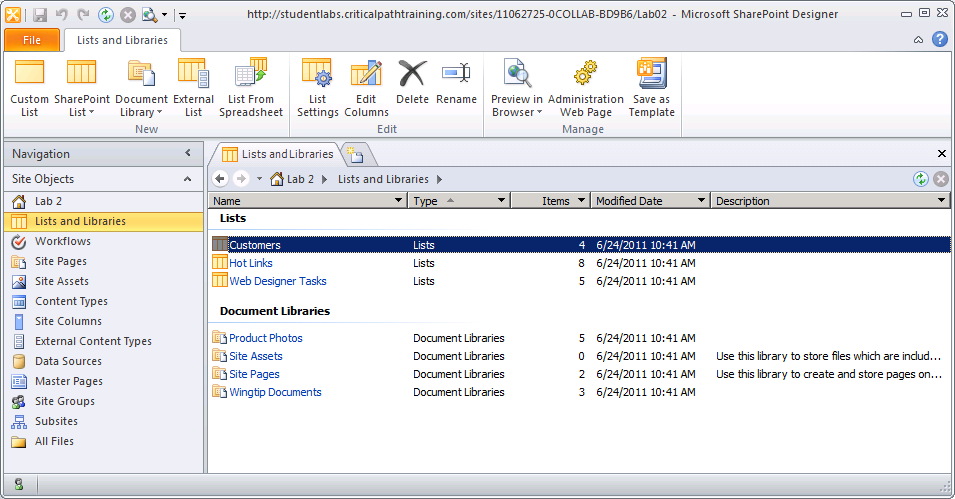
1. You should now be at the home page in the Lab 2 site.



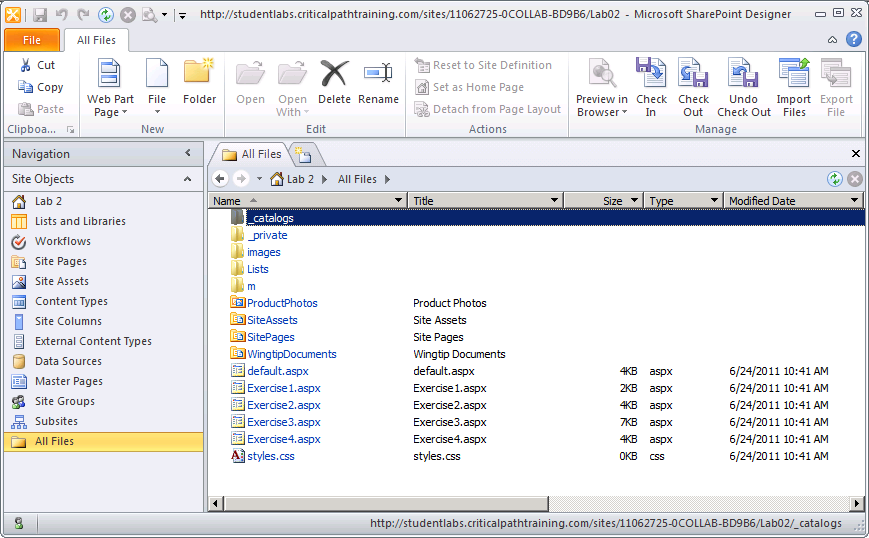
1. Launch SharePoint Designer 2010 and open the site at **http://[[COLLAB-SITE]]/Lab02**. If prompted to log in, enter the site collection owner as **[[AD\_DOMAIN]]\[[USERNAME]]**. For example, if you were supplied the user account login credentials for Ken Sanchez with a user account name **ken** and a domain name **SBO,** then theaccount name for logging in should be entered as **SBO\ken**. The password for all accounts should be set to **Password1**.



