Loop Code and Forms:

So far we've used an if condition to control when we do particular statements. In an if condition, you say something like,

```
if (var1 == "cat")
{
   document.write("<p>It's a cat!</p>")
}
```

In this case the document.write code is only executed when (var1 == "cat") is true (meaning var1 does, in fact, hold "cat").

A variant on this is a loop. With a loop, javascript ***continues to do the statements as long as the condition remains true***. So, for instance, in javascript a loop would look like this:

```
line 1.   countvar = 1;
line 2.   while (countvar < 5)
line 3.   {
line 4.         document.write("<p>really </p>");
line 5.         countvar = countvar + 1;
line 6.   }
line 7.   document.write("<p> cute! </p>");
```

In the above code:
line 1. the number 1 is first placed into the variable countvar.
line 2. while the value inside of countvar is less than 5, we do the code inside the { and }, which is:
line 3. defining the start of the code that will be done repeatedly until the countvar is >= to 5
line 4. the word "really" is written to the html document
line 5. the value inside the countvar increases by 1 (to hold 2).
line 6. This is the end of the code that is done repeatedly until countvar is >= to 5. At this point, we go back to line 2 and check to see: is the value in countvar < 5? If it is (if this condition is true), we do lines 3 through 6 again, so the word "really" will be written again, and the countvar will change to 3.
Again, we go back to line 2 (remember, it's a loop. It happens over and over again, until countvar >= to 5). Again we check, and find that countvar is still less than 5, so we do lines 3 through 6 again, the word really is written out again, and countvar becomes 4.
Again, we check. Is countvar <5? Yes, so we write "really", and change countvar to 5
We check. is countvar < 5? NO! countvar now is equal to 5. So we don't do the code in lines 3-6 again. We now skip down to line 7
line 7. When the loop is done and the loop's condition is no longer true, we write the word "cute!" to the web page.

That's a loop. Whatever code is between the { and the } will happen again and again until the condition is no longer true. Try it:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
      <title> First Javascript </title>
      <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
      <script type = "text/javascript">
        function JS(countvar)
            while (countvar < 5)
            {     document.write("<p> really </p>");
                  countvar = countvar + 1;
            }
```

```
            document.write("<p> cute </p>");
        }
        </script>
    </head>
    <body>


    <p id = "here" onClick = "JS(1)">Click to see the secret message </p>

    </body>
    </html>
```

Question: (DO NOT TRY THIS!!!) What would happen if you didn't include the line "countvar = countvar + 1" inside the loop?

Another example of a loop would be:

```
continuevar = true
while (continuevar == true)
{
    count = count + 1
    document.write("<p>Because </p>")
    continuevar = confirm("But Why?")
}
```

In this case, the loop (all the code between { and } will continue to happen as long as the value inside of continuevar is true. So the value inside continuevar will change whenever the confirm box pops up and the user either hits "ok" (which is true), or "cancel" (which is false). Try this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
   <script type = "text/javascript">
       function loopfun()
       {
          continuevar = true
          while (continuevar == true)
          {
             document.write("<p> Because </p>");
             continuevar = confirm("But Why?")
          }
          document.write("<p> Because I said so and that's final! </p>");
        }
      </script>
   </head>
   <body>
        <p onclick = "loopfun()">Why?</p>
   </body>
</html>
```

