JavaScript

Topic 3

CMPT 350

Overview of JavaScript

- Was developed by Brendan Eich and NetScape in 1995 and then became a joint of Netscape and Sun Microsystems.
- Was originally named Livescript
- JavaScript can be divided into three parts: the core, client-side and server-side
- The core is the heart of the language, its operators, expressions, statements, and subprograms.
- The client-side is a collection of objects that support the control of a browser and user interactions.
- The server side is a collection of objects that make the language useful on a web server, for example, communication with the database.

What is JavaScript?

- JavaScript is a scripting language that allows you to implement complex features on web pages.
  - Make webpages interactive
  - Dynamically update data
  - Control multimedia
  - Response to events
  - Get information about user’s computer
  - Client-side validation (form validation)

  - It is not related to java.

HTML link to JavaScript file

* Always separate content, presentation and behavior.

```
<head>
…
<script src="myscripts.js"></script>
…
</head>
```

- script tag should be placed in HTML’s head

JavaScript Language

Review syntax and features

Comments

- Single line comments start with //.
- Multi-line comments start with /* and end with */.

JavaScript Primitives

- A primitive is a data that is not an object and has no methods
- JavaScript has five Primitive types: Number, String, Boolean, Undefined, and null.

What is the difference between undefined and null

```
let a = null;
console.log(a);
// null
```

```
let b;
console.log(b);
// undefined
```

Variables

- variables are declared with one of the available three keyboards (case sensitive)
  - var (Function scope variable) x= 15;
  - let (Block scope variable) let course=‘cmpt350’;
  - const (Block scope constant) const PI = 3.141592653589793;
- const variables must be assigned a value when they are declared
- JavaScript is loosely typed or dynamically typed. That means types are not specified, but JS automatically types a variable.
- can find out a variable's type by calling typeof

JavaScript Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

JavaScript Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to
?	ternary operator

JavaScript logical Operators

Operator	Description
&&	logical and
||	logical or
!	logical not

JavaScript Numbers

- JavaScript has only one type of number.
- JavaScript Numbers are always 64-bit Floating Point
- Numbers can be written with or without decimals.

```
var x = 25.76;    // A number with decimals
var y = 300;      // A number without decimals
```

JavaScript number methods

- The toString() method returns a number as a string

```
var x = 123;
x.toString();             // returns 123 from variable x
(123).toString();         // returns 123 from literal 123
(100 + 23).toString();   // returns 123 from expression 100 + 23
```

- Converting Variables to Numbers

```
number()    Returns a number, converted from its argument
number(true);          // returns 1
number(false);         // returns 0
number("10");          // returns 10
```

- parseInt() method parses a string and returns an integer.

```
parseInt("10.33");     // returns 10
```

JavaScript String

- JavaScript strings are used for storing and manipulating text.
- A JavaScript string is zero or more characters written inside quotes (double or single).

```
var name = "John Doe";
 var name = 'John Doe';
```

JavaScript String methods and properties

- The **length** property returns the length of a string:

```
var txt = “HelloWorld!";
var sln = txt.length;  //returns 11
```

- The **indexOf()** method returns the index of (the position of) the first occurrence of a specified text in a string:

```
var str = "HelloWorld!”
var pos = str.indexOf(“e"); //returns 1
var pos = str.indexOf("World"); //return 5
```

- The **lastIndexOf()** method returns the index of the last occurrence of a specified text in a string
- Both indexOf() and lastIndexOf() methods accept a second parameter as the starting position for the search

JavaScript String methods and properties

- The search() method searches a string for a specified value and returns the position of the match:

  ```
  var str = "HelloWorld!”
  var pos = str.search("World"); //return 5
  ```

  Is search method same as indexOf() method?

indexOf() and search(), are not equal

- The search() method cannot take a second start position argument.
- The indexOf() method cannot take powerful search values (regular expressions).

More String methods

- The slice(start, end) method extracts a part of a string and returns the extracted part in a new string
- The replace() method replaces a specified value with another value in a string:
- The concat() method joins two or more strings
- The trim() method removes whitespace from both sides of a string

.

.

.

Iterations

- for-loops:

  for ([initialExpression]; [condition]; [incrementExpression])
  statement

- while-loops:

  while (condition)

  statement

- do-while

  do

  statement

  while (condition)

- for-of

  for (variable of object)

  statement

- for-in

  for (properties of object)

  Statement

Conditions

if-else statements

```
        if (...) {
                ... }
        else {
                ... }
```

Conditions

Switch Statement

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

JavaScript Functions

- A JavaScript function is a block of code designed to perform a particular task.

```
Function name(parameter1, parameter2) {
        statement;
        statement;
         …
}
```

Function invocations

- When an event occurs for example when a user clicks a button
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

JavaScript objects

- An object is a collection of properties, and a property is an association between a name (or key) and a value.
- You access the properties of an object with a simple dot-notation:

```
          objectName.propertyName
```

```
var person = {
  firstName: ‘Bob’,
  lastName:  ‘Smith’,
  age: 32,
  gender: 'male’
}
person.firstName
```

JavaScript Object methods

- An object in JavaScript may also have a function as a member, in which case it will be known as a method of that object.

```
let person = {
  firstName: 'Bob',
  lastName:  'Smith',
  displayName: function(){
              console.log(`the person  name is ${person.firstName} ${person.lastName} `) ;
              }
}

person.displayName();                // the person  name is Bob Smith
```

JavaScript Arrays

Store multiple values in one variable.

```
var array_name = [item1, item2, ...];
 var fruits = ['apple', 'banana', 'orange’];
var cars = new Array("Saab", "Volvo", "BMW");
```

Array elements are accessed using their index number

```
fruits[0];
```

The length property of an array returns the length of an array

```
fruits.length;
```

Array Methods

- The toString() converts an array to a string of (comma separated) array values
- The join() is similar to toString(), but in addition you can specify the separator
- The pop() method removes the last element from an array
- The push() method adds a new element to the end of an array
- The sort() method sorts an array alphabetically
- The reverse() method reverses the elements in an array