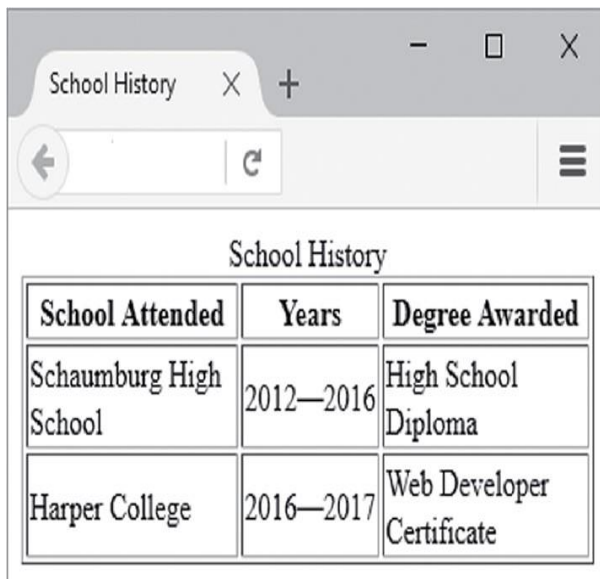


# CP 160 - Web Programming and Design

## Lab 9: Table, Form

### Hands-On Practice 1: Configure a table

Create a web page similar to **Figure 1** that describes two schools you have attended. Use “School History” as the caption. The table has three rows and three columns. The first row will have table header elements with the headings School Attended, Years, and Degree Awarded. You will complete the second and third rows with your own information within table data elements.



School Attended	Years	Degree Awarded
Schaumburg High School	2012—2016	High School Diploma
Harper College	2016—2017	Web Developer Certificate

**Figure 1 School History Table.**

To get started, launch a text editor and open the **template.html** file. Modify the title element. Use table, table row, table header, table data, and caption elements to configure a table similar to **Figure 2**.

The table has three rows and three columns. To configure a border, use `border="1"` on the `<table>` tag. Use the table header element for the cells in the first row.

Create a folder named **ch9table**. Save your file as `school.html` to the `ch9table` folder. Display it in a browser. It should look similar to **Figure 1**.

*Hints:* the sample code is as follows; **change** to your own school information; save it as `school.html`.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>School History</title>
<meta charset="utf-8">
</head>
<body>
<table border="1">
<caption>School History</caption>
  <tr>
    <th>School Attended</th>
    <th>Years</th>
    <th>Degree Awarded</th>
  </tr>
  <tr>
    <td>Schaumburg High School</td>
    <td>2012&mdash;2016</td>
    <td>High School Diploma</td>
  </tr>
  <tr>
    <td>Harper College</td>
    <td>2016&mdash;2017</td>
    <td>Web Developer Certificate</td>
  </tr>
</table>
</body>
</html>
```

## Hands-On Practice 2: Configure an informational table using CSS

In this Hands-On Practice, you will code CSS style rules to configure an informational table on a web page. Create a folder named `ch9table`. Copy the **starter.html** file to your `ch9table` folder. Display the file in a browser. The page should look similar to **Figure 2**.

Lighthouse Island Bistro Specialty Coffee Menu

Specialty Coffee	Description	Price
Lite Latte	Indulge in a shot of espresso with steamed, skim milk.	\$3.50
Mocha Latte	Choose dark or milk chocolate with steamed milk.	\$4.00
MCP Latte	A luscious mocha latte with caramel and pecan syrup.	\$4.50

**Figure 2** The table before CSS.

Launch a text editor and open the `starter.html` file from your `ch9table` folder. Locate the style tags in the head section. You will code embedded CSS in this Hands-On Practice. Position your cursor on the blank line between the style tags.

1. Configure the table element selector to be centered, have a dark blue, 5 pixel border, and have a width of 600px:

```
table { margin: auto; border: 5px solid #000066; width: 600px; }
```

Save the file as `menu.html` and display your page in a browser. Notice that there is a dark blue border surrounding the entire table.

2. Configure the `td` and `th` element selectors with a border, padding, and Arial or the default sans-serif font typeface:

