1. Create an Array Object to Hold your Favorite Foods

For this lab, use the file called arrays.html

Concept: Arrays

Arrays are objects that can hold a number of values for related things, covered by a single variable name. When writing an array, you need to know three things:

1. the name of your array;
2. how many elements it will contain; and
3. 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

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

Notice the array elements are numbered starting at 0!

A. Start with a `<script>` tag in the body of your array.html document.

B. Declare your array with the following code:

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

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

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

2. Use Iteration to Print the Values in Your Array to the Web Page

Note

Use your own Array object name, and put in your own favorite foods—unless, of course, they are the same as mine!
If I asked you to print the values of your array to the Web page you might start typing something like:

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

You get the point. To print all your favorite foods to the page, you would need to write each down. 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 the lesson states: *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 our example the statements cause our first array item to be printed to the Web page, then the second array item, then the third, and finally the fourth. The condition to be met to stop our iteration is that the last, or fourth, item prints to the Web page.

**Skill: While Iteration Syntax for JavaScript**

```javascript
while [<stop condition>]
{
  code statement;
  code statement;
}
```

A. Declare a variable to hold a number, and assign it a value of 0.

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

```javascript
var num
num = 0

while()
{
  
}
```
C. Write a stop condition that causes the loop to stop after `num` equals 3.

```javascript
var num
num = 0

while(num <= 3)
{

}
```

D. Write the code statements within the curly brackets to cause your favorite foods to print to the screen.

```javascript
var num
num = 0

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

Notice the variable `num` is used in place of a literal number. This is because each time this loop runs, `num`'s value increases by one, and as a result, prints 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 then or equal to 3? Yes, because num equals 0, so I will perform the code statements in the loop. The array element I'm printing will be the 0 element, and it will print Sushi. Then 1 is added to num, so num equals 1. This is the end of the loop, so I'll start at the beginning again. Is num less than or equal to 3? . . ."

E. 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 . . ."
3. Modifying Code: Change the Provided Code to Create Your Own Random Links

Use the file randomlink.html to do the following steps.

In this part of the lab you will create a link to be randomly generated from a list of your favorite links, using sample code we provide. As you surf the Web, you are likely to encounter many pages you like that are written in JavaScript. The challenge is making sure you know how to use elements of their scripts in your own pages. The best way to test something new is to create a blank page with only the elements necessary to test your script. You will be using randomlink.html to do this test.

Your task is to modify the given code, so when your Web page loads, a random link to one of your favorite sites is generated on the page. You will need to include at least four different locations, or URLs, to which the random link can point.

Here is the code:

```html
<script>
var random = Math.random();
var randomurl = new Array(3);
randomurl[0] = "http://www.ischool.washington.edu/";

if (random < .33) {
  document.write("<a href=" + randomurl[0] + "">Random Link</a>"");
}
else if (random > .33 && random < .66) {
}
else if (random > .66 && random < 1) {
}
</script>
```
You can already see some problems. The `<script>` tag does not indicate we are using *JavaScript*, nor are there the comments necessary for older browsers to ignore your *JavaScript*. The script will work as is, but it is good practice to include these items.

A. Copy this code into the provided Web page, `randomlink.html`. You will need to decide where to put the code.

B. Before you change anything, save your work, and check to see if the code works. If it does not, ask yourself, “Did I copy all the code? Did I accidentally erase any of the HTML already written in my document?”

C. Add another element to the array. Read the book or online lesson narrative, or review Step 1 for help doing this.

D. Change the conditional to reflect four possible outcomes instead of three.

E. Save your work. Reload the page several times, and determine whether all four links show up eventually. Remember, this is random, so it might take a while for some of them to appear.

F. If your script doesn’t work, take a peek at the working solution, either [here](#), or from the main Lab 10 Web page.

4. **Add Your Modified Code to Your Web Page**

In Lab 3 you created a Web page entitled `lab3.html`. One of the requirements for that page was to link to some of your favorite Web sites.

Add the code you modified earlier to your `lab3.html` Web page to allow users to go to a randomly-selected Web page.

5. **Integrate Your *JavaScript* into Your Web Page**

In the same way you added code from `randomlink.html` to your `lab3.html` Web page, you will add the `mouseover` image of Red Square, and the array containing your favorite foods.
A. Add a picture of Red Square to your lab3.html Web page if you have not already, and cause the image to change to your modified version when a user mouses over it.

**Notice:** not all the code goes in one place!

B. Add a header for the picture instructing the user to mouse over the image.

C. Add your favorite foods array to your lab3.html Web page. Because it already has a list of your favorite foods, make sure you've added new ones.

D. Add formatting to explain your array output.