**Interactive Development with JavaScript – Module 8**

**Built-In Functions**

Functions are blocks of code designed to perform a specific action. Methods are blocks of code (functions) designed to perform actions on an object. To access a method, you will use the syntax structure of the class name followed by a period, followed by the method name (function name), followed by parentheses either empty or filled with arguments. This lesson will cover predefined methods and in the next lesson we will cover writing functions.

The basic rules for invoking (calling) methods are:

* Syntax
* objectName.methodName();
* objectName.methodName(argument);
* objectName.methodName(arg1, arg2, agr3);

There are several JavaScript classes holding methods. For this lesson we will cover some of the methods in the following classes: Object, Math, Number, Array, and String.

**Object**

JavaScript is an Object Oriented (OO) language. The basic OO language capabilities are encapsulation, aggregation, inheritance, and polymorphism. Objects may hold variables referred to as properties and functions referred to as methods. The Object() constructor may be used to create and initialize an object, a topic we will cover more in the next lesson.

* Encapsulation
  + Encapsulation is information hiding in that the inside actions of a class may be altered without changing the interface.
* Aggregation
  + An aggregate object is one containing other objects. For example, a computer class contains a system board, memory, storage device, and input and output devices.
* Inheritance
  + A superclass/base class is extended/inherited by a subclass/derived class with the subclass holding similar implementation. A subclass is a base class. For example, a Ford is a vehicle, however a vehicle is not always a Ford.
* Polymorphism
  + A single interface enabling different responses based on the data type. For example, a method toString() may be used to return a String containing information about a class with different results based on differing class types.

**Math**

The JavaScript static Math object provides methods (functions) and properties to be used for operations performing mathematical tasks on numbers. Because this object is static we are not required to create an instance of the object. We can simply access a method using the class name followed by a period and the method name, example Math.round(7.8). Likewise a Math class property can be accessed in the same manner, Math.PI.

* Some of the Math Methods
  + abs()
    - This method will return the absolute value of a number
    - document.write(Math.abs(7));

// Output 7

* + - document.write(Math.abs(-7));

// Output 7

* + ceil()
    - This method will return the smallest integer value greater than or equal to a number.
    - document.write(Math.ceil(7.003));

// Output 8

* + - document.write(Math.floor(7.003));

// Output 8

* + floor()
    - This method will return the largest integer value less than or equal to a number.
    - document.write(Math.floor(7.003));

// Output 7

* + - document.write(Math.floor(7.893));

// Output 7

* + round()
    - This method will return the value of a number rounded to the nearest integer, a whole number that is not a fraction.
    - document.write(Math.round(7.893));

// Output 7

* + - document.write(Math.round(7.893));

// Output 8

* + max()
    - This method will return the largest value of 0 or more numbers.
    - document.write(Math.max(72, 40, 92, 14));

// Output 92

* + - document.write(Math.max(7.3, 9.3, 1.2));

// Output 9.3

* + - document.write(Math.max());

// Output – Infinity (Negative Infinity)

* + min()
    - This method will return the smallest value of 0 or more numbers.
    - document.write(Math.min(72, 40, 92, 14));

// Output 14

* + - document.write(Math.min(7.3, 9.3, 1.2));

// Output 1.2

* + - document.write(Math.min());

// Output – Infinity (Positive\_Infinity)

* + pow()
    - This method returns a base multiplied to the exponent power, baseexponent.
    - document.write(Math.pow(3, 3));

// Output 8

* + - document.write(Math.pow(3, 3));

// Output 27

* + sqrt()
    - This method returns a square root of a value.
    - document.write(Math.sqrt(4));

// Output 2

* + - document.write(Math.sqrt(9));

// Output 3

* + random()
    - This method returns a random value starting at 0 going to a value less than 1.
    - document.write(Math.random());

// Output 0 <= x < 1

* + Invoking more than a single method.
    - // Output 1 – 6
    - document.write(Math.floor((Math.random() \* 6) + 1));
    - // Output 1 - 100
    - document.write(Math.floor((Math.random() \* 100) + 1));
* Some of the Math Properties
  + PI
    - PI is the ratio of the circumference of a circle to its diameter, approximately 3.141592653589793.
    - Math.PI
    - document.write(Math.PI);
  + SQRT1\_2
    - SQRT1\_2 is the square root of ½. This is approximately 0.7071067811865476.
    - Math.SQRT1\_2
    - document.write(Math.SQRT1\_2);
  + SQRT2
    - SQRT2 is the square root of 2. This is approximately 1.4142135623730951.
    - Math.SQRT2
    - document.write(Math.SQRT2);

**Number**

The Number object is to be used to help working with numbers. Numeric primitive literals do not have properties or methods, however the Number object allows primitives to have properties and methods.

