JAVASCRIPT INTRODUCTION

Presentation by Juha M Soderqvist

JAVASCRIPT ENGINE

A JavaScript engine is a program which executes **JavaScript** code.

A JavaScript engine is most commonly used in web browsers

The first JavaScript: SpiderMonkey engine was created by Brendan Eich at Netscape

SpiderMonkey, the first JavaScript engine is used in Firefox and Mozilla.

V8 - open source, developed by Google in Denmark, part of Google Chrome

JavaScriptCore - open source, marketed as Nitro and developed by Apple for Safari

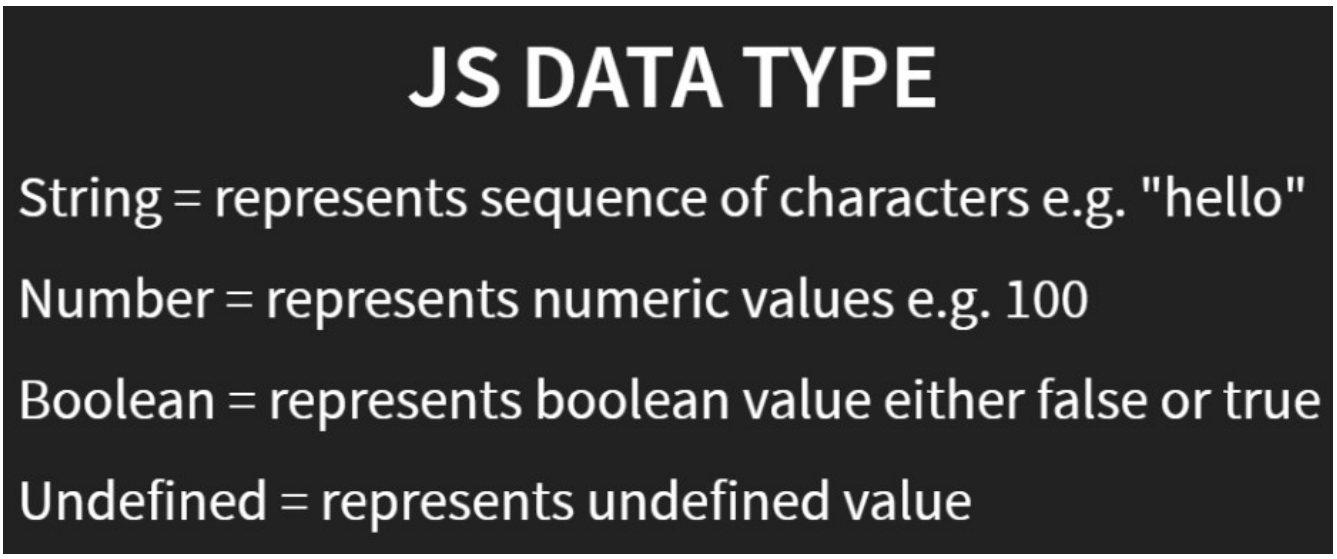
Chakra (JavaScript) for Microsoft Edge

JAVASCRIPT

Paradigm:	Multi-paradigm: scripting, object-oriented (prototype-based), imperative, functional, event-driven
Designed by:	Brendan Eich - Netscape Communications Corporation, Mozilla Foundation, Ecma International
First appeared:	First appeared May 23, 1995
Major implementations:	V8, JavaScriptCore, SpiderMonkey, Chakra

SYMBOLS

()	**Round brackets**	:	**colon**
{}	**Curly brackets**	;	**semicolon**
<>	**Angle brackets**	&	**ampersand**
,	**comma**	#	**hash**
' "	**single and double quotation**	.	**dot, period**



JS VARIABLES

DECLARE A VARIABLE

```
var x;
```

var = variable keyword

x = variable name

; = end with semicolon

note: variables are Case sensitive

JS VARIABLES

ASSIGN A VALUE TO A VARIABLE

```
x=1;
```

x = variable name

= is an assigment operator

1 = variable value

; = end with semicolon

JS VARIABLES

DECLARE A VARIABLE + ASSIGN VALUE

```
var x=1;
```

var = variable keyword

x = variable name

= is an assigment operator

1 = variable value

; = end with semicolon

JS EXPRESSIONS

TWO TYPES OF EXPRESSIONS

1. ASSIGN A VALUE TO A VARIABLE

```
var numberx = 1;
```

2. MULTIPLE VALUES TO RETURN A SINGLE VALUE

```
var area = 3 * 2;
```

JS OPERATORS

assigment	color = 'black';
arithmetic	area = 3 * 2;
string	greeting = "Hi " + "John";
comparison	newhighscore = score > highscore;
logical	sell = (cprice > oprice) && (nasdaqindex > tokyoindex);

CODE

…
CODE #1 - VARIABLE NUMBER
Writing into the browser console, using console.log().
var variable = 5;
console.log(variable);
Result: 5

CODE #2 - VARIABLE STRING
var variable = 'text5';
console.log(variable);
Result: text5

CODE #3 - VARIABLE BOOLEAN
var variable = true;
console.log(variable);
Result: true

CODE #4 - TYPEOF OPERATOR
JavaScript typeof operator to find the type of a JavaScript variable.

