1. Add an Image Link to your Web Page Using an HTML `<img>` Tag

A. Open the file `mouseover.html` and add your image of Red Square to the Web page.

B. Because we will need to reference this image later, we need to add a name attribute to this `<img>` tag. For example:

```
<img src="RedSquare.jpg" name="redsquare">
```

C. Add an anchor tag around the `<img>` tag to make the image a link. Use this link to go to the University of Washington homepage:

```
http://www.washington.edu/cambots/
```

where you can see a current picture of Red Square.
2. Create Two New Image Objects and Give Them Values

In this exercise you will add *JavaScript* to the `<head>` area of our Web page. By placing *JavaScript* in the head you cause that script to be loaded before anything else on the Web page.

**Note**

Without *JavaScript* to pre-load images, the image would only load when you rolled over it. This might cause a noticeable and undesirable delay the first time the image is transferred.

An image is a type of object in *JavaScript* that has several properties. In the same way we created and named an instance of the date object, we are going to create and name instances of an image object using the following format:

```javascript
YourImageName1 = new Image
YourImageName2 = new Image
```

A. Establish the beginning and end of your script with the `<script>` tag. You will put this in the `<head>` of your document anywhere between the `<head>` and `</head>` tags.

B. Create two variables to hold your two Red Square images (see format above).

C. Define the `src` property of each image. For example:

```javascript
RedSquare1.src = “RedSquare.jpg”
```

3. Add *mouseover* and *mouseout* Event Handlers to Your Link

**Events**

The concept of events is key to *JavaScript*. We know from our lessons that *JavaScript*, like many other programming languages, is written to run in linear order. In other words, it will start at the beginning of your code, and execute each line in the order it is written. As a result, if you have one line of code that references the variable `dog`, then you must have assigned a value to that variable *before* you get to that reference.
You can skip lines of code by using constructs such as the **conditional**. For example, "If the following expression is false, skip the rest of the conditional." To create an interactive element, however, you need to add lines of code to be executed only when a specific **event** occurs. Events can range from the click of a button to a movement of the mouse. These events trigger lines of code that would otherwise be left unexecuted.

Two such event handlers are called **onMouseover** and **onMouseout**. These event handlers are available with several objects; in this case, you will use them with the **link** object, otherwise known as the **anchor** tag.

**Note**

`onMouseover` is triggered when the user moves the mouse over something (a link), and `onMouseout` happens when the mouse moves off (the link).

**Document Object Model (DOM)**

The document object model, among other things, allows you to reference objects, properties, styles, and tags on your Web page by providing a structural representation of the document and the objects it contains. *JavaScript* uses the DOM to reference objects, and their properties, methods and events, available to us. In this case you will use the DOM to reference the image you placed on your Web page.

```javascript
document.ImageNameYouCreatedEarlier.src=YourFirstImage.src
```

This is the basic format you will be using in the lab. In this course we will almost always be starting with `document` when using the DOM. After that, you may reference the name of the object to which you are referring. In this case, it is the name you added to your `<img>` tag. Next, you are referencing the property `src` of that image. This allows you to change the source (`src`) of the image (your linked image) located on your document (`document`). The information following the `=` references the specific image to display.

A. Add an attribute to your anchor tag that causes the picture to change when you mouse over the image.

```html
<a href="http://www.washington.edu/
  onMouseover="document.ImageName.src="RedSquare2.src">
```
This should cause the picture to change to your second picture of RedSquare, the one with Mt. St. Helens in the background.

B. Add the `onMouseout` attribute to your anchor tag.

```html
<a href="http://www.washington.edu/"
    onMouseover="document.ImageName.src="RedSquare2.src"
    onMouseout="document.ImageName.src="RedSquare1.src"></a>
```

C. Save your work and reload `mouseover.html`. Does it work? The most common errors with this script are typos in the names of your objects and events. Remember that JavaScript is case sensitive!

4. Create a Function to Run When the "Submit" Button Is Clicked

**Important Note!**

For this exercise you will be working with your `lab7.html` file

In previous labs you created a form to allow users to enter information as they rate a Web page on a number of different criteria. In this lab you will be writing code that executes when the user of your Web page clicks the "Submit" button. The code JavaScript will execute when the form is submitted is called a **function**. Your lesson material and reading introduced you to functions, so you know they are blocks of code that can be reused, or **called**, from several different places in your program.

