

Expression Web Practice 2A

CSS Website Layout

Introduction: Creating a Web site layout from a wireframe diagram will help a designer to identify the boxes that need to be created in a layout. An external CSS style sheet will allow for the separation of the presentation from the content of a page. Refer to Layout with Expression Web and the Layout and CSS slideshows for help on this practice.

This is the Web page we will create in this practice:



Step 1. Getting Started

Download the ExpWeb2.zipped file to your desktop. Extract the ExpWeb2 folder and store it on your desktop. In Expression Web, use Site=>Open Site and choose the ExpWeb2 folder. This site has an index.html page built using the start.html page.

Step 2. Creating and Attaching an External Style Sheet

Create an external stylesheet using the File->New->CSS menu option. Name your stylesheet: **mystyles.css**. Open the index.html file and go to Code View of your index.html page. In the Apply Styles task pane click on the Attach a Style Sheet link. Browse and find the mystyles.css style sheet. Select a Link to attach

the style sheet and click OK. You should now see a <link> tag in the <head> of the index.html page that looks like this:

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

Step 3. Adding Boxes to a Sample Page

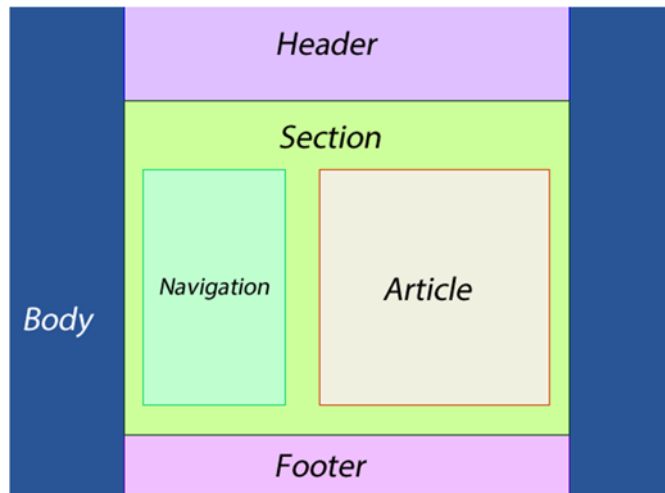


Figure 1 Box Structure

The box structure of the Web Site we will build is shown in the wireframe drawing above.

The outer box (blue background) is the body of the page so we don't need to add a box for this area.

Inside the <body> tag, add the following HTML5 elements to create the needed boxes for our layout (refer to Figure 2). It may help to indent these elements for readability.

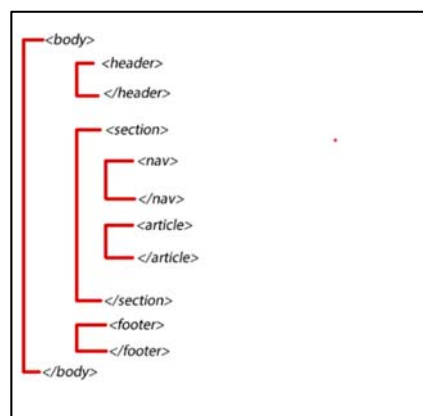


Figure 2 HTML 5 Document Structure

Step 4. Styling the Selectors for the Layout

The method that we will use to create the style rules in the external CSS stylesheet is to directly type them. IntelliSense can help with finding the properties we want. You may also use the Styles panel to create some of your style rules.

Add a new **<body>** selector style rule to mystyles.css. The body style should provide a margin and background image around the page content. Remember the body automatically wraps around everything else on the page.

```
margin: 0;
background-image: url('Images/background.gif');
font-family: Arial, Helvetica, sans-serif;
height: auto;
```

Add a new **<header>** selector style that will have the same width as the container, a fixed height, a background color, some top padding, and text alignment. The border will just add the dashed line on the bottom of the masthead box.

```
width: 760px;
height: 150px;
background-color: #CCCCCC;
text-align: center;
padding-top: 10px;
border: dashed medium navy;
margin: auto;
```

