

FIT 100: Fluency with Information Technology

LAB 11/12: Events and Functions

Required Reading for Lab 11/12

- Chapter 18 of Snyder, *Fluency with Information Technology*

Optional Reading:

- Chapter 2 of Negrino/Smith, *Javascript for the World Wide Web*

Table of Contents

1. Add an image link to your webpage using an HTML tag.....	1
2. Create two new image objects and give them values.....	3
3. Add onMouseover and onMouseout event handlers to your link.....	4
4. Create a function that will run when the submit button is clicked.	6
5. Verify that your user has entered a name into your form.	7
6. Call your function when the submit button is clicked.	9
7. Add an expression to your conditional to verify the last name also.	10
8. Declare variables that will hold the values from the fields your form.....	11
9. Assign your new variables the values from your field.	12
10. Write the values of the fields to the document.....	13

1. Add an image link to your webpage using an HTML tag.

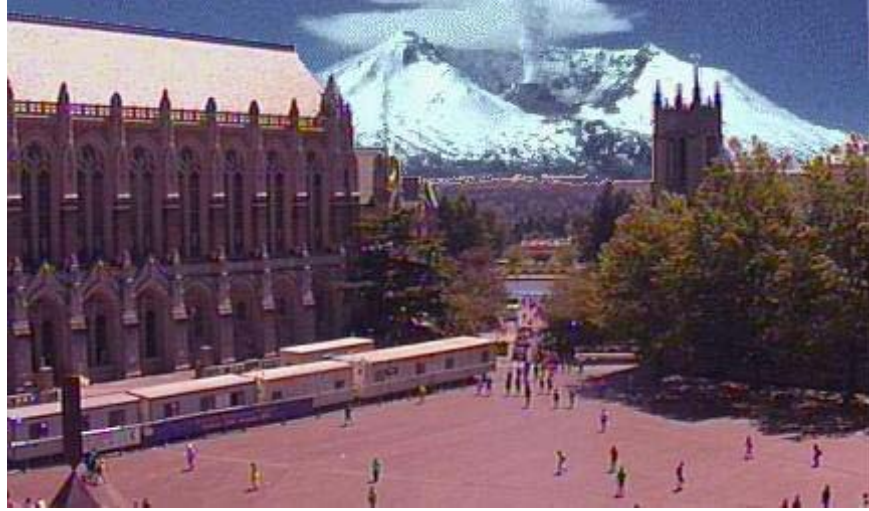
- Open up the file mouseover.html and add your image of Red Square to the webpage.
- Because we will need to reference this image later, we need to add a name attribute to this tag. For example:

```

```

- Add an anchor tag around the tag to make the image a link. Use this link to go to the University of Washington homepage: <http://www.washington.edu/> where you can see a current picture of Red Square.





2. Create two new image objects and give them values.

In this exercise we will be adding JavaScript to the <head> area of our webpage. By placing JavaScript in the head we cause that script to be loaded before anything else on the webpage.

Note: Without the 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 as the image is transferred.

An image is a type of object in JavaScript that has several properties. In the same way that 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:

Note: The name that you use for your image name should NOT be the same name that you use in your tag. In this case, we used "mouseover" as the example name for our tag in Exercise 1. Therefore, you could use any name for your image name except *mouseover*.

```
YourImageName1 = new Image;  
YourImageName2 = new Image;
```

- D. 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.
- E. Create two variables to hold your two Red Square images. (see above format)
- F. Define the src property of each image. For example:

```
RedSquare1.src = "RedSquare.jpg";
```

3. Add onmouseover and onmouseout 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.

We have ways of skipping lines of code by using constructs such as the conditional. (If the following expression is false, skip the rest of the conditional.) However, to create an interactive element we need to add lines of code that are executed only when a specific *event* occurs. Events can range from a 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, but in this case, we will be using them with the link object, otherwise known as the anchor tag.

Note: `onmouseover` is triggered when someone moves their mouse over something (a link) and `onmouseout` happens when the mouse moves back off it (the link).

Document Object Model (DOM):

The document object model, among other things, allows us to reference objects, properties, styles and tags (to name a few) on our webpage by providing a structural representation of the document and the objects they contain. JavaScript uses the DOM to reference objects (and their properties, methods and events) available to us. In this case we will be using the DOM to reference the image we have placed on our webpage.

```
document.ImageNameYouCreatedEarlier.src=YourFirstImage.src;
```

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

- G. Add an attribute to your anchor tag that causes the picture to change when you mouseover the image.

```
<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 Mount Saint Helen's in the background.

- H. Add the `onmouseout` attribute to your anchor tag.

