

FIT 100: Fluency with Information Technology

LAB 13/14: Arrays and Iteration

Required Reading for Lab 13/14

- Review Chapters 19 and 20 of *Fluency with Information Technology*

Optional Reading

- Chapter 7 *Javascript for the World Wide Web*

Table of Contents

1. Create an Array Object to hold your favorite foods..... 1
2. Print the values in your array to the webpage using iteration..... 3
3. Modifying Code: Change the provided code to create your own random links. 5
4. Add your modified code to your lab3.html webpage. 7
5. Integrate your mouseover.html and array.html JavaScript into lab3.html. . 7

1. Create an Array Object to hold your favorite foods.

For this Exercise, use the file provided called [arrays.html](#)

Concept: Arrays

Arrays are objects that can hold a number of values for things that are related in a single variable name. When writing an array you need to know three things, the name of your array, how many elements it will contain, and the values of those elements. It is possible to refer to a single element in the array using the name of the array and its index number.

Skill: Array syntax in JavaScript:

```
var ArrayName = new Array(6);
    ArrayName [0] = "value";
    ArrayName [1] = "value";
    ArrayName [2] = "value";
    ArrayName [3] = "value";
    ArrayName [4] = "value";
    ArrayName [5] = "value";
```

Note: Notice that the array elements are numbered starting at 0! These numbers just work as a counter to refer to specific elements in our array.

- A. Start a `<script>` tag in the body of your array.html document.
- B. Declare your array with the following:

```
var YourArrayName = new Array(4);
```

C. Assign values to your array elements. For example:

```
MyArrayName[0] = "Sushi";  
MyArrayName[1] = "Pizza";  
MyArrayName[2] = "Pita and Hummus";  
MyArrayName[3] = "Waffles";
```

Note: Remember to use your own Array object name and put in your own favorite foods (unless, of course, they are the same as mine!)

2. Print the values in your array to the webpage using iteration.

If I asked you to print the values of your array to the webpage you might decide to start typing something like:

```
document.write(MyArrayName[0] + "<p>" + MyArrayName[1] + ..... )
```

You get the point. To print all of your favorite foods to the webpage you would need to write each one out. With four elements, that might not be so bad, but what if you had 100? or even 10,000? Iteration allows us to do this task much more efficiently.

Concept: Iteration:

As it says in the book, "Iteration is the repeated execution of a series of statements in programming a specified number of times or until a certain condition is met." In this particular example, our statements cause our first array item to be printed to the webpage, then the second array item, then the third and finally the fourth item. The condition that needs to be met to stop our iteration is that we have printed the last (or fourth) item to our webpage.

Skill: While Iteration Syntax for JavaScript:

```
while (<stop condition>)  
{  
    code statement;  
    code statement;  
}
```

D. Declare a variable that will hold a number and assign it a value of 0.

Note: This is just a counter. We are assigning it a value of 0 to make sure we know what number our counter is starting with

E. Establish the while iteration by typing the following bolded text:

```
var num;  
num = 0;  
  
while()  
  {  
  
  }
```

F. Write a stop condition that will cause the loop to stop after num equals 3.

```
var num;
```

```
num = 0;

while(num <= 3)
{

}
```

- G. Write the code statements within the curly brackets that will cause your favorite foods to print to the screen. Also add a separate line of code that will increase our counter.

```
var num;
num = 0;

while(num <= 3)
{
    document.write("<p>" + YourArrayName[num]);
    num = num + 1;
}
```

Notice that the variable `num` is used in place of a literal number. This is because each time this loop runs, `num`'s value will increase by one and as a result, print all four of your favorite foods. The best way to make sure you really understand the iteration is to do the loop yourself. Start at the top and say to yourself:

The first time the while iteration runs `num` equals 0. Is `num` less than or equal to 3? *Yes because `num` equals 0*, so I will perform the code statements in the loop. The array element that I'm printing will be the 0 element at `YourArrayName[0]` which is *Sushi*. Then 1 is added to `num` so `num` equals 1. This is a loop so I will start at the beginning again. Is `num` less than or equal to 3?