```
td, th { border: 1px solid #000066; padding: 5px;
        font-family: Arial, sans-serif; }
```

Save the file as **menu.html** and display your page in a browser. Each table cell should now be outlined with a dark blue border and should display text in a sans-serif font.

3. Eliminate the empty space between the borders of the table cells with the **border-spacing** property. Add a `border-spacing: 0;` declaration to the table element selector. Save the file and display your page in a browser.

4. Configure the caption to be displayed with Verdana or the default sans-serif font typeface, bold font weight, font size 1.2em, and 5 pixels of bottom padding:

```
caption { font-family: Verdana, sans-serif; font-weight: bold;
          font-size: 1.2em; padding-bottom: 5px; }
```

5. Let's experiment and configure background colors for the rows instead of cell borders. Modify the style rule for the td and th element selectors, remove the border declaration, and set border-style to none:

```
td, th { padding: 5px; font-family: Arial, sans-serif;
          border-style: none; }
```

6. Create a new class called altrow that sets a background color:

```
.altrow { background-color: #eaeaea; }
```

7. Modify the <tr> tags in the HTML: assign the second and fourth <tr> tags to the altrow class.

Save the file. Display it in a browser. The table area should look similar to **Figure 3**.

Notice how the background color of the alternate rows adds subtle interest to the web page.

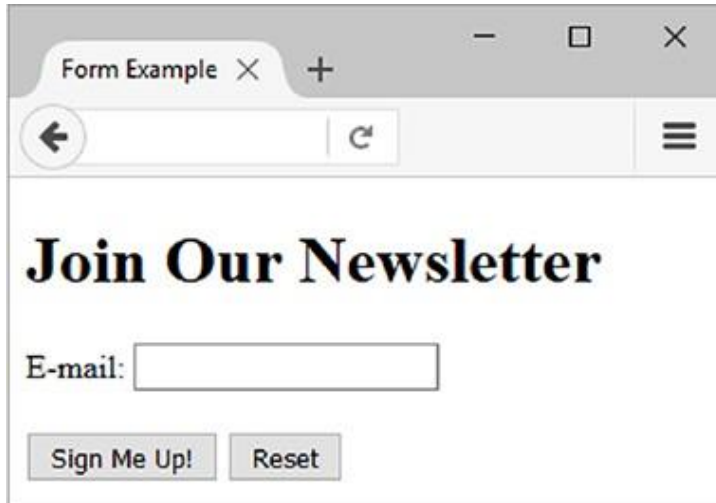
### Lighthouse Island Bistro Specialty Coffee Menu

Specialty Coffee	Description	Price
Lite Latte	Indulge in a shot of espresso with steamed, skim milk.	\$3.50
Mocha Latte	Choose dark or milk chocolate with steamed milk.	\$4.00
MCP Latte	A luscious mocha latte with caramel and pecan syrup.	\$4.50

**Figure 3** Rows are configured with alternating background colors.

## Hands-On Practice 3: Code a form

You will code a form in this Hands-On Practice. To get started, launch a text editor and open the `template.html` file. Save the file with the name `join.html`. You will create a web page with a form similar to the example in **Figure 4**.



**Figure 4** Example form.

1. Modify the title element to display the text: Form Example.
2. Configure an `h1` element with the text: Join Our Newsletter.
3. You are ready to configure the form area. A form begins with the form element. Place your cursor on a blank line under the heading you just added and type in a `<form>` tag as follows:

```
<form method="get">
```

As you read through the chapter, you will find that a number of attributes can be used with the `<form>` element. In your first form, we are using the minimal HTML needed to create the form.

4. To create the form control for the visitor's e-mail address to be entered, type the following code on a blank line below the form element:

```
E-mail: <input type="text" name="email" id="email"><br><br>
```

This places the text "E-mail:" in front of the text box used to enter the visitor's e-mail address. The input element has a `type` attribute with the value of `text` that causes the browser to display a text box. The `name` attribute assigns the name `email` to the information entered into the text box (the `value`) and could be used by serverside processing. The `id` attribute uniquely identifies the element on the page. The `<br>` elements configure line breaks.

5. Now you are ready to add the submit button to the form on the next line. Add a `value` attribute set to "Sign Me Up!":