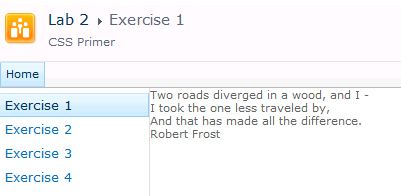
1. Once the site has opened in the SharePoint Designer, explore the left-hand **Site Objects** section to see the available collections of site items such as **Lists and Libraries**, **Site Pages** and **Master Pages**. Take a moment to see what items are available inside these collections.



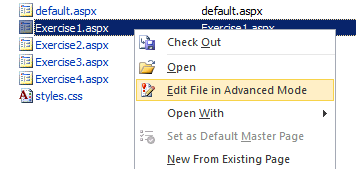
1. Click the **All Files** link at the bottom of the **Site Objects** section. This will allow you to see the virtual file system for the current site. You should be able to see the files for pages named **Exercise1.aspx**, **Exercise2.aspx**, **Exercise3.aspx** and **Exercise4.aspx** and a style sheet named **styles.css**. These are the files you will begin working with in the next exercise.



1. At this point, you should have the **Lab 2** site open in both the browser as well as the SharePoint Designer. Leave both open as you will be moving back and forth between them for all the remaining exercises of this lab.
2. In the browser, navigate to the page for **Exercise 1** by clicking the navigation link in the Quick launch. Currently, the content on this page contains a poem by Robert Frost but lacks any CSS rules to add more appealing styles.



1. Your job is to make this page look more visually appealing by adding CSS. You are going to perform the work with SharePoint Designer and make a direct customization to the site page named **Exercise1.aspx**.
2. Go back to the SharePoint Designer. Click on the **All Files** view in the **Site Objects** view to see the pages at the root of the site. Open the file for **Exercise1.aspx** by right-clicking it and clicking **Edit File in Advanced Mode**.



1. Inspect the contents of the page. You should be able to see that there is a content placeholder control with a **ContentPlaceHolderId** of **PlaceHolderMain** that contains the HTML layout with the page content. Note that there is also a content placeholder control with a **ContentPlaceHolderId** of **PlaceHolderAdditionalPageHead** where you will add your styles in this exercise. Also take a moment to understand the structure of the div element with the class named **poem**. It contains two inner div elements with classes named **poem\_content** and **poem\_author**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

<!-- add styles or links to CSS files here -->

</asp:Content>

<asp:Content ContentPlaceHolderId="PlaceHolderMain" runat="server">

<div class="poem">

<div class="poem\_content">

Two roads diverged in a wood, and I -<br/>

I took the one less traveled by,<br/>

And that has made all the difference.

</div>

<div class="poem\_author">Robert Frost</div>

</div>

</asp:Content>

1. Update this code by adding the following style element with a CSS rule to add padding to the div with the ID of **MSO\_ContentTable**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

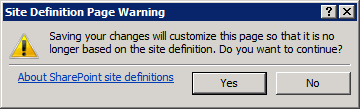
<style type="text/css" >

#MSO\_ContentTable { padding: 8px; }

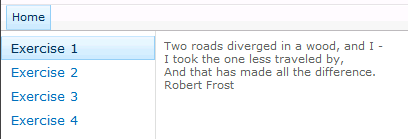
</style>

</asp:Content>

1. Save the page. When you save your work for the first time, SharePoint Designer will prompt you with the following dialog asking if it is OK to customize (i.e. unghost) the page. Click **Yes**.



1. Refresh the browser and you should now see padding between the poem and the left border of the content area.



1. Now add the following CSS to stylize the poem content.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

<style type="text/css" >

#MSO\_ContentTable {

padding: 8px;

}

.poem {

margin-top: 20px;

margin-left: 20px;

padding: 12px;

background-color: #EEEEEE;

border: solid 1px #CCCCCC;

width: 480px;

}

.poem\_content {

font-size: 18pt;

line-height: 22pt;

color: #333333;

display: block;

}

.poem\_author {

margin-top: 8px;

font-size: 14pt;