Now let's put this all together. We're going to write a javascript that cycles through potential pictures for a background image on your page. When the user decides on an image (By clicking cancel when asked, "Do you want to see another picture?") the current image being displayed becomes the background image.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
   <script type = "text/javascript">
      ls = new Array()
      ls[0] = "bg1.jpg"
      ls[1] = "bg2.jpg"
      ls[2] = "bg3.jpg"
      ls[3] = "bg4.jpg"
      ls[4] = "bg5.jpg"
      ls[5] = "bg6.jpg"
      count = -1
      function loopforbackground(idholder, idholder2)
      {                                                    /* COMMENTS ON CODE */
        document.body.style.backgroundImage = “”           /* sets the background image to blank */
        document.getElementById(idholder2).innerHTML = " "  /* sets the test for idholder2’s element to blank */
        continuevar = true                                 /* INITIALIZE continuevar to hold true */
            while (continuevar == true)                    /* the condition checked to know whether to keep doing the loop */
            {                                              /* indicates where code in loop starts */
              count = count + 1                            /* increase the count by 1 (used In the array) */
              if (count == ls.length)                      /* (Still in loop) checks if count is equal to the length of the array */
              {                                            /* Start of code done if the if condition is true */
                 count = 0                                 /*  set count to 0 (done if count was equal to the array length */
              }                                            /* end of if statement (but still in loop)*/
              document.images[idholder].src = ls[count]    /* displays the image at ls[count] */
              continuevar = confirm("Do you want to see another picture?")
                            /* asks: do you want to see anotherpicture?, and puts your answer (true or false) into continuevar */
          }                 /* end of loop – we go back and check if the while condition is true or not */
          document.body.style.backgroundImage = "url(' " + ls[count] + " ')"
                           /* outside of loop, only done when continuevar is false.  We’re displaying the background image here*/
          document.getElementById(idholder2).innerHTML = "click here for other background options"
                           /* changing text in idholder’s innerHTML */
      }
    </script>
  </head>
  <body>
     <img width = "500" height = "374" alt = "area in which pics will be displayed" id = "display" />
     <h3 id = "firsttext" onClick = "loopforbackground('display','firsttext')">
          Click here to see potential pics... </h3>
  </body>
</html>
```

We now have a loop (the while loop). The code inside the while loop keeps happening while continuevar holds a true value. What happens is the count value is increased by 1, then we check to see if we’re at the end of the array, and if we are, we reset the count back to 0. The image is then displayed (the image in the array at count). You’re asked, “do you want to see another picture?”. You click “okay” (which is equivalent to typing ‘true’) or “cancel” (which is equivalent to typing ‘false’). If you hit “okay”, the continuevar holds ‘true’. We’re now at the end of the code inside the loop, so we go back to the beginning of the loop and check – does continuevar hold true? If it does, we do all the code inside the loop again. So count increases by 1 again, we check to see if count has hit the end of the loop, and if it has, we set it back to 0. We then display the next image. And we ask (again), “Do you want to see another picture?” You either hit “okay” (true) or “cancel” (false). We do all the code inside the loop until you hit “cancel”. When you hit “cancel”, continuevar now holds ‘false’, so when we check the loop’s condition (is continuevar == true?), the answer is “no”, so we go to the code below the loop and execute that.

We now set the background image to be whatever image in the array we chose (determined by the number inside of the count variable). Then we set the text inside 'firsttext' to say, "Click here for other background options.

Note: What's going on here?

```
document.body.style.backgroundImage = "url(' " + ls[count] + " ')"
```

Let's break it down. First, we're changing the body of the html document's background image. If you were changing the background image of the body to "bg3.jpg" in css, you'd use the following line (remember?):

```
body {
        background-image: url('bg3.jpg')
}
```

In JavaScript the equivalent would be:

```
document.body.style.backgroundImage = "url('bg3.jpg')"
```

In the case of our code, however, the background image is the image in the array at whatever number is currently inside of count, or ls[count]. If we did this:

bad: `document.body.style.backgroundImage = "url('ls[count]')"`

Javascript would try to set the background to an image called "ls[count]" (and not what is inside ls[count]). Clearly that would not work because there is no image called "ls[count]". So what we do instead is join together three different things: The string between the first set of double quotes: "url(' " , the value inside of ls[count] and string between the second set of double quotes, " ')" We join these three strings together using the + sign. At this point we'd have:

```
document.body.style.backgroundImage = "url(' " + ls[count] + " ')"
```

That should do it!

Part 2: Forms:

So far you've created user interfaces that have allowed the user to interact with your web page in a number of ways: By clicking on text or images (using onClick), by running your mouse over something (onMouseOver), by running your mouse off of something (onMouseOut), by using the prompt pop-up box (in which the user typed something in, and you stored the result in a variable), and through the confirm pop-up box, in which the user either clicked on "okay", and the variable held "true", or the user clicked on "cancel" and the variable held "false". These are all ways in which the user is interacting with your web page. You're creating an interactive web page.

Another way in which users can interact with your web page is through forms. Remember forms? In HTML, you use the <form> </form> tags to surround the form, and then enter elements like text fields, textboxes, radio buttons, etc. A basic html form might be (and it would go in the body of your web page):

