1. Set Up Your Web Page for JavaScript

Before you begin to write JavaScript, you must set up your Web page so JavaScript will run. In our first HTML lab you learned you need to put the <html> tag at the beginning of your Web page so the browser knows it is reading an HTML file. Now that we are about to write code, we must similarly inform the Web browser the script language we will be using is JavaScript.

Look at the source code you find at this location:

lab7.txt

Save the file as lab7.html on your "Desktop." This will be the template you work from in the next several labs.

Skill: JavaScript Definition

To alert the browser the statements that come next are in JavaScript, you will want to use the following basic format:

```
<script type="text/javascript">
  your JavaScript will go here
</script>
```

You can see that the <script> tag must have a corresponding </script> tag to complete the JavaScript coding.

Where to Put Your JavaScript

JavaScript code can be written in the <head> of your Web page, or in the <body> of your page. Where you put your script depends on what you want to do. This time we will put JavaScript into the <body> of our HTML page.

Note

You can have as many <script> tags as you want, so don’t feel limited to putting all your code in one place.

2. Hide JavaScript from Old Browsers Using Comments
Because *JavaScript* is not always supported by older browsers, we want to "hide" the code by commenting it out. As with cascading style sheets (CSS), we will insert comment tags around our *JavaScript*, so if a browser is too old to understand *JavaScript*, it will not display the script on your Web page.

In HTML the characters <!-- represent the start of a multi-line comment. In *JavaScript* <!-- represents a single-line comment. *JavaScript* will ignore this line even though it is within the beginning and ending <script> tags.

In *JavaScript* the characters // represent the start of a single-line comment. This means anything after those two characters, on that line of code, will be ignored by *JavaScript*. Later you will learn how to do multi-line comments in *JavaScript*.

If the browser can understand *JavaScript*, it will ignore this effort to hide the code from the browser. A multi-line comment, in html, ends with the characters: -->. Therefore, you need to end the comment with the characters: -->, so if the browser cannot understand *JavaScript*, it will ignore the script code between the comments to hide the *JavaScript*.

### Note

Adding the text "begin comment to hide JavaScript" and "end comment to hide JavaScript" is a coding convention that allows you or others looking at your code to easily see where the comments begin and end. However, this text is not required for the tags to be functional.

A. Add the comment tags just after the beginning <script> tag and just before the end </script> tag.

```html
<script type="text/javascript">
    <!-- begin comment to hide JavaScript
    // end comment to hide JavaScript. -->
</script>
```

A. Add the beginning and ending script tags to your page just after the second <hr> tag as follows:

### Note

Code in **bold font** is the new code you should add to your page, while regular font indicates pre-existing code.
3. 43Write Text to the Page Using JavaScript

Using HTML, you were able to write text to your Web page using a variety of formatting tags, such as the `<p>` tag. Next you will write text on your Web page by using the JavaScript command `document.write`. Before you do that, let's break down the command itself.

**Concept: Understanding Objects and Properties**

*JavaScript* is an **object-oriented** programming language. **Objects** allow us to organize and manipulate information for components of our program. Most objects in *JavaScript* have **properties**. Properties allow us to define a specific instance of an object as having certain qualities. One real-world example of an object is a book. A book has properties, such as a title, author, and publisher.

**Note**

The specific title, author and publisher of the book are the **values** of the properties.
In our example, "book" is the object, the "title," "author," and "publisher" are the properties, and "The Hobbit," "Tolkien," and "Houghton Mifflin" are the values.

In this lab, we will work with the object document. The document object has properties, such as bgColor (background color), fgColor (foreground or text color), and lastModified (the last time the page was modified).

**Methods**

The easy way to think about methods is to look at them as the actions an object can take. In our example, write is an action the object document can take. Using the command document.write, we will be able to write text to our document. In other words, we will be able to write text to our Web page. The syntax for using document.write is:

```
document.write("The text you would like to appear on your page")
```

**Note**

Every time you write text, the text is considered a string, and you must put quotation marks around it.

A. Between your beginning and ending <script> tags insert the following (bold) code:

```
<script type="text/javascript">
   <!-- begin comment to hide JavaScript
       document.write("This page was last modified on: ");
   // end comment to hide JavaScript. -->
</script>
```
4. **Add a Last Modified Time/Date Stamp to Your Web Page**

In Project 1 you were required to put code on your Web page to display the time your Web page was last modified. Now you are going to add that code to our Web page, but this time we will know exactly what each part means.

**Concept: Variables**

Understanding **variables** is important to programming. We will be revisiting this concept several times in the next few labs and lessons. The best way to think about variables is to think of them as **containers** that store a piece of information. There are two important things to know about using variables:

1. Variables can contain one **value**. That value might be as short as a single character, or as long as a paragraph. However only one value will be stored in a variable at any given time.

2. Before using a variable, you should **declare**, or **create** it.

**Skill: JavaScript Variable Declaration Syntax**

To declare a variable, use the following command:

```
var variablename
```

**Note**

Variable names should not have any spaces, and they should not be individual numbers.

Declaring a variable announces to your program that you will be using the variable later in your program, or container, to store information.