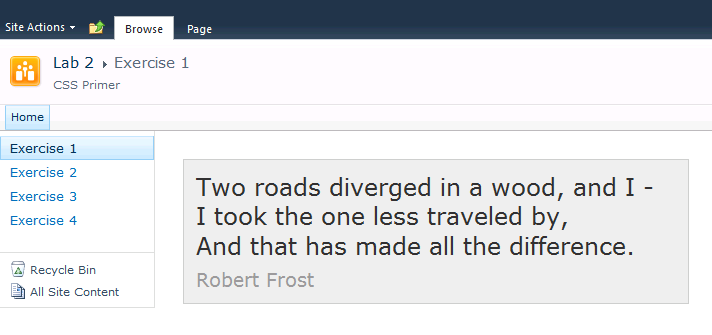
color: #999999;

}

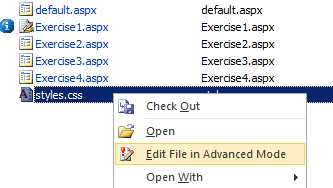
</style>

</asp:Content>

1. When you are done, the page should now look like the one shown in the following screenshot. Feel free to make creative changes to the CSS shown above if you want to further enhance the style.



1. As you saw earlier in this exercise, the **Lab 2** site already includes a CSS file at the root of the site named **styles.css**. Open this file in SharePoint Designer in Advanced Edit Mode.



1. Copy all the styles rules from **Exercise1.aspx** into **styles.css**. Save your changes to **styles.css**.
2. Inside the **PlaceHolderAdditionalPageHead** placeholder in **Exercise1.aspx**, remove the style rules and replace them with a link to the style sheet named **styles.css**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

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

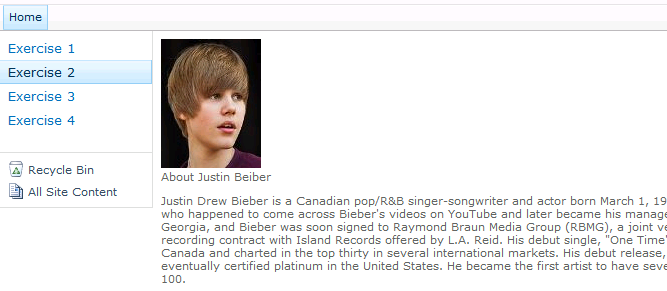
</asp:Content>

1. Save all your work and test through the browser. You should see the page for **Exercise1.aspx** looks the same as it did before through the browser. However, the difference is that the CSS rules are now being read from a separate CSS file that can be used across pages.

### Exercise 2: Floating a div element

In this exercise you will use CSS to float a div with a picture.

1. In this exercise you will work on the page named **Exercise2.aspx**. In the browser, examine the page by clicking the **Exercise 2** link in the Quick launch bar. As you can see this page needs some work.



1. In SharePoint Designer, open the page named **Exercise2.aspx** in advanced edit mode and examine the contents of the page. As you can see, this page is already linked to the style sheet named **styles.css**. This means you can make changes to the look and feel of **Exercise2.aspx** by modifying **styles.css**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

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

</asp:Content>

1. Examine the HTML content inside of **PlaceHolderMain**. Note that there is a top-level div element with a class of **bio**. There are three more inner div elements with classes named **bio\_picture**, **bio\_title** and **bio\_content**.

<div class="bio">

<div class="bio\_picture"><img src="images/Justin\_Bieber.jpg" /></div>

<div class="bio\_title">About Justin Beiber</div>

<div class="bio\_content">

<p>Justin Drew Bieber is a Canadian...</p>

<p>His first full studio release... </p>

<p>The singer has been nominated... </p>

<div>

</div>

1. Add the following CSS rules to **styles.css** and save your work.

.bio {

margin-top: 20px;

margin-left: 20px;

padding: 12px;

background-color: #EEEEEE;

border: solid 1px #CCCCCC;

width: 480px;

