ITSE412– Week 3

JavaScript Basic Elements

JavaScript Variables

- Data types: 3 primitive data types (number, string, boolean).
- Composite Data Type: Object and Array.
- Special values: null, undefined, NaN, Infinity.
- Variables: first character must be letter or underscore, subsequent characters can be letter, number, or underscore. (No space, no reserved words).
- Use var or let for variables and const for a constant value

var num = 10; or let num = 10; const x = 20;

JavaScript expressions

- Expression will evaluate to a single value with single data type.

  var a = 3 + 4;

// 3 + 4 is an expression, and will evulate to 7 (data type is number)

- Single line comment: // end with new line
- Multiple line comment: /* start here

  second line */

JavaScript operators

- Operators:
  - Unary operator (one operand): -88, count++, !flag
  - Binary operator (two operands): 7 + 8, num1 < num2
  - Ternary operator (three operands): (condition) ? True expression : False expression ;

JavaScript operators

- 5 categories of operators:
  - String operators: -, +=
  - Arithmetic operators: +, -, *, /, %, ++, --
  - Assignment operators: =, +=, -=, *=, /=, %=
  - Comparison operators: ==, !==, >, >=, <, <=, (automatic data type conversion)

    === (equivalent), !== (not equivalent) (No automatic data type conversion)
  - Logical operators: &&, ||, !
- Conditional Operator: var age = 16;

  (age < 18)? answer=“Child" : var answer="Adult";

JavaScript operators

- Special operators:
  - delete (delete array entry or object)
  - new (use to create new object)
  - this (refer to current object)
  - void (tell JavaScript to evaluate an expression and returns no value)

The order of operations

()
. []
new
function()
! - + ++ -- void delete
* / %
+ -
< <= > >=
== != === !==
&&
||
Conditional Operator
= += -= *= /= %=

JavaScript Functions

- Function declaration:

```
function function_name ()  {
}
```

- Invoke function:

```
function_name();
```

- Use parameters/arguments to pass data to a function:

```
function function_name(fname) {
        console.log(fname);
}
function_name("My Name");
```

- Returning data/value from function:

```
function function_name(a, b) {
        return a + b;
}
```

JavaScript selection

- **If statement:**

```
if (condition) {
        // true statements
} else {
        // false statements
}
```

- **If else if statement:**

```
if (condition)  {
        // if condition is true
} else if (cond2) {
        // if cond2 is true
}  else {
        // catch all statement


}
```

JavaScript switch statement

```
switch (expression)  {
case label1:
        // statements
        break;
case label2:
        // statements
        break;
default:
        //statements;
}
```

JavaScript Loops

- JavaScript for loop:

```
for (var i=0; i < 3; i++) {
    // statements;
}
```

- JavaScript while loop:

```
var i = 0;
while (i < 3) {
    // statements;
    i++;
}
```

JavaScript do while loop

- Do while loop:

```
var i = 0;
do {
        // statements;
        i++;
} while(i < 3);
```

NOTE: special statements:

```
break;  // get out of inner loop
continue;  // go to testing and repeat the loop
```

JavaScript Arrays

- Create JavaScript Array:

```
var ary1 = [" Salem", " Halima"];
```

- OR

```
var ary1 = new Array();
   ary1[0] = " Salem ";
   ary1[1] = " Halima ";
```

JavaScript Objects

- Create JavaScript Object:

```
var obj = { firstName : "Salem", lastName : "Alwafi"
  };
```

- OR var obj = new Object();

```
obj.firstName = "Salem";
obj.lastName = "Alwafi“;
```

- Use Dot notation to access object prop/methods:

```
      obj.firstName = “Ahmed”;
      obj.setName(“Ahmed”);
```