WDV221 Intro to Javascript

**Project-5 Loops**

Whenever you say “I need to do this seven times” you need to use a loop. A LOOP allows you to perform the same action multiple times. The logical name for loops is Repetition. But we call them loops. All programming related languages such Javascript, PHP, Actionscript, etc. have loops.

There are three key pieces of information that you need to remember when working with a loop. If you think about what you need done you can identify the following pieces.

The **Loop Control Variable**: This variable controls the number of times a loop will run. We check the value of this variable to know when to end the loop.

**Check for the Ending Condition**: This is a comparison (IF statement) that ends the loop at the right time. This conditions compares the Loop Control Variable to a value or condition that tells us the loop is finished.

**Update the Loop Control Variable**: Each time through the loop you should update or allow the loop control variable to be updated. If this variable is never changed the loop condition will never end the loop. This results in the ‘endless loop’ effect.

**Loop Control Variable**

There are several ways to use variables to control a loop. Many loops will run a certain number of times. These are called **Counter Controlled** loops. These loops can be based upon the number of rows in an array, the number of records in a row, the number of output lines to display, etc. Counter Controlled loops usually add 1 to a counter variable every time the loop runs. This counter acts as the Loop Control Variable.

Some loops will run until a certain condition is met. These loops are called **Sentinel Controlled** loops. For example: This loop will run until the number of rows is greater than 10. This loop will run until all of the teams have been processed. The loop will run until the customer is done shopping. The loop control variable is usually something besides a number when using a Sentinel Controlled loop.

**Check for the Ending Condition**

You need two pieces of information for this to work. You have to have the Loop Control Variable and you have to have some way of knowing how many times the loop is supposed to run. Sometimes you know the number of times to run, sometimes we base it off of another piece of information such as the length of the array or a response from the user telling us they need to add another item, enter another value, etc.

**Pre-test loops:** These loops will check the end condition BEFORE processing the body of the loop. This ensures that the loop is supposed to run before any processing is done. Most loops are pre-test loops. This includes the FOR and the WHILE loop. It is possible that the loop will not run if the end condition is met before the loop starts.

**Post-test loops:** These loops will check the end condition AFTER processing the body of the loop. This type of loop will always perform the body of the loop at least once before checking the end condition. The DO WHILE loop is an example of this format.

**Update the Loop Control Variable**

The Loop Control Variable is critical to this process. Once you have created the variable you need to initialize it to a beginning value. In many cases this is 0, but it can be other values based upon what the loop needs to do. You MUST update the Loop Control Variable inside the body of the loop. Failure to properly update the Loop Control Variable can create an ‘infinite loop’. This is a loop the never ends because we have written the wrong ending condition or failed to properly update the Loop Control Variable.

For Loop

This is a counter controlled loop. It is best used when you know the number of times you need the loop to run. The For loop has all three pieces required for a loop. The general format of a for loop looks like this.

for (*initialize the loop control variable ; check the end condition ; update the loop control variable* )

{

*body of the loop; //These commands will be processed every time the loop runs*

}

Working example:

for( var i=1 ;i<=5 ; i++)

{

document.write(“<p>Hello</p>”);

}

//Print out the word Hello five times

for( var i=0; i<names.length; i++)

{

document.write(“<p>Team: “ + names[i] + “</p>”);

}

//Print out a list of team names from the names array

While Loop

This is a sentinel controlled loop. It is best used when you need to run a loop until a certain condition is met. If you don’t know how many times it will run then you are using a Sentinel controlled loop.

Example situations: I need to add these numbers until the user is done. I am going to print out the results for each product the user enters. I am going to keep adding to the shopping cart until the user selects “Checkout”. Etc.

Setting up a While loop takes more planning than the For loop. You have to decide on the loop control variable and figure out the initial value. You have to determine when the loop will end. You also have to figure out how to update the loop control variable.

while(check end condition)

{

*body of Loop //this will run as long as the condition is true*

}

Examples:

This loop will run as long as the rowNumber is less than 10.

var rowNumber = 1; //create and initialize loop control variable

while(rowNumber <= 10) //Run the loop as long as the rowNumber is less than or equal to 10

{

Document.write(“<p>Row number: “ + rowNumber + “</p>”);

rowNumber++; //update the loop control variable inside the body of the loop

}

This loop will run as long as the customer has more products to add. When they are done they will type ‘no’ into the prompt and that will change moreProducts from ‘yes’ to ‘no’.

var moreProducts = “yes”; //create and initialize loop control variable LOOP CONTROL VARIABLE

while(moreProducts == “yes”) //Run the loop until condition is false CHECK CONDITION

{

addProducts(); //run this function until the user does not have any more products

moreProducts = prompt(“Do you have more Products? yes/no); //Update loop control

}