**The Structure**

```javascript
function YourFunctionName [arguments] {
    code statement;
}
```

Remember that all the code this function executes must be between the two curly brackets. Each line, if there are multiple lines, should end with a semicolon.

Ultimately, your function will do the following:

- verify that your user has entered something into the field for their name; and
• print the information entered to another screen for easy printing.

A. In the <head> section of your HTML, establish your beginning and ending <script> tags.

B. Create the outline of your function inside your new script. For this situation we will not use any arguments, so leave the area between the parentheses empty.

5. Verify Your User Entered a Name into Your Form

In this step you will write code to do the following:

"If the user has not entered anything into the First Name field of your form, then display an alert box to remind them to do so."

This statement should remind you of something from the last lab. Can you think of what kind of code you would need to write to implement this statement? Yup, that’s right, a **conditional**. Before you get started writing the statement, there are a few things you need to know.

**How do I reference the value of the first name field?**
As you learned in Step 3, you can use the DOM to reference different objects and their properties. In this case you will reference the value of an **object** (the **first name** field), that is part of a **form** on your **document**. You'll do this in the following format:

```
document.YourFormName.YourFieldName.value
```

**How do I check to see that nothing is entered?**
Information entered into a text box is going to be in the form of a **string**. A string is referenced with quotations around it. For example, "Fred" and "156 Pike Street." A string with nothing in it is simply two quotes together: ""

**How do I create an alert box?**
The **alert()** method allows you to write a message to a special dialog box the user will see. Use this method the same way you use the **document.write()** method, with the string you want to be printed inside the parentheses.

The Visual
Would you like to play a Game?
The Code

alert("Would you like to play a game?")

A. Declare a variable to hold the value from your first name field, and assign it the value from that field.

```
Var YourFirstNameVariable
YourFirstNameVariable = document.YourFormName.YourFieldName.value
```

B. Within your function, set up a conditional with a T/F expression that checks to see if your First Name field is == 0. Remember to:
- use two equal signs when asking if the values on either side of that operator are equal; and
- use your variable that holds the value for your first name field.

```
if (YourFirstNameFieldName == ")
{
}
```

C. Enter the code statement that pops up an alert box informing your users they must enter a name. The code statement goes between the two curly brackets. See the beginning of this step for details on alert boxes.

6. Call Your Function When the "Submit" Button Is Clicked

As with the onMouseover event handler, you will use the onClick event handler to call your function. The onClick event handler can be used with any HTML tag, but in this case you are going to add it to the <input> tag that displays your "Submit" button.

A. Add the following to your submit button <input> tag:

```
onClick="YourFunctionName()"
```

Note

Remember to put quotation marks around your function name, as well as including the opening and closing parentheses.
B. Save your work, and test your page by reloading and clicking on "Submit". Remember not to enter any information in the first name field!

What happens when you enter information to the **last name** field, but not the **first name**, and then click "Submit"? Try it and see for yourself. Your form is refreshed, and the information you entered is lost, right?

The way to fix this is to add a statement to your `onClick` event handler that tells the browser not to refresh after it calls your function. Do this by entering:

```
onClick="YourFunctionName(); return false"
```

**Note**

Don’t forget the semicolon `;` that separates your two code statements!

7. Add an Expression to Your Conditional to Verify the Last Name

You know from the lesson that there are operators that compare values. In this step you will be using such a **logical operator**. There are two major logical operators used with conditionals:

1. `&&` **AND**  
   Example: if (cake < $3.00 && nuts == false) {buy the cake}

2. `||` **OR**  
   Example: if (fireplace == true || garage == true) {buy the house}

Our new requirement for a conditional is the following, with new statements in bold:

"If the user has not entered anything into the First Name field of your form **OR the user has not entered anything into the Last Name field**, then display an alert box to remind them to do so."

You will need to add a logical operator with a new T/F expression to your conditional.
A. Add another variable that will hold the value from the **last name** field.

B. Assign this variable the value from the **last name** field. See Step 7, Part A for details.

**Note**

It is common practice to declare all your variables at the top of your function. Feel free to organize your code in your own way, but make sure you always declare your variables **before** you give them values.

C. Inside the parentheses of your **if/then** statement, add an **or** operator and the additional **T/F** expression.

**Hint**

If you are having trouble getting started, break the steps down.