}

.bio\_picture {

float: left;

margin-right: 8px;

margin-bottom: 8px;

}

.bio\_title {

font-size: 12pt;

color: #333333;

}

.bio\_content {

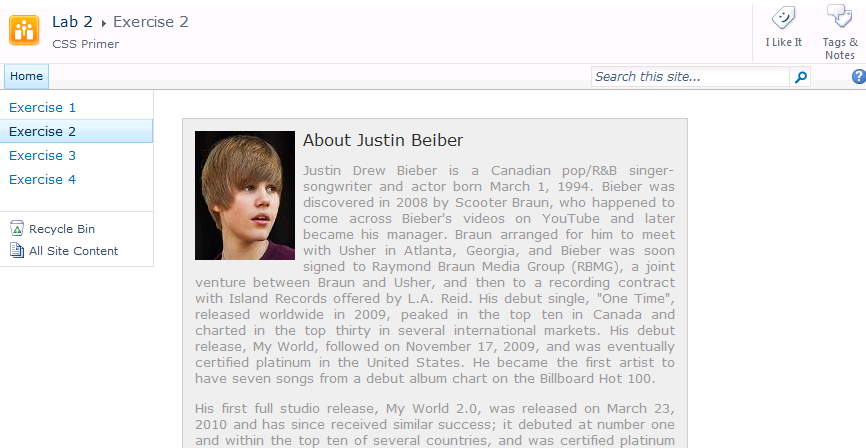
font-size: 10pt;

color: #999999;

text-align:justify;

}

1. In the browser, refresh the page named **Exercise2.aspx**. You should see a better looking page. Feel free to add embellishments to the styling if you would like.



### Exercise 3: Creating Columns with CSS

In this exercise you will work with CSS for modifying div elements that act as containers for inner elements. This will give you a chance to work with floating div elements and creating page layouts using columns.

1. In SharePoint Designer, open the page named **Exercise3.aspx** in advanced edit mode and examine the contents in the page. As you can see, this page is already linked to the style sheet named **styles.css**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

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

</asp:Content>

1. Examine the HTML content inside of **PlaceHolderMain**. Note that there are top-level div elements with id values of **left\_column** and **right\_column**. Inside these top-level div elements there are div elements structured just like the div elements in **Exercise 2**. More specifically, there is an outer div element with a class of **bio** and three inner div elements with class named **bio\_picture**, **bio\_title** and **bio\_content**.

<div id="left\_column" >

<div class="bio">

<div class="bio\_picture"><img src="images/Beatles.jpg" /></div>

<div class="bio\_title">The Beatles</div>

<div class="bio\_content">

<p>The Beatles were an English rock band..</p>

<p>Initially a five-piece line-up...</p>

<p>During their studio years, they...</p>

</div>

</div>

</div>

<div id="right\_column" >

<div class="bio">

<div class="bio\_picture"><img src="images/RollingStones.jpg" /></div>

<div class="bio\_title">The Rolling Stones</div>

<div class="bio\_content">

<p>The Rolling Stones are an English rock band...</p>

<p>Jones initially led the band...</p>

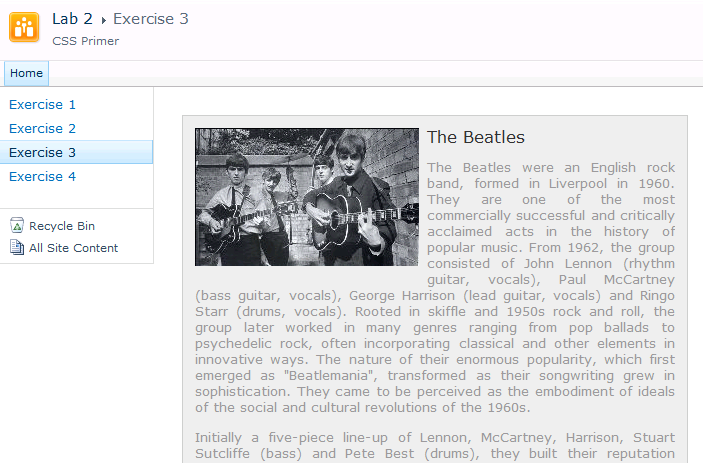
<p>First popular in Europe, The Rolling Stones...</p>

