WEB DEVELOPMENT COURSE

JAVASCRIPT – PART 2

- Strings and their methods
- Numbers and their methods
- Arrays and their methods
- Objects and their methods

STRINGS AND THEIR METHODS

- Strings in JavaScript can be written either between single or double quotes.
- For escaping characters, backslash \ is used.
  For example:
  \' is used to insert a single quote in the string.
  \n is used to make a new line in the string.
  Etc.....
- You can also use single quotes inside double quotes and the opposite.
- All string methods return a new string. They don't modify the original string.

STRINGS AND THEIR METHODS

- Addition sign + can be used to either add numbers or concatenate strings
- Adding a number to a string will result in concatenation
- Refer to the code for more illustrative examples

STRINGS AND THEIR METHODS

- Strings Methods:
- myString.length
- myString.indexOf(substring, start position)
- myString.lastIndexOf(substring, start position)
- myString.search(substring)
- myString.slice(start position, end position)
- myString.substr(Start position, substring length)
- myString.replace(substring, replacement)
- myString.charAt(position)

STRINGS AND THEIR METHODS

- Strings Methods:
- myString.charCodeAt(position)
- myString.toUpperCase()
- myString.toLowerCase()
- myString.split(separator)

NUMBERS AND THEIR METHODS

- Numbers Methods:
- myNumber.toString()
- myNumber.toExponential(precision)
- myNumber.toFixed(precision)
- myNumber.toPrecision(precision)
- Number(myNumber)
- parseFloat(myNumber)
- parseInt(myNumber)

ARRAYS AND THEIR METHODS

- Arrays in JavaScript are treated as objects, therefore, array's elements can have different data types.
  For example:
  let array = ['Ahmed', 20, true];
- We can loop through an array's elements either with for loops or forEach loops. Refer to the code for more illustrations.

ARRAYS AND THEIR METHODS

- Arrays Methods:
- Array.isArray(array)
- myArray.toString()
- myArray.join(joiner)
- myArray.pop()
- myArray.push(element)
- myArray.shift()
- myArray.unshift(element)
- myArray.length

ARRAYS AND THEIR METHODS

- Arrays Methods:
- myArray.splice(position, # deletions, elements.....)
- myArray1.concat(myArray2, myArray3, ...)
- myArray.slice(start position, end position)
- myArray.sort(function)
- myArray.forEach(function(value, index, myArray))
- myArray.map(function(value, index, myArray))
- myArray.filter(function(value, index, myArray))
- myArray.reduce(function(variable, value, index, myArray), initalValue)

ARRAYS AND THEIR METHODS

- Arrays Methods:
- myArray.every(function(value, index, myArray))
- myArray.some(function(value, index, myArray))
- myArray.indexOf(element, start position)
- myArray.find(function(value, index, myArray))
- myArray.findIndex(function(value, index, myArray))

OBJECTS AND THEIR METHODS

- Objects are just like arrays, but their indexes are names, while arrays' indexes are numbers starting from 0.
- An object is declared as follows:
  let object = {property: value, property: value, ...};
- Object's properties can be accessed either by the dot operator or indexes.
- To loop through an object's properties use for ... in loops.
- Objects can use the same methods of arrays since arrays are of type object in the first place

OBJECTS AND THEIR METHODS

- You can add a new property to an existing object as follows:
  if the myObject was equal to {name: 'Ahmed', age: 21}
  then you wrote myObject.gender='male'
  the result will be:
  myObject={name: 'Ahmed', age: 21, gender: 'male'}
- Deletion of a property can be done as follows:
  delete myObject.gender
  Therefor, myObject={name: 'Ahmed', age: 21}

OBJECTS AND THEIR METHODS

- Objects Methods:
- Object.values(myObject)
- for (x in myObject) { ..... }

THANKS!

Any questions?

Feel free to leave your questions in the classroom