```
<a href="http://www.washington.edu/"
```

```
onmouseover="document.ImageName.src="RedSquare2.src;"  
onmouseout="document.ImageName.src="RedSquare1.src;">
```

- I. 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 that will run when the submit button is clicked.

Important Note: For this exercise we will be working with our lab8.html file.

Concept: Functions

In prior labs we have created a form that will allow users to enter in information as they rate a web page on a number of different criteria. In today's lab we will be writing code that will execute when the user of your webpage clicks the submit button. The code that JavaScript will execute when the form is submitted is called a function. Your lesson readings have introduced you to functions so you know that they are blocks of code that can be reused, or called, from several different places in your program.

Skill: JavaScript Syntax for Functions

The Structure:

```
function YourFunctionName(arguments)
{
    code statement;
}
```

Remember that all the code that this function will execute needs to be between the two curly brackets. Each line (if multiple lines) should end with a semicolon.

The steps below will create the outline of a function that we will fill in later.

- J. In the <head> section of your HTML, establish your beginning and ending <script> tags.
- K. 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 that your user has entered a name into your form.

Remember to read the beginning of this exercise before beginning Steps A, B and C.

In this exercise we will write code that will do the following:

If the user has not entered anything into the First Name field of your form, then display an alert box that will 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 in code? Yup, that's right, a conditional. **Before you get started**, there are a few things you need to know before you can write this statement.

How do I reference the value of the first name field?

As we learned in Exercise 3, we can use the DOM to reference different objects and their properties. In this case we will be referencing the value of an object (first name field) that is part of a form on our document. We 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 as a string. Remember that a string is referenced with quotations around it. For example, "Fred" and "156 Pike Street". A string with nothing in it would be simply two quotes together: ""

How do I create an alert box?

The alert() method will allow you to write a message to a special dialog box that the user will see. Use this method in the same way that you use the document.write() method, with the string you wish to be printed inside the parentheses.

The visual:



The code:

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

- L. Inside the function you created in Exercise 4, 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;
```

M. Within your function that you created in Exercise 4, 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.
- Use your variable that holds the value for your first name field.

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

N. Enter the code statement that will pop up an alert box informing your user that they must enter a name. Remember that the code statement goes between the two curly brackets. See the beginning of this exercise for details on alert boxes.

6. Call your function when the submit button is clicked.

Like the onMouseover event handler, we will be using the onClick event handler to call our function. The onClick event handler can be used with any HTML tag, but in this case we are going to add it to our <input> tag that displays our submit button.

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

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

Note: Remember to put quotations around your function name, as well as including the opening and closing parentheses.

- P. 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 have entered information to the last name field, but not the first name, and you click submit? Try it to see for yourself. Your form is refreshed and the information you had 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: Remember that you need the semicolon to separate your two code statements!

7. Add an expression to your conditional to verify the last name also.

You know from lecture and the readings that there are operators that compare values. In this Exercise we will be using a logical operator. There are two major logical operators that we use with conditionals:

&& AND Example: if (cake < \$3.00 && nuts == false) { buy the cake }
|| 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 that will remind them to do so and add the additional statement:

return false;

So that the page is not refreshed and they can fill in the text boxes needed.

We will need to add a logical operator with a new T/F expression to our conditional.

- Q. Add another variable that will hold the value from the last name field.
- R. Assign this variable the value from the last name field. See exercise 7, step 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.

- S. 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. First, what symbol, explained in this exercise, is used to represent **OR**? 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. Now that you have the symbol and the expression, put those together inside the parentheses. See the OR example above for more detail.

- T. Add a comment in your code that explains what you did including why you used the variable and field names you did, as well as 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 the specific names.) Give at least three examples and what would happen in each example.

8. Declare variables that will hold the values from the fields your form.

In the previous exercises we have created variables that will hold the values in our first and last name fields. In this exercise you will create the seven additional variables that will hold the rest of the values from your fields.

- U. Declare your new variables in the function you have created in the <head> of your document.
- V. At the end of this exercise you should have variables for the following fields:
 - First Name (Done in Exercise 5)
 - Last Name (Done in Exercise 7)
 - Url that the person will be rating
 - Accuracy Rating
 - Authority Rating
 - Objectivity Rating
 - Currency Rating
 - Coverage/Scope Rating
 - Accessibility Rating

Note: Because you have 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 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 property on the left side

Examples:

`greeting = "Hello World!"` Note that `greeting` is a variable, and that *Hello World!* is in quotes because it is a string value.

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

`fname = document.MyForm.first.value` The value that is 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 that 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*.

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

10. Write the values of the fields to the document.

In this exercise we will be using the write() method to write the value of each field to the webpage. This will make printing the results of your survey easy.

Note: You do NOT need a printer for this exercise! Imagine that your user will have a printer and that they will be instructed to print out their webpage results and mail them to you. However this will **NOT** be required.

This exercise will be written within our function. Let's review what our function looks like so far.

1. The <script> tag that tells the browser that this is a JavaScript script
2. The function name and the curly brackets indicating the start of our function
3. At least 9 variable declarations, one for each field on our form.
4. Each field has its value assigned to the one of the variables above.
5. A conditional that checks to see if either the first or last name has been left blank and if it has, is set to return false so the form is not reloaded.

Remember that 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 have assigned a value to the variable before you use it in the conditional.

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

```
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!

- Y. Save your file and test out your form. Does it work?

Oops! All of 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. "Did Accuracy come first or was it Authority?"

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

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

Added bonus work (not required!):

- AA. Add a header and/or some other information to this form so that it is more user friendly. A title and an explanation of the contents, and perhaps some instructions

for your mythical users to follow in returning the form to you.

BB. Add a personalized element to your form. For example, something that says:
"Thank you Fred, your participation is much appreciated!"

Remember that you have your users first name filled out and stored in a variable!