</div>

</div>

</div>

1. In the browser, examine the page named **Exercise3.aspx** by clicking the **Exercise 3** link in the Quick launch bar. As you can see, it has the same bio formatting you created in the previous exercise. The first bio on top shows the Beatles while the second bio below shows the Rolling Stones.



1. Your task in this exercise is to make these two bios show up side by side as left and right columns. Add the following CSS rules to **styles.css** and save your work.

#left\_column{

float:left;

width: 48%;

margin-top: 0px;

}

#right\_column{

width: 47%;

margin-left: 51%;

}

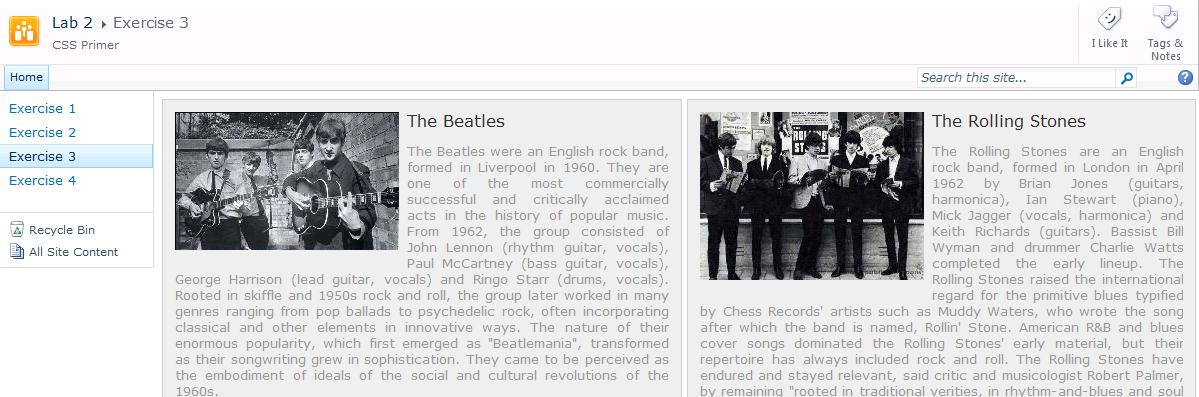
#left\_column > .bio, #right\_column > .bio {

width: 100%;

margin: 0px;

}

1. Save your work in **styles.css** and refresh the page in the browser. You should see the CSS rules you added produce the effect of a left column and a right column.



### Exercise 4: Changing the Display property of a div element to inline-block

In this exercise you will modify the **display** property of div elements to inline-block to produce a flow where the number of div elements shown per line changes dynamically as the page width is resized.

1. In SharePoint Designer, open the page named **Exercise4.aspx** in advanced edit mode and examine the contents in the page. As you can see, this page is already linked to the style sheet named **styles.css**.

<asp:Content ContentPlaceHolderId="PlaceHolderAdditionalPageHead" runat="server">

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

</asp:Content>

1. Examine the HTML content inside of **PlaceHolderMain**. Note that there are five top-level div elements containing information about five legendary rock stars. Each of these top-level div elements has a **rockstar\_bio**. Inside these top-level div elements are inner div elements structured just like in previous labs. More specifically, there is an outer div element with a class of **rockstar\_bio** and three inner div elements with class named **bio\_picture**, **bio\_title** and **bio\_content**. Below is an example of the basic HTML structure.