Let’s take a look at the code from Project 1:

```html
<script type="text/javascript">
<!-- begin comment to hide JavaScript
var modified;
modified = document.lastModified;
document.write(modified);
// end comment to hide JavaScript. -->
</script>
```
You should recognize the script tag from Step 1.

**var modified**—This declares "modified" as a variable, telling our *JavaScript* program to create a variable for us to use later in the program. This variable is a container that will hold values.

**modified = document.lastModified**—Here we are assigning, or giving, a value to our variable. We will talk more about assignment later. Notice our *variable*, modified, gets the *value* given by referencing a *property*, lastModified, of our *object*, document.

**document.write(modified)**—This line references our *object*, document, again and invokes a *method*, write, that will write the contents of our *variable*, modified, to our Web page.

</script>—End our script.

A. Insert the following code onto your Web page:

```javascript
<script type="text/javascript">
    <!-- begin comment to hide JavaScript
    
    document.write("This page was last modified on: ");

    var modified;
    modified = document.lastModified;
    document.write(modified);

    // end comment to hide JavaScript. -->
</script>
```

5. Create a Form for Your Web Page

Forms allow you to create a Web page with interactive elements. You have seen forms when using Web search engine pages, filling out online surveys, or buying products through the Internet. Forms are everywhere online. In this exercise we will be laying the groundwork for creating your own interactive Web page with a form. In Project 1, you had the option to have
several people rate your Web page. In this lab, we will create a survey asking those very same questions.

Forms allow you to produce several input types. A short, and incomplete, list follows:

<table>
<thead>
<tr>
<th>Text</th>
<th>Button</th>
<th>Checkbox</th>
<th>Radio</th>
<th>Submit</th>
</tr>
</thead>
</table>

These different types of inputs allow users to enter information into your form. Let’s create a few:

A. **Place** the beginning and ending **form tags** between the two `<hr>` tags in the sample given to you:

```html
<form>
</form>
</form>
```

B. **Add a name attribute** to your form. A form is an object just like document is an object, and you should always name your objects!

**Note**

As you learned in your variable assignment, it is always a good idea to name your objects something logical you can remember, and that describes the object you are naming.

For example, naming a form **FIT100quiz1** would be good for a FIT100 quiz, but **fred** or **xLw29** would not work as well.

```html
<form name="YourNameForThisForm">
</form>
```

C. Add a **Text Input** box to your form. Like the `<form>` tag, the `<input>` tag has a name attribute. This text box is going to hold the first name of the person who will be completing your form. Add text before the field so your user knows what to put in the field.

```html
<form name="YourNameForThisForm">
<p>First Name:
<input type="text" name="YourNameForThisTextBox">
</form>
```
D. **Add text fields** to hold the following pieces of information within your **one** form tag:

**Note**

Remember to name your fields!

First Name (already done in Step C)  
Last Name  
URL of the Web page the person will be rating  
Accuracy Rating  
Authority Rating  
Objectivity Rating  
Currency Rating  
Coverage or Scope Rating  
Accessibility Rating

E. **Add an input** for a **generic button**. Remember to name this as well. If you are unsure how to add a **submit** button, refer to your HTML book or one of the Web sites recommended for the HTML section of the course.

F. **Add instructions to your survey** so users will know what to enter in each box. For example, if you want your user to rate things on a scale of 1 to 10, say so. This task will require you to use your knowledge of layout, formatting, and HTML to provide a good user interface.

G. As a bonus, **specify the length of your text boxes** to fit your information. That is, your text box for the URL might be longer than the one needed to rate currency.

6. **Add Comments Describing Your JavaScript**

**Concept: Comments**

There are a number of reasons to put comments in the code you write. Comments let you describe exactly what your code does, and why you made the code choices you did. You might be working on a project
with other people, and comments will help your team members read your code more easily. Even if you are not working on a team, comments help you remember what you were thinking when you coded something. Picture yourself looking at something you wrote and asking, "What was I thinking??!"

In practical terms for this course, comments also help you and any people evaluating your work to understand the process you went through to get to a specific solution. Even if you can't get something to work, it is a good idea to leave the code written within a comment. If you realize your error later, you don't have to start all over again.

**Skill: HTML Commenting**

<!-- your comment goes here and can span multiple lines -->

**Skill: JavaScript Commenting**

// will comment out anything on a single line

/* will comment out everything between it and the end comment, even on multiple lines. This is how to end the comment */

A. Write comments for each step of this exercise. Make sure you use JavaScript commenting within the `<script>` tags and HTML commenting outside these tags. Explain in detail what each line of code you wrote does, and how it does it. Feel free to explain several lines of code at once if you are comfortable doing so.

**Example**

```html
<script type="text/javascript">  // start javascript
<!-- begin comment to hide JavaScript
var length;   // declares the variable to hold length
var width;      // declares the variable to hold width
var result;   // declares the variable to hold the result
length = 5;  // assigns length the value of 5
width = 6;  // assigns width the value of 6
result = length * width; /* result gets the value of length multiplied by width. I will use this variable later in the program in
```