**Interactive Development with JavaScript – Module 6**

**Iteration**

Iteration statements in JavaScript, as in other languages, are used to allow statements to be repeated. There are several different types of loops, however the basic concept is the same, they are to repeat code as long as needed, or execute no code if not needed. The types of loops JavaScript currently supports are while, do/while, for, for/in, and, for/of.

* while
  + A while loop is a statement or block of code where execution is based on condition. The condition is evaluated (tested). If the evaluation is true, the following statement or block is executed. When the execution is complete the loop will again evaluate the condition and execute if the test is true. When the condition is evaluated with the results of false, the loop is then exited.
  + A while loop is often referred to as a pre-test loop as the condition is checked before the loop executes.

Syntax:

while (condition) {

// statement or block of code

}

Example 1:

var i = 1;

while (i <= 10) {

document.writeln(i);

++i;

}

Output: 1 2 3 4 5 6 7 8 9 10

Example 2:

var i = 1;

while (i <= 10) {

document.writeln(i);

i += 2;

}

Output: 1 3 5 7 9

* do/while
  + A do/while loop is a statement or block of code that is executed before a condition is tested. After execution the condition is evaluated (tested). If the evaluation is true, the loop will execute again. When the execution is complete the loop will again evaluate the condition and execute if the test is true. When the condition is evaluated with the results of false, the loop is then exited.
  + A do/while loop is often referred to as a post-test loop as the condition is checked after the loop executes.

Syntax:

do {

// statement or block of code

}while (condition);

Example 1:

var i = 1;

do{

document.writeln(i);

++i;

}while (i <= 10);

Output: 1 2 3 4 5 6 7 8 9 10

Example 2:

var i = 1;

do{

document.writeln(i);

i += 2;

}while (i <= 10);

Output: 1 3 5 7 9

* for
  + A for loop is a statement or block of code where execution is based on a condition. The loop is first unitized. The condition is then checked to see if true. If the evaluation is true, the following statement or block is executed. When the execution is completed the loop will then adjust the condition, and again evaluate the condition. The loop will then execute if the condition is true. When the condition is evaluated with the results of false, the loop will exit.
  + A for loop is similar to a while loop as it is tested before the loop executes.
  + Syntax:

Syntax:

for (Section\_01; Section\_02; Section\_03) {

// statement or block of code

}

Section\_01 is executed only once before execution of the loop starts.

Section\_02 holds the condition for the loop to execute. This section is checked each time before the execution of the loop. If the condition is true, the loop executes. When the condition reaches a point where it is false, the loop then exits.

Section\_03 is executed after the loop and is designed to alter the loop condition.

Example 1:

for(var i = 0 ; i < 5 ; ++i){

document.writeln(i);

}

Output: 0 1 2 3 4

Example 2:

for(var i = 0 ; i < 10 ; i += 2){

document.writeln(i);

}

Output: 0 2 4 6 8

* + The next example is to show all sections of a for loop are optional, however for loops with an empty header section should be replaced with a while loop.

Example 3:

var i = 0;

for( ; ; ){

document.writeln(i);

++i;

if(i == 10){

break;

}

}

Output: 0 1 2 3 4 5 6 7 8 9

**Loop Statements**

These types of loop statements are the only method for exiting an iteration of a loop prior to its normal completion. Generally, it is always best to have both a break; and a continue; on a separate line.

* break
  + The break statement is a method for exiting a loop.
  + As we remember a break can also exit a switch which we saw in the last module.
  + Lastly a break can exit a labeled block which is beyond our scope.

Example 1:

var i = 0;

while (i < 10) {

++i;

if (i == 5) {

break;

}

document.writeln(i)

}

Outputs: 1 2 3 4

Example 2:

for (var i = 1; i <= 10; ++i){

if (i == 5) {

break;

}

document.writeln(i);

}

Output: 1 2 3 4

* continue
  + The continue statement is a method for exiting one iteration of a loop, and then continue with the next iteration.

Example 1:

var i = 0;

while (i < 10) {

++i;

if (i == 5) {

continue;

}

document.writeln(i);

}

Outputs: 1 2 3 4 6 7 8 9 10

Example 2:

for (var i = 1; i <= 10; ++i){

if (i == 5) {

continue;

}

document.writeln(i);

}

Output: 1 2 3 4 6 7 8 9 10

* Duration
  + Finite (Limited)
  + A finite loop is one that executes until a reachable false condition results.
  + Infinite (Unlimited)
  + An infinite loop is one that executes unstopped as the false condition is never reached. An infinite loop can cause both browser and system problems. Most infinite loops are a result of coding errors and help to reinforce a sufficient level code of testing.

Example:

for(var i = 20 ; i > 10 ; i += 2){

document.writeln(i);

}

Chart

Description automatically generated