<div class="rockstar\_bio">

<div class="bio\_title">Jimi Hendrix (1942 - 1970)</div>

<div class="bio\_picture"><img src="images/jimi\_hendrix.jpg" /></div>

<div class="bio\_content">

<p>James Marshall Hendrix was an American guitarist...</p>

</div>

</div>

<div class="rockstar\_bio">

<div class="bio\_title">Janis Joplin (1943 - 1970)</div>

<div class="bio\_picture"><img src="images/janis\_joplin.jpg" /></div>

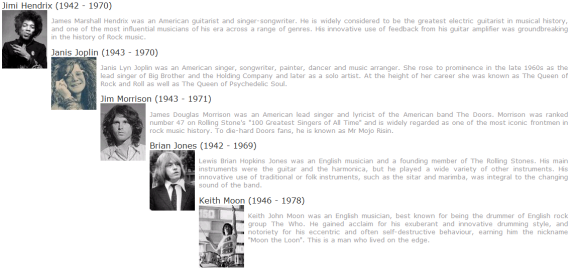
<div class="bio\_content">

<p>Janis Lyn Joplin was an American singer...</p>

</div>

</div>

1. In the browser, examine the page named **Exercise4.aspx** by clicking the **Exercise 4** link in the Quick launch bar. You should see a page that looks like the following screenshot.



1. You job is to use CSS to give each top-level div with the rockstar\_bio class a fixed width and fixed height, and to change the **display** property to **inline-block**. You will also set the **overflow** property to **hidden** so that the div elements never have scroll bars. Add the following CSS rules into **styles.css**.

div.rockstar\_bio {

margin: 10px;

padding: 0px;

background-color: #EEEEEE;

border: solid 1px #CCCCCC;

width: 280px;

height: 200px;

overflow: hidden;

display: inline-block;

}

div.rockstar\_bio > div.bio\_title {

background-color: Black;

color: White;

text-align: center;

padding: 4px;

}

div.rockstar\_bio > div.bio\_picture {

float: left;

margin: 8px 8px 4px 4px;

}

div.rockstar\_bio > div.bio\_content {

margin: 6px 8px 4px 8px;

}

div.rockstar\_bio > div.bio\_content > p {

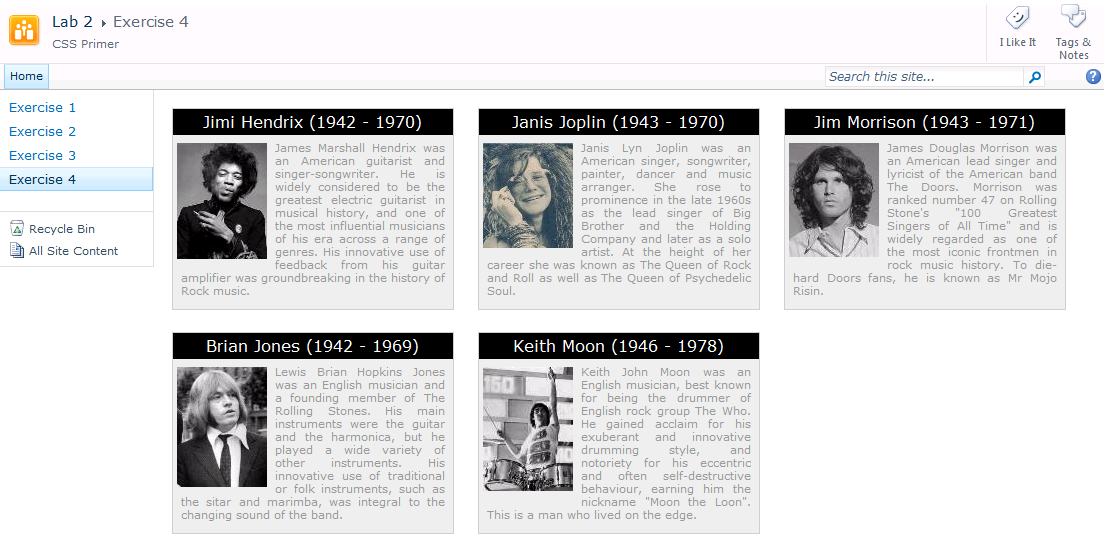
margin: 0px;

font-size: 8pt;