The typeof operator returns the type of a variable or an expression

var var1 = 6.5;
console.log(typeof var1);
Result: number

var var1 = "hello";
console.log(typeof var1);
Result: string

CODE #5 - TYPEOF OPERATOR NAN
"Not-a-Number" value. This property indicates that a value is not a legal number

var var1 = NaN;
console.log(typeof var1);
Result: NaN

var var1;
var1 = 99 + aa;
console.log(var1);

Result: NaN

CODE #6 - TYPEOF EMPTY AND NULL
empty string variable is assigned with ""

In JavaScript null is "nothing". null is an object

```
// empty string
var var1 = "";
console.log(typeof var1);
```
Result: string

```
// You can empty an object by setting it to null
var var1 = null;
console.log(typeof var1);
```
Result: object

CODE #7 - TYPEOF UNDEFINED
a variable without a value, has the value undefined. ""

The typeof is also undefined.

```
// Value is undefined, type is undefined
var person;
console.log(typeof person);
```
Result: undefined

null and undefined: Different datatypes

```
console.log((null === undefined));
```
Result: false

null and undefined: Same values

```
console.log((null == undefined));
```
Result: true

CODE #8 - ARRAYS
An array is a special variable, which can hold more than one value at a time.

An array is an object.

```
var var1 = [1, 2, 3]
console.log(var1[0]);
console.log(typeof var1);
```
Result: 1

object

CODE #9 - MIXED DATA TYPES
Mixed data types in array

```
var var1 = [1, 2, 3, 'juha']
console.log(var1[3]);
console.log(typeof var1);
```
Result: juha

object

CODE #10 - DYNAMIC DATA TYPES
JavaScript has dynamic types.

This means that the same variable can be used to hold different data types

```
var x;
console.log(typeof x);
var x = 5;
console.log(typeof x);
var x = "John";
console.log(typeof x);
```
Result: undefined

Number

String

CODE #11 - EXPRESSION EVALUATION
JavaScript evaluates expressions from left to right. Different sequences produce different results

```
var x = "juha" + 16 + 4;
console.log(x);
```
Result: juha164

```
var x = 16 + 4 + "juha";
console.log(x);
```
Result: 20juha

CODE #11.1 - OBJECTS
Objects are variables too. But objects can contain many values.

Assigns many values (Fiat, 500, white) to a variable named car

```
var car = {type:"Fiat", model:"500", color:"white"};
console.log(car.type);
```
Result: Fiat

```
var person = {firstName:"John", lastName:"Doe", age:50};
console.log(person.age);
```
Result: 50

CODE #12 - OBJECT EXAMPLE
Object.Property.Value

code is inside html document "body" and id="_demo" remove _

```
<p>Creating a JavaScript Object.</p>
<p id="_demo"></p>

<script>
var person = {
   firstName : "John",
   lastName  : "Doe",
   age       : 50,
   eyeColor  : "blue"
};

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>
```
Result: Creating a JavaScript Object.

John is 50 years old.

CODE #13 - OBJECT SEMICOLON
Note: semicolon after the last curly brace

```
// show object
var var1 = {
name: 'juha'
}; // dont forget semicolon
console.log(var1);
```
Result: will show the object

```
// access the property inside the object
var var1 = {
name: 'juha'
};
console.log(var1.name);
```
Result: juha

CODE #14 FUNCTIONS
function is a block of code designed to perform a particular task.

function inside a object or class is called a method

// = comments in javascript
function = function keyword
hello = function name
parenthesis = ()
{} = curly brace - code block for the function
// hello function date 20170115
function hello() {
console.log('hello my friend');
}

CODE #15 CALLING A FUNCTION
calling a function

// call the function
hello();
hoisting

// call the function
hello();

//define function after function call = hoisting
function hello() {
console.log('hello my friend');
}

CODE #16 ASSIGN FUNCTION TO VARIABLE
var cool = function() {
console.log('hello my friend');
};

