* In these section we will cover:
* Functions
* Arrays
* Objects
* Functions will be our main building blocks.
* They will allow us to easily reuse code more than once and not constantly repeat ourselves.
* The general syntax for JS function is:
* function func\_name( parameter 1 , parameter 2 ,…)

{

//Code to be executed

}

* We use the **function** keyword to indicate that we have a function.
* Then we have the option of passing in parameters (we could also not have any parameters).
* Someone using the function could pass in parameters required for the function to do something with.
* **Switch-Case**
* The general syntax of switch-case is:

Switch( expression )

{

case x:

//code block

break;

case y:

//code block

break;

default:

//code block

break;

}

* There can be n number of cases.
* **&nbsp** is used for spacing (like tab).
* **ARRAYS**
* Mutable in nature unlike strings.
* Can hold mixed type of elements in javascript.
* JavaScript arrays are resizable and can contain a mix of different data types
* JavaScript arrays are not associative arrays and so, array elements cannot be accessed using arbitrary strings as indexes, but must be accessed using nonnegative integers (or their respective string form) as indexes.
* JavaScript arrays are zero-indexed.
* JavaScript array-copy operations create shallow copies rather than deep copies.
* **Some of the Array methods**
* **concat()**
* array1.concat(array2) --- Gives concatenation of array1 and array2.
* **flat(Infinity)**
* array.flat(Infinity) --- Gives a flat or 1-D Array.
* **indexOf()**
* array.indexOf(element) --- Gives index of given element.
* **lastIndexOf()**
* array.lastIndexOf(elemet) --- Gives last index of the given element.
* **push()**
* array.push(element) ---- pushes the element inside array.
* **pop()**
* array.pop(element) ---- pops the element inside array.
* **sort()**
* array.sort() --- sorts the array in ascending order.
* **splice()**
* array.splice(index,number) ----- removes the index element

the number of elements needs to be removed.

* **Javascript Objects**
* JS Objects are the hash tables, they store information in a key-value pair.
* In other languages this is sometimes called as Dictionary.
* Unlike array a JS Object does not retain any ordering.
* The typical JS Object is in the form
* A typical JS Object is in the form:

{ key1 : “value one” , key2 : “value2”,…}

* You can access values through their corresponding key.
* **JS Object Method**
* Object methods are essentially functions that are inside of an object.