1. First, what symbol, explained in this exercise, is used to represent **OR**?
2. How do you write a **T/F** expression to see if the last name field has anything entered in it? This should be the same as the first **T/F** expression you wrote; but instead of using the **first name**, use the **last name**.
3. Now that you have the symbol and the expression, put those together inside the parentheses. See the **OR** example above for more detail.

D. Add a comment in your code that explains what you did, including why you used the variable and field names you did, and examples of what would cause your conditional to execute. For example: "If my user enters X into Y field but nothing in the Z field, then ...." (where X, Y and Z are specific names.) Give at least three examples, and describe what would happen in each example.

8. **Declare Variables to Hold the Values from your Form's Fields**
In the previous steps you created variables to hold the values in your first name and last name fields. In this step you will create the seven additional variables that will hold the rest of the values from your fields.

A. Declare your new variables in the function you created in the `<head>` of your document.

B. At the end of this step you should have variables for the following fields:

- First Name (Done in Step 5)
- Last Name (Done in Step 7)
- URL the user will be rating
- Accuracy Rating
- Authority Rating
- Objectivity Rating
- Currency Rating
- Coverage/Scope Rating
- Accessibility Rating

**Note**

Because you placed your variables inside your function and you used `var` when declaring them, you will only be able to use them in this specific function. This is the difference between local and global variables.

**Local** variables are used in specific areas of your program, such as within a function. **Global** variables can be used anywhere within your program. There are reasons to use both, but always know the scope of your variables!

9. Assign Your New Variables the Values from Your Field

You have already assigned the first name and last name values to their respective variables. However, let’s briefly review assignment.

**The Basic Structure**

\[ x = y \]  

The value on the right side is assigned to the variable or object on the left side

**Examples**
greeting = “Hello World!”  Note that greeting is a variable, and that Hello World! is in quote marks because it is a string value.

width = 4  Note that because 4 is a number, not a string, you do not need quote marks.

fname = document.MyForm.first.value  The value held in the field named first when this assignment occurs is assigned to the variable.

boxarea = length * width  The variable on the left is assigned the value of length * width. Note the assigned value can be in the form of a literal value, a variable, or an expression.

document.MyForm.area.value = boxarea  This assignment takes the value held in the variable to the left and places it in the field named area.

A. Assign the values for each of your fields into the corresponding variable you created. The third example above should help you with the syntax.

10. Write the Field Values to the Document

In this step you will use the write() method to write the value of each field to the Web page. This will make printing the results of your survey easy.

Note

You do not need a printer for this exercise! Imagine your user will have a printer, and that they will be instructed to print their Web page results and mail them to you. This will not be required.

This step will be written within your function. Let’s review what your function looks like so far.

1. The <script> tag tells the browser this is a JavaScript script.
2. The function name and the curly brackets indicate the start of your function
3. There are at least nine variable declarations—one for each field on your form.
4. Each field has its value assigned to the one of the variables.
5. A conditional checks to see if either the first or last name was left blank.

Remember this script will be run in the order it is written! Declare your variable before you assign a value to it, and make sure you assigned a value to the variable before you used it in the conditional.

B. **After** your conditional, use `document.write()` to write the values for each field to your form. For example:

```javascript
document.write(YourVariableNameforFirstName)
```

**Note**

Make sure you put this **after** your conditional, and **inside** the curly brackets defining your function. If the program is reading the function line by line, what will happen if you put the `document.write()` line **before** your conditional? That’s right: it will print the items **before** it checks to see if your user has put in a name!
C. Save your file, and test your form. Does it work?

Ooops! All your information is in one line, with no spaces, right? Also, you will note that—while the name and URL fields are obvious—if you looked at this page alone, you would have a hard time figuring out which field output was which. You might wonder, “Did Accuracy come first, or was it Authority?”

D. Add an expression to each `document.write()` method you just wrote that will add a `<p>` tag as well as a heading. For example:

```javascript
Document.write("<p>First Name: " + YourFirstNameVariable)
```

**Bonus Work (Not Required!)**

E. Add a header or some other information to this form to make it more user-friendly. For example, you might add a title, an explanation of the contents, and perhaps some instructions for your mythical users to follow in returning the form to you.

F. Add a personalized element to your form. For example, something that says:

“Thank you Fred, your participation is much appreciated!”

**Hint:** You have your user’s first name completed and stored in a variable!