// call variable which is a function
cool();
result: hello my friend

CODE #17 FUNCTION ARGUMENT AND RETURN
function talkbox() {
return 'bla bla';
}

var string1 = talkbox();
console.log(string1);
result: bla bla

CODE #18 FUNCTION ARGUMENT AND RETURN

```
function hello() {
var hellomsg = "hello my friend"
return hellomsg;
}

var returned = hello();
console.log(returned);
result: bla bla
```

ARITHMETIC OPERATORS

+ Addition - Subtraction
*** Multiplication / Division**
% Modulus ++ Increment

```
----------------------------------------------------

CODE #19 CALCULATION
function calc(number1, number2) {
return number1 + number2;
}

var returned = calc(1, 2);
console.log(returned);
result: 3
----------------------------------------------------
CODE #20 CALCULATION
function calc(number1, number2) {
return number1 * number2;
}

var returned = calc(2, 2);
console.log(returned);
result: 4
----------------------------------------------------
CODE #21 LOGICAL OPERATORS
&&      AND
||      OR
!       NOT
^(bitwise)      XOR
```

```
var x1 = 6;
var x2 = 3;
var boolx = (x1 < 10 && x2 > 1);
console.log(boolx);
result: true

var condition = true;
if (condition) {
console.log("1");
}
result: 1
```

CODE #22 COMPARISON OPERATORS

==	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

```
var x1 = 6;
var x2 = 6;
if (x1 == x2){
console.log("x1 and x2 same value"}
} else {
console.log("x1 and x2 not same value");
}
```

CODE #23 CONDITIONS

```
//example1
var status = true;
if (status) {
console.log("its true";
}
// same in BASIC
// if status = true then print "its true"



//example2
var condition = true;
```

```
if (condition) {
console.log("1");
} else {
console.log("0");
}
```

CODE #24 CONDITIONS ELSE IF

```
var condition = true;
var another = true;
if (condition) {
 console.log("condition is true");
} else if (another) {
 console.log("another is true");
} else {
 console.log("something else");
}
```

CODE #25 CONDITIONS ELSE IF

```
        <script>
function myFunction() {
    var time = new Date().getHours();
    var greeting;
    if (time < 10) {
        greeting = "Good morning";
    } else if (time < 20) {
        greeting = "Good day";
    } else {
        greeting ="Good evening";
    }
 console.log(time);
 console.log(greeting);
}
</script>
```

CODE #25 CONSOLE OUTPUT

```
> var time = new Date().getHours();
    var greeting;
    if (time < 10) {
        greeting = "Good morning";
    } else if (time < 20) {
        greeting = "Good day";
    } else {
        greeting ="Good evening";
    }
  console.log(time);
  console.log(greeting);
4
Good morning
```

CODE #26 SWITCH

```
var lucknr = 8;
switch (luckynr) {
  case 1:
  console.log("is 1");
  break;
  case 8:
  console.log("is 8");
  break;
  default:
  console.log("default");
  break;
}
// break = jumps out of the switch
```

CODE #27 SWITCH (LUCKY NR) IN ELSE IF

```
//switch versus IF


if (luckynr == 1) {
 console.log("is 1");
} else if (luckynr == 8) {
console.log("is 8");
} else {
console.log("default");
```

```
}
```

CODE #28 LOOPS

```
//example 1 loop until variable i is less than 5
for (var i = 0; i < 5; i++) {
console.log(i);
}

//EXAMPLE 2 loop until variable i is less than 5
for (var i = 0; i < 5; i = i + 3) {
console.log(i);
}

//INFINITE LOOP
for (var i = 0; i < 5; i--) {
console.log(i);
}
```

CODE #29 FOR LOOPS IN ARRAYS

```
//example hardcoded array list - we know the size of the array
var array = [1, 2, 3];
for (var i = 0; i < 3; i++) {
console.log(array[i});
}

//example length of array list - aka output array
var array = [1, 2, 3];
for (var i = 0; i < array.length; i++) {
console.log(array[i});
}
```

CODE #30 WHILE

```
//example while 1
var number = 5;
while (number < 7) {
console.log(number);
number++;
}


//example while 2
var condition = true;
var i = 2;
```

```
while (condition) {
 if (i == 3) {
  condition = false;
}
console.log(i);
i++;
}
```

CODE #30.1 DO WHILE

```
//example do while
var condition = false;

do {
 console.log("blabla");
}
while (condition) {
console.log("now its true");
}
```

CODE #31 HELLO WORLD

```
<p id="demo2"></p>
 <button onclick="grazy()">Click me</button>

function grazy() {
document.getElementById("demo2").innerHTML = "Hello World";
}
```