```
<form action="" method="GET">
        <input type="text" id="firsttextbox" size = "100" value=" " ><br />
        <input type="button" id="button1" value="Click here " >
</form>
```

This form has 2 elements: a text box, with the unique id of "firsttextbox" and a width of 100, and a button, with the unique id of button1 and the value (the text that shows up on the button) of "click here". Try it.

In order to be able to use the data in a form, we want to send the entire form into a function. Up to this point, we've sent numbers, ids, and names of pictures into functions. Now we want to send an entire form. In javascript the form is considered as a single object.

We are going to use the button inside the form to call the function, and I'm going to change the text inside the text field in the form. To do this, let's first add the call to the function inside the button:

```
<form action="" method="GET">
        <input type="text" id="firsttextbox" size = "100" value=" original text in textbox " ><br />
        <input type="button" id="button1" value="Click here " onClick = "ChangeText(this.form, 'firsttextbox')">
</form>
```

Note that to send the entire form into a function's parameter, you use this.form, without quotes around it. this.form is the form object – it includes everything between the opening and the closing <form></form> tags. I also sent into the function the id of the text field, so I can change the text inside it.

Now to write the function ChangeText, you'd create a function in the head section (inside a script) that has two parameters, the first to hold the form, and the second to hold the id 'firsttextbox'. So it might look something like this:

```
<head>
        <script type = "text/javascript">
                function ChangeText(formpar, idpar)
                {
                }
        </script>
</head>
```

Now the formpar holds the entire form, and the idpar holds 'firsttextbox'.

We want this function to change the value inside the textbox. Right now it's "original text in textbox" Let's change it to say "Great job clicking!" after you click on the button (we've already got the onClick call to the function in place).

To change the value of the textbox with the id "firsttextbox", inside the function, you'd do the following:

```
formpar[idpar].value = "Great job clicking!"
```

This is the same as saying,

```
formpar['firsttextbox'].value = "Great job clicking!"
```

Which is the same as saying, Take the form, and look at all the elements that are in the form. Find the one that has the id 'firsttextbox', and then change that element's value.

You can also change the element's size and type, but NOT it's id (you can, but DON'T)

Put it all together, you should get:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />
        <script type = "text/javascript">
```

```
                function ChangeText(formpar, idpar)
                {       formpar[idpar].value = “Great job clicking!”
                 }
            </script>
  </head>
  <body>
        <form action="" method="GET">
                <input type="text" id="firsttextbox" size = "100" value=" original text in textbox " ><br />
                <input type="button" id="button1" value="Click here " onClick = “ChangeText(this.form, ‘firsttextbox’)”>
        </form>
  </body>
</html>
```

Try it.

You can also read in values from the form using a text box: To do this, you create a variable and set the variable to hold the element with the id ‘inputbox’s value, as follows:

```
        <script type = "text/javascript">
                function readText (formpar,idpar) {
                        inputvar =formpar[idpar].value
                        formpar[idpar].value = "You typed: " + inputvar
                }

        </script>
        </head>
        <body >
                <form action="" method="GET">
                        Enter something in the box: <br />
                        <input type="text" id="inputbox" value=""><br />
                        <input type="button" id="button1" value="Read" onClick="readText(this.form,’inputbox’)">
                </form>
        </body>
```

Try this and make sure you understand it. inputvar holds the value from the form’s text field with the id inside idpar, or inputbox. You then change that value to be “You typed: “ + whatever you just typed into the box, because that is what is inside the variable inputvar.

Now let’s make it a bit more interesting. Remember the magic 8 ball when you were a kid? The magic 8 ball was a ball with black ink in it and an octagon with 8 sides inside it. Each side had a different answer, like “Definitely, yes!”, or “It might be so.” or “odds don’t look good”, etc. You would ask the magic 8 ball a question, then shake it up and see the answer to your question. We’re going to create a version of that with a form.

The form will have 2 text fields(each with a unique id) and a button. The first text field is where the user will type their question, and the second text field where the answer to the question will appear when the user clicks on the button. So the form will look like this:

```
        <form action="" method="GET">
                 <input type="text" id="questiontext" size = "100" value="Type your question here."> <br />
                <input type="text" id="answertext" size = "100" value="Your answer will appear here"><br />
                <input type="button" id="button1" value="Click here to get answer"
                 onClick="magic8(this.form,'questiontext','answertext')">
        </form>
```

Modify the form above so that it looks like this. Notice that we've now got 2 text fields, one with the id "questiontext" and one with the id "answertext". Both of these ids, along with the form are sent into the function called magic8.