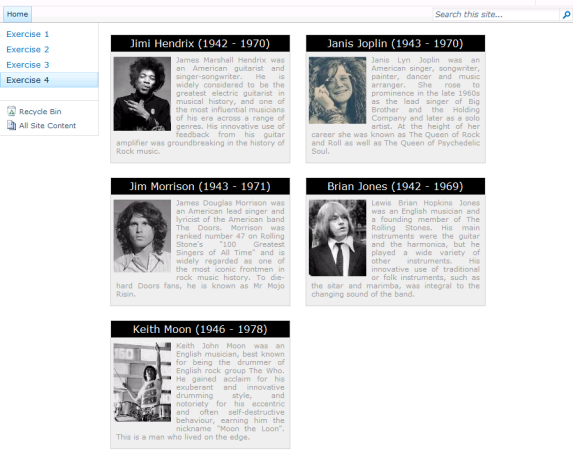
text-align: justify;

}

1. Save your work in **styles.css** and refresh the page in the browser. You should see the CSS rules you added product the effect of five div elements that span across a smaller number of rows.



1. Try dynamically resizing the width of the browser. You should be able to see the number of div elements shift from 3-per-row down to 2-per-row by making the browser window narrower.



### Exercise 5: Overriding CSS Rules from corev4.css

In this final exercise you will override CSS rules defined in corev4.css to change the color of the page title and to hide standard SharePoint page elements defined in v4.master.

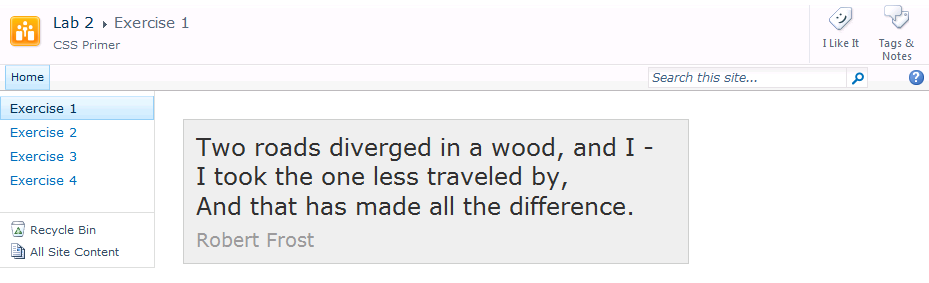
1. Add the following CSS rule to **styles.css** to change the color of the page title to a dark red color.

.s4-title H1 A, .s4-title H2 {

color: #800000;

}

1. Test your work. Save your changes to **styles.css** and refresh any page that links to **styles.css** including **Exercise1.aspx**, **Exercise2.aspx**, **Exercise3.aspx or Exercise4.aspx**. You should see that the Page Title now is displayed using a dark red font color.
2. Now it is time to get rid of some unwanted page elements. If you examine a standard page such as **Exercise2.aspx**, there are standard elements that you might want to hide such as the links for the **Recycle bin** and **View Site Content** link, the search area, and the social buttons that read **I like it** and **Tags and Notes**.



1. Add the following rule to **styles.css** to hide the div with the **Recycle bin** and the **View Site Content** link:

.s4-specialNavLinkList {

display: none;

}

1. Add the following rule to hide the div with the **Search area**:

#s4-searcharea {

display: none;

}

1. Add the following rule to hide the div with the **Social Buttons**:

.ms-socialNotif-Container {

display: none;

}

1. Test your work. Save your changes to **styles.css** and refresh any page that links to **styles.css** including **Exercise1.aspx**, **Exercise2.aspx**, **Exercise3.aspx or Exercise4.aspx**. You should see that your changes have removed those unwanted elements.