* Some of the Number Methods
  + toFixed()
    - Returns the number of digits to the right of the decimal point.
    - var x = 7.34251678;
    - document.write(x.toFixed(2));

// Output 7.34

* + - document.write((7.34251678).toFixed(2));

// Output 7.34

* + - document.write((7.8).toFixed(2));

// Output 7.80

* + toString()
    - Returns the numeric value in a String data type.
    - document.write((555).toString() + (55).toString() +
    - (5555).toString());

// Output 555555555

* + - Helps to see the difference:
    - document.write(ss1.toString() + ss2.toString() +
    - ss3.toString());

// Output 555555555

* + - document.write(ss1 + ss2 + ss3);

// Output 6165

* Some of the Number Properties
  + MAX\_VALUE
    - Largest value a number in JavaScript can hold.
  + MIN\_VALUE
    - Smallest value a number in JavaScript can hold.
  + NaN
    - A value that is not a number.
  + NEGATIVE\_INFINITY
    - A value less than MIN\_VALUE
  + POSITIVE\_INFINITY
    - A value greater than MAX\_VALUE

**Array**

The Array object is for storing multiple values. Items in the array can be accessed using an index with the first element being 0 and the last element being the length – 1.

* Some of the Array Methods
  + concat()
    - Returns a new array composed of two or more arrays without changing the existing arrays.
    - var array\_01 = [2, 4, 6];
    - var array\_02 = [3, 5, 7];
    - var array\_03 = [7, 8, 9];
    - var allArrays = array\_01.concat(array\_02, array\_03);
    - document.write(allArrays);

// Output 2,4,6,3,5,7,7,8,9

* + indexOf()
    - Used to locate an element in an array, returns the first indexed location found, returns -1 if the value is not found.
    - var array\_01 = [2, 4, 6, 8, 1, 3, 5, 7, 9];
    - document.write(array\_01.indexOf(2));

// Output 0

* + - document.write(array\_01.indexOf(8));

// Output 3

* + - document.write(array\_01.indexOf(8));

// Output 8

* + toString()
    - returns a string of the array elements.
    - var array\_01 = [7, 2, 4, 6, 2, 3, 5, 7, 8, 9];
    - document.write(array\_01.toString());
  + sort()
    - The method returns a sorted array.
    - var array\_01 = [7, 2, 4, 6, 2, 3, 5, 7, 8, 9];
    - document.write(array\_01.sort());
* One of the Array Properties
  + Length
    - The array object length property holds the number of elements in an Array object.
    - var myArray = new Array(2, 4, 6, 8);
    - var myArray = new Array(2, 4, 6, 8);
    - document.write("The Array length : " +
    - myArray.length);

// Output 4

**String**

The String object is a collection of characters. The String has a few properties and several methods.

* Some of the String Methods
  + charAt()
    - Returns the character at the index provided.
    - var str = new String("Welcome to JavaScript");
    - for(var i = 0; i < str.length; ++i){
    - document.write(str.charAt(i));
  + concat()
    - The concat() method similar to the concat() method of an array. This method returns two or more Strings in a new single String.
    - var str1 = "String 1 ";
    - var str2 = "String 2 ";
    - var str3 = "String 3";
    - var str4 = str1.concat(str2, str3);
    - document.write(str4);
  + toLowerCase()
    - This method converts all of the characters in a String to lowercase and then returns the new String.
    - var str = "Welcome to JavaScript";
    - var lower = str.toLowerCase();
    - document.write(lower);

// Output: welcome to javascript

* + toUpperCase ()
    - This method converts all of the characters in a String to uppercase and then returns the new String.
    - var str = "Welcome to JavaScript";
    - var upper = str.toUpperCase();
    - document.write(upper);

// Output: WELCOME TO JAVASCRIPT

* One of the String Properties
  + length
    - The length property is very similar to the length property of an array in that it holds the number of characters inside the String.
    - Example 1:
    - var welcomeString1 = new String("Welcome to
    - JavaScript");

document.write(welcomeString1.length); ​

* + - Example 2:
    - var welcomeString2 = "Welcome to JavaScript";

document.write(welcomeString2.length);​