Now we need to create the proper javascript. To start, you need to create an array with 8 strings, each representing a possible answer:

```
<script type = "text/javascript">
        ls = new Array()
        ls[0] = "Yes"
        ls[1] = "Maybe"
        ls[2] = "Outlook not looking good"
        ls[3] = "hopeful"
        ls[4] = "definitely not"
        ls[5] = "possibly"
        ls[6] = "Without a doubt"
        ls[7] = "Very doubtful"

        function magic8(formpar, idholder, idholder2)
        {
        }
</script>
```

What we want to do is to first randomly choose one of the answers in the array, and set the value of the text field with the Id 'answertext' to be that answer. So first we need to generate a random number the length of the array, and then we need to set the value of the text field with the id answertext to that value:

```
<script type = "text/javascript">
        ls = new Array()
        ls[0] = "Yes"
        ls[1] = "Maybe"
        ls[2] = "Outlook not looking good"
        ls[3] = "hopeful"
        ls[4] = "definitely not"
        ls[5] = "possibly"
        ls[6] = "Without a doubt"
        ls[7] = "Very doubtful"

        function magic8(formpar, idholder, idholder2)
        {
            answervar= Math.floor(Math.random() * ls.length);
             formpar[idholder2].value = ls[answervar]
        }
</script>
```

And finally, we want to change the value of the question's text field to "Type your next question here". Or, more specifically, we want to change the value of the form's text field with the id "questiontext" to be equal to "Type your next question here." Your code should look like this:

```
<script type = "text/javascript">
        ls = new Array()
        ls[0] = "Yes"
        ls[1] = "Maybe"
        ls[2] = "Outlook not looking good"
```

```
        ls[3] = "hopeful"
        ls[4] = "definitely not"
        ls[5] = "possibly"
        ls[6] = “Without a doubt”
        ls[7] = “Very doubtful”

        function magic8(formpar, idholder, idholder2)
        {
            answervar= Math.floor(Math.random() * ls.length);
             formpar[idholder2].value = ls[answervar]
             formpar[idholder].value = “Type your next question here”
        }
</script>
```

Try this. Kinda fun, huh. Think of all the fun things you could now create.

Project 2:

BlackJack (sort of) (100 pts)

Overall Game: You’re going to write a web page in which you get asked if you want another card continuously, and, if you say yes, a new card will be displayed in a different place on the web page. You’re trying to make the cards come as close to but not go over 21.

Steps:

1. Create an array of images. The images (pictures) should be of the numbers 1 through 11
   (Note: the image of number 1 will go in the array at 0 because we always start our arrays at 0)
2. Write an html page with space for 12 images (probably a table with images in it.) Each image should have a unique id.
3. Create a counter variable.
4. Write a function GetCards()
5. Inside GetCards, write a confirm asking, “do you want a card”.
6. Write a loop. Your loop will continue while the answer from confirm is true
7. Inside your loop, generate a random number between 0 and the length of your image array.
8. Slightly challenging part:
    1. Now (still inside your loop), you will create an if statement, that goes something like this:
    2. If the counter is equal to 0, use the random number generated to display the image from the array (like we’ve done before). You will display the image in the first id on the web page.
    3. If the counter is equal to 1, you’ll display the image in the second id on the web page.
    4. If the counter is equal to 2, you’ll display the image in the third id on the web page,
    5. …
    6. If the counter is equal to 11, you’ll display the image on the twelfth id on the web page.
9. Now you will use the confirm to ask, “do you want another card”.
10. End your loop
11. End your function

Keep track of the total number generated by adding each random number generated together. If the number goes over 21, modify text on the page to say, “You lose!”

Extra Credit: BMI (25 pts)

This project allows you to practice with forms. Create a form with 4 text fields and a button. The first text field will be where you enter your weight in pounds. The second is where you'll enter your height in inches. Thus you will have to read both these values into 2 separate variables (see the section on forms where you take the values of forms and put them into variables). So now you've got a variable that holds someone's weight, and a separate variable that holds someone's height. You use these two variables to calculate the bmi, using the formula:

bmi = (weight * 703)/(height*height)

(in javascript, and most programming languages, * means to multiply, and / means to divide, so you'd calculate bmi in javascript literally the way it is written here, with your weightvariable substituted for weight, and your heightvariable substituted for height).

In the third text field, write out "Your bmi is: " + bmi to the value of that text field.

And finally, in the fourth text field, write out: "You are obese" if the bmi is over30, "You are underweight" if the bmi is less than 18, and "You are normal, otherwise.