```
<input type="submit" value="Sign Me Up!">
```

This causes the browser to display a button with “Sign Me Up!” instead of the default value of Submit Query.

6. Add a blank space after the submit button and code a reset button:

```
<input type="reset">
```

7. Next, code the closing form tag:

```
</form>
```

Save your file. Test your page in a browser. It should look similar to **Figure 4**.

Try to enter some information into your form. Try to click the submit button. Don’t worry if the form redisplayes but nothing seems to happen when you click the button—you haven’t configured this form to work with any server side processing.

## Hands-On Practice 4: Create a contact form

In this Hands-On Practice, you will create a contact form with the following form controls: a First Name text box, a Last Name text box, an E-mail text box, and a Comments scrolling text box. You'll use the form you created in **Hands-On Practice 3** (see **Figure 4**) as a starting point. Launch a text editor and open the file `join.html`. Save the file with the name `contact.html`.

1. Modify the title element to display the text: Contact Form.
2. Configure the h1 element with the text: Contact Us.
3. A form control for the e-mail address is already coded. Refer to **Figure 5** and note that you'll need to add text box form controls for the first name and last name above the e-mail form control. Position your cursor after the opening form tag and press the enter key twice to create two blank lines. Add the following code to accept the name of your web page visitor:

```
First Name: <input type="text" name="fname"
           id="fname"><br><br>
Last Name: <input type="text" name="lname"
           id="lname"><br><br>
```

4. Now you are ready to add the scrolling text box form control to the form using a `<textarea>` tag on a new line below the e-mail form control. The code follows:

```
Comments:<br>
<textarea name="comments" id="comments">
</textarea><br><br>
```

Save your file and display in a browser to view the default display of a scrolling text box. Note that this default display will differ by browser. Some browsers initially display a vertical scroll bar, while other browsers only render scroll bars after enough text is entered to require them. The developers of browser rendering engines determine the default display of form controls.

5. Let's configure the `rows` and `cols` attributes for the scrolling text box form control. Modify the `<textarea>` tag and set `rows="4"` and `cols="40"` as follows:

```
Comments:<br> <textarea name="comments" id="comments" rows="4"
cols="40"></textarea><br><br>
```

6. Next, modify the text displayed on the submit button (set the `value` attribute to "Contact").

Save your file. Test your page in a browser. It should look similar to **Figure 5**.

Try entering some information into your form. Try clicking the submit button. Don't worry if the form redisplayes but nothing seems to happen when you click the button—you haven't configured this form to work with any server-side processing. Connecting forms to server-side processing is demonstrated later in this chapter.

The image shows a web browser window with a single tab titled "Contact Form". The browser's address bar is empty, and the page content is a contact form. The form has a heading "Contact Us" in a large, bold, serif font. Below the heading are three input fields: "First Name:", "Last Name:", and "E-mail:", each followed by a rectangular text box. Below these is a "Comments:" label followed by a larger rectangular text area. At the bottom of the form are two buttons: "Contact" and "Reset". The browser window has standard OS controls (minimize, maximize, close) in the top right corner.

Contact Form

## Contact Us

First Name:

Last Name:

E-mail:

Comments:

**Figure 5** A typical contact form.



## Hands-On Practice 5: Style a form with CSS

You will style a form with CSS in this Hands-On Practice. To get started, launch a text editor and open the starter\_form.html file. Save the file with the name contactus.html.

The HTML for the form is shown below for your reference:

```
<form>
  <label for="myName">Name:</label>
  <input type="text" name="myName" id="myName">
  <label for="myEmail">E-mail:</label>
  <input type="text" name="myEmail" id="myEmail">
  <label for="myComments">Comments:</label>
  <textarea name="myComments" id="myComments" rows="2"
cols="20"></textarea>
  <input type="submit" value="Submit">
</form>
```

Configure embedded CSS within the style element as follows:

1. **The form element selector.** Configure with a #EAEAEA background color, Arial or sans serif font, 350px width, and 10 pixels of padding:

```
form { background-color: #EAEAEA;
        font-family: Arial, sans-serif;
        width: 350px; padding: 10px; }
```

2. **The label element selector.** Configure to float to the left, clear left floats, and use block display. Also set width to 100px, 10 pixels of right padding, a 10px top margin, and right aligned text:

```
label { float: left; clear: left; display: block;
        width: 100px; padding-right: 10px;
        margin-top: 10px; text-align: right; }
```

3. **The input element selector.** Configure with block display and a 10px top margin:

```
input { display: block; margin-top: 10px; }
```

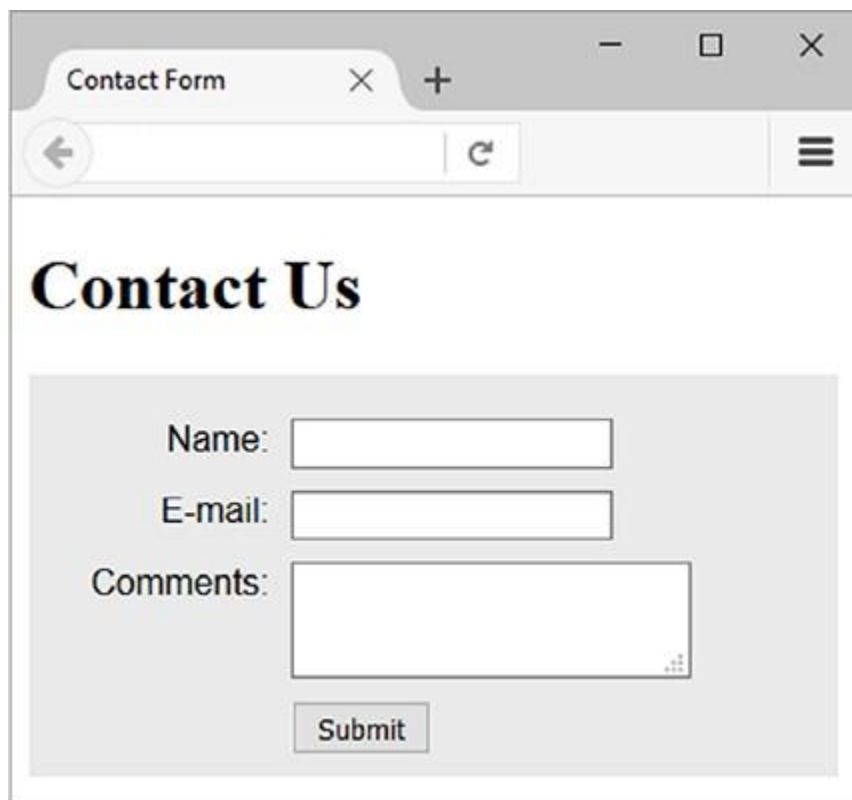
4. **The textarea element selector.** Configure with block display and a 10px top margin:

```
textarea { display: block; margin-top: 10px; }
```

5. **The submit button.** The submit button needs to display under the other form controls, with a 110 pixel margin on the left. You could configure a new id or class and then edit the HTML, but there is a more efficient method. You will configure a new type of selector, an **attribute selector**, which allows you to select using both the element name and attribute value as the criteria. In this case, we need to style input tags that have a `type` attribute with the value `submit` in a different manner than the other input tags, so we'll configure an attribute selector for this purpose. The CSS follows:

```
input[type="submit"] { margin-left: 110px; }
```

Save your file and test your page in a browser. It should look similar to **Figure 6**.



**Figure 6** The form is styled with CSS.

## Lab Completion / Submission:

Complete all the lab practices. Take the screenshots of your completed webpages; put them into a single word file and submit it to **Blackboard -> CP160 -> Assessments -> Lab9 / Assignment 9**; due date: today

## Homework Assignment:

1. Create a one-paragraph conclusion of what you have learned during the lab today.
2. Write the HTML code for a three-column table to describe the courses you are taking this semester. The columns should contain the course number, course name, and instructor name. The first row of the table should use `th` tags and contain descriptive headings for the columns. Use the table row grouping tags `<thead>` and `<tbody>` in your table.
3. Write the code to create a text box named `username` that will accept the user name of web page visitors. The text box should allow a maximum of 30 characters to be entered.

Submit to **Blackboard -> CP160 -> Assessments -> Lab9 / Assignment 9**; due date: 1 week from today.