Add a new **<section>** selector style that will have a fixed width and wrap around the **<nav>** and **<article>** elements. This style should contain a margin, width, and border. The auto margin on the left and right will automatically center the page content.

```
width: 760px;
background-image: url('Images/inner_background.gif');
background-repeat: repeat-y;
overflow: hidden;
border-right: dashed medium navy;
border-left: dashed medium navy;
margin: auto;
```

Add a new **<nav>** selector style that will have a fixed width and will float left in the **<section>** box. If you add padding to any box, the browser will also add that to the width of your column.

```
float: left;
width: 110px;
padding-left: 10px;
```

Add a new **<article>** selector style that will have a fixed width and float to the left of the **<nav>** box.

```
float: left;
width: 600px;
padding: 5px;
text-align: center;
```

Add a new **<footer>** selector style that will have the same width as the **<header>** and a fixed height, It will also have the same background color as the **<header>**, a text alignment, a border, and some padding.

```
text-align: center;
width: 760px;
height: 70px;
border: dashed medium navy;
background-color: #CCCCCC;
margin: auto;
```

You should already be able to see some of these styles being applied on the index.html page. The boxes do not yet contain content, so it will still be hard to see the final effect of these styled selectors.

Step 5. Adding Needed Class Styles and Contextual Selectors for CSS a Navigation Bar

Let's also add a style to float the logo left with some padding. Because this style may need to be used more than once on a page this style rule should be classes instead of IDs. Remember, a class name must begin with the . character.

We will also want to position the logo to the left of the banner when we add it to the **** element inside the **<header>** element.

Add a new **.logoleft** class style that will float a logo image left and add padding on the top and left side.

```
float: left;
padding-top: 30px;
padding-left: 15px;
```

In order to create a style for the links in our navigation bar, we will use a contextual selector for links that appear only in the **<nav>** area. We will create a style for a regular anchor and then a style for the pseudo-class **a:hover**. If you want the visited state to also be styled, you could add another contextual selector for **<nav> a:visited**.

Add a new contextual selector **nav a** style (needs both the nav element and the a element) that will include border styling that mimics an unpressed button, a background color, a line height, a text color, a font-weight and font-size, and if you want, a font-family. We also want to prevent the default underline, so the text-decoration is set to none.

```
border: 2px inset #CCCCCC;  
background-color: maroon;  
line-height: 250%;  
color: #FFFFFF;  
font-weight: bold;  
font-size: 17px;  
text-decoration: none;
```

Add another contextual selector with a pseudo-class for a **nav a:hover** style (needs both the nav element and the a:hover). This style will change the border style to mimic a pressed button, and change the text color.

```
color: yellow;  
border: 2px outset #CCCCCC;
```

Step 6. Adding Content to the Sample Page

Add a banner image in the <header> element. Use the ccbanner.png image; add alternate text for the banner. The banner should be centered in the header.

If we want the logo to be placed to the left of the banner, first click your mouse before the banner image and then add the cclogo.gif image there. Apply the **.logoleft** style rule and click on it to apply this style to the logo. This logoleft class should float the logo to the left of the banner and add some padding. The banner will now center in the remaining space.

Next, we will add a navigation bar in the <nav> box using the <a> element. Add an anchor tag with the href attribute to index.html and the hypertext: Home. Add a
 element, and then add another anchor tag with the href="#" (we can create additional pages in a later lesson) and the hypertext: About Us. Add a
 element, and then add another anchor tag with the href="#" (we can create this page later) and the hypertext: Tributes. If you want the buttons to be longer, add some space characters () before and after the hypertext to lengthen the button.

Next, add the Welcomebear.gif image inside the section box and provide its needed alternate text. Add a couple of
 tags and then add the sewbearlialogo.gif image with its needed alternate text. Add a hyperlink to this image to link to the artist's Web site. The href should be:
<http://www.countryclipart.com> . Set the target frame of the hyperlink to be New Window or target="_blank" to open the site in a new window.

Click inside the footer box and add the heartline.gif with its needed alternate text. This image should center in the footer. At this point, you could save and validate your index.html page.