# Javascript

## Introduction

### JavaScript (or JS) Versions

- JavaScript 5 (or ECMAScript 5.1)
- Available in all the browsers
- ECMAScript 2015 (or ECMAScript 6 or ES6)
- Not all the specs are implemented (browser checking required)

### Features

- **Imperative** and structured programming (if statement, loops, etc.)
- **Object-oriented** programming language based on **prototype** for inheritance (no class).
    - Function as constructor
    - Function as method
- **Functional** programming paradigm (first-class: functions considered as object)
    - High-order functions
    - Anonymous
    - Variables, Arguments, etc.

In a HTML page, the code is written within a <script> element. It may be located:

- within the <head> element

```
<script type="text/javascript">
console.log("Hello World");
</script>
```

- By calling an external file in html:

```
<script type="text/javascript" src="myScript.js"></script>
```

- Everywhere in the <body> element. **Best place**: At the end of <body> when the page is fully loaded.
  **Deprecated!!** Directly in HTML element `<p onclick="alert('Hello');">Click on me </p>`

## JS: The Basics

### Comments

To document the program, you may add comments that won't be executed by the language engine.

- In JS, we have two different comments for single-line and multi-lines.

```
var pi = 3.14; // Single-line comment

// This code is commented → var i = 10;

/*
Multi-line
comment
*/
```

### Semicolon

Each expression must be ended by a semicolon.

```
var i = 1 ;
i = i + 3 ;
i = 2 * i ;
i = i / 10
```

### Variables

```
/*
 2017-09-12
 Jean-Christophe Taveau

 Basics of JS Programming
*/
```

```
// Display Hello World
IJ.log('Hello World');
IJ.log(3.14); // PI number

/**** TD #1: Variables ****/
var i; // step : declaration
i = 12; // step: init
var this_is_a_variable = 1;
var var0 = 23;
// used as iterators in loops:
// i j k l m n p
var width = 100;
var height = 200;
var widthOfImage = 200;
var width_of_image = 300;

// JavaScript 5.0
// ECMAScript 2015 or 6
// let foo = 123;
// const PI = 3.14;

var fooz = 234;
var baz;
fooz = 567;
baz = fooz + 1;

// Op: + * / - %
var myCos = Math.cos(0.0);
IJ.log(myCos);
IJ.log(baz);

fooz = 10;
baz = 20;
var tmp;
tmp = fooz;
fooz = baz;
baz = tmp;
```

### Number

```
/*
2017-09-19
Name

Basics of JS programming
*/

var i=10;
var j=10.12;
var k=1e16;
```

```
IJ.log(i);
```

### String

```
var txt="this is a string";
var txt1='this is also a string';
/*var txt2='doesnot work";
var txt2 = "c'estle printemps";
var txt2 = 'c\'estle printemps';*/
var concatenate = 'today'+'is tuesday'; // today is tuesday
var concatenate1= 'value'+txt;//value: this is a string
```

### Array

```
var arr=[];//empty array
arr[0] = 4;
arr[1]=6;
arr[2]=8;

var arr2=[4,6,8];

var len=arr.length; // 3
var len2 = arr2.length; // length is an attribute of arr2


// functions
arr.push(10);
arr.push(12);

// Read the last element of the array
var last = arr[arr.length-1];
var first = arr[0];

var arr3 = [10,'text',true,[1,2,3]];
// ACGT
//A 1000
//C 0100
//G 0010
//T 0001
```

### Array of Arrays

```
var identity = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]];
var row = identity[1]; // [0,1,0,0]
var value = row[1]; // [1]

var value2 = identity[1][1]; // 1

var row3 = identity[2]; // [0,0,2,0]
var value3 = row3[3]; // [0]
var myvalue3 = identity[2][3]; // 0

var new_identity = [1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1];
var row2_column0 = new_identity[0 + 4 * 2]; // x + width * y = 0+4*2 = 8
```

**Object in JS**

```
var obj = {}, // empty object;
obj.width = 2;
obj.height = 2;
obj.pixels = [12,13,14,15];

IJ.log(obj.width)

var obj2 = {
  width:3,
  height:3,
  pixels[0,1,2,3,4,5,6,7,8]}
```

- boolean :

  > True False

- number:

  > byte int (32 bits) long (int 64 or 128 bits) float (floating point numbers)

- string

- array (python = list)

- object

## Loop

**while**

- Step 1 : initialization (start)
- Step 2 : exit condition
- Step 3 : increment step

```
var i = 1; // initialization
while(i<=10){ // (exit condition){defining a bloc of code}
```

```
        IJ.log(i);
        i = i+1; // Incrementation
}
```

**for** (most used loop "for")

```
for(var i = 1;i<=10;i = i + 1){ // all step in here
        IJ.log(i);
}
```

**for...in** : exist but not used a lot.

Assignement op :

```
i = i + 1;
i += 1; // -= ; /= ; *=
i ++
```