- H. Write comments by your code that explain what your while iteration is doing. You will probably be more brief than the example above, you may use phrases like, "each time the iteration is performed..."
- I. **BONUS!!!** There is a particular property of arrays that actually stores the length of the array. Can you figure out how to change your iteration so that the loop will run as long as the value in `num` is less than the length of the array?

3. Modifying Code: Change the provided code to create your own random links.

Use the file [random.html](#) to do the following exercises.

In this exercise you will be creating a link that will be randomly generated from a list of your favorite links. For this exercise you will be given some code to work with. As you surf the web you are likely to encounter many different pages written in JavaScript that you like. The challenge is making sure you know how to use that script in your own pages. The best way to test something new is to create a blank page that has only the elements necessary for you to test your script. You will be using random.html to do so.

Your task will be to modify the given code so that when your webpage loads, a random link to one of your favorite sites is generated on the page. You will need to include at least 4 different locations that the random link will point to.

Here is the code you have to work with:

```
<script>

var random = Math.random();
var urlArray = new Array(3);

urlArray[0] = http://www.ischool.washington.edu/;
urlArray[1] = http://www.cs.washington.edu/;
urlArray[2] = http://www.washington.edu/;

if (random < .33)
    {
        document.write("<a href=\"\" + randomurl[0] + \"\"> Random Link
</a>");
    }
else if (random > .33 && random < .66)
    {
        document.write("<a href=\"\" + randomurl[1] + \"\"> Random Link
</a>");
    }
else if (random > .66 && random < 1)
    {
        document.write("<a href=\"\" + randomurl[2] + \"\"> Random
Link</a>");
    }

</script>
```

Already you can see some problems. Some of the syntax may be confusing to you and you might not know why the author chose to do it the way they did. For example, what is the \ doing before the quotation mark in the anchor tag? You may or may not know that this allows you to tell JavaScript to ignore the next character

and print it as is. However, you do know that the script works as is, so you wouldn't need to modify that part. Another odd choice on the part of the author is the lack of a final "else" statement that tells the script what to do if none of the above are true. In this script, that else is not necessary, so the author has skipped using it.

There are even some obviously poorly written elements. The `<script>` tag does not indicate that we are using JavaScript, nor are the comments that are necessary for older browsers to ignore your JavaScript. The script works as it is, but you know that it is good practice to include these items.

- J. Copy this code into the webpage provided for you, random.html. You will need to decide where to put the code.
- K. Before you change anything, save your work and check to see if the code works. If it does not, did you copy all of the code? Did you accidentally erase any of the HTML already written in your document?
- L. Add another element to the array. Read the book, the lessons or review Exercise 1 for help doing this.
- M. Change the conditional to reflect 4 possible outcomes instead of the 3 that are there now.
- N. Save your work. Reload the page several times and determine that all 4 links show up eventually. Remember, this is random, so it might take a while for some of them to appear.

4. Add your modified code to your lab3.html webpage.

In Lab 3 you created a webpage called lab3.html. One of the requirements for this lab was to link to some favorite websites. Add the code you have modified above to your lab3.html webpage to allow users to go to a randomly selected webpage.

5. Integrate your mouseover.html and array.html JavaScript into lab3.html.

In the same way that you added the code from randomlink.html into your lab3.html webpage, you will add the mouseover image of Red Square, and the array that contains your favorite foods.

- O. Add a picture of Red Square to your lab3.html webpage (if you have not already done so) and cause the image to change to your modified version when you mouse over it.

Notice that not all of the code goes in one place!

- P. Add a header for the picture instructing the user to mouse over the image.
- Q. Add your favorite foods array to your lab3.html webpage. Because you already have a list of your favorite foods, you may want to make sure you have added new ones to your page with this exercise.
- R. Add formatting to explain your array output.