JAVASCRIPT IN HTML

FOUNDATION OF WEB DEVELOPEMENT

Javascript

We'll review the following topics

- Data types
- Arithmetic operators
- Condition & Loops
- Functions
- Arrays
- Objects & JSON

Introduction

- Javascript is an interpreted language, browser interpet it on the fly. Unlike compiled language.
- Javascript is written in a text file with the extension „.js“ or as part of an html document.
- The text contains the commands that make up the program.
- Javascript is a standart that each vendor can follow entitled with: ECMA-262
- Javascript is a very popular language and it allows us to create dynamic & interactive websites.

Hello World!

Type the following code in a new document titled HelloWorld.html

mind case sensitive.

```
<script type="text/javascript">

    // our main function Hello World
    function helloWorld() {
        alert('Hello World!');
    }

    helloWorld();

</script>
```

Hello World (2)

In an external link:

```
<html>
   <head>
        <script src="myscript.js">
        </script>
   </head>
   <body>
     <input type="button"
onclick="msg()" value="Message">
   </body>
</html>
```

JS file:

```
function  msg()
{
        alert("Hello World!");
}
```

Displaying text on web page

- window.alert(): writing into an alert box
- document.write(): writing into HTML output
- innerHTML: writing into HTML element
- console.log(): writing into browser console.
- document.getElementbyId("demo"): indexing element by Id

```
<!DOCTYPE html>
<html>
<body>
<h1> My first Web Page </h1>
<p> My first paragraph. </p>
<script >
    // our logic
    window.alert(5+6);
</script>
```

```
<!DOCTYPE html>
<html>
<body>
<h1> My first Web Page </h1>
<p> My first paragraph. </p>
<p id="demo"></p>
<script >
    // our logic
    Document.getElementById("demo").
    innerHTML=5+6;
    // window.alert(5+6);
</script>
```

Variables & data types

Variable who?

- Variable is a piece of data that contains information while our code is executing.
- Variables can be used to store data and retrieve data later on.
- Each variable has a data type (number, string, date).
- Javascript is an implicit language and don‟t need to specify the variable data type.

Variables & data types

Declaring variables

To declare an variable specify var before the name e.g:

var {NAME};

Declaring a var named “num”:

```
var num;
```

Declaring multiple variables in one var statement

```
var name, address;
```

Initializing variables with initial value:

```
var x = 100, message = 'Hello';
```

Variables & data types

Variable scope

Variable scope confines the variable to the block(curly brackets) in which he was created:

Global scope:

```
<script type="text/javascript">
    var x = 10;
</script>
```

Variable y is known only inside function “doSomething”:

```
<script type="text/javascript">
    function doSomething() {
        var y = 99;
    }
    alert(y); //Uncaught ReferenceError: y is not defined
</script>
```

Variables & data types

assignment

Using Assignment we can store data in a given variable.

The sign = means assignment. e.g:

Place the value 3 inside the variable num:

```
num = 3;
```

One row declaration + assignment:

```
var num = 1;
```

Switch var

Variables & data types

Data types

Javascript base data types:

- string
- number
- boolean
- function
- Array
- object

Variables & data types

```
// string
var companyName = ‘Google';

// number
var pi = 3.14;
var year = 2013;

// boolean
var flag = true;
var FALSE = false;

// function
var sayHello = function () {
    alert('hello world!');
}

// array
var numberArray = [1, 2, 3];
var animals = new Array("cat", "dog", "mouse", "lion");

// object / json
var person = {
    name: 'Barack Hussein Obama II',
    age: '51',
    title: '44th President of the United States'
```

Operators

Arithmetic operation:

```
<script>
    var x = 10, y = 20;

    var addition = x + y;
    var subtraction = y - x;
    var multiplication = x * y;
    var division = x / y;
    var remainder = x % y;
</script>
```

if-else statement

if-else allows us to control the flow of our program.

```
if (condition){
    //code
}

if (condition){
    //code
} else {
    //code
```

if-else statement

if-else

Example:

```
var currentTime = 8;

if (currentTime > 6 && currentTime <= 7) {
    alert('wake up!');
} else if (currentTime > 12 && currentTime <= 13) {
    alert('launch time!');
} else {
    alert('spare time at last...');
}
```

Boolean Conditions

Types of Boolean comparison:

Operator	Name	Description
x < y	Less than	true if x is less than y, otherwise false.
x > y	Greater than	true if x is greater than y, otherwise false.
x <= y	Less than or equal to	true if x is less than or equal to y, otherwise false.
x >= y	Greater than or equal to	true if x is greater than or equal to y, otherwise false.
x == y	Equal	true if x equals y, otherwise false.
x != y	Not Equal	true if x is not equal to y, otherwise false.

Boolean Conditions

```
<script>
    var num1 = 10;
    var num2 = 20;

    if (num1 > num2) {
        alert('num1 is bigger');
    }

    var num2bigger = num1 > num2;
    if (num2bigger) {
        alert('num2 is bigger');
    }

    if (num1 == num2) {
        alert('num1 equal num2');
    }

    if (num1 != num2) {
        alert('num1 not equal num2');
    }
</script>
```

Boolean Conditions

More operators, and / or / not

Operator	Name	Description
x && y	And	True if both x and y are true, otherwise false.
x || y	Or	True if at least one of x or y are true, otherwise false.
! x	Not	True if x is false, otherwise false.

Boolean Conditions

Conditional operators and / or / not

Operator	Name	Description
x && y	And	True if both x and y are true, otherwise false.
x || y	Or	True if at least one of x or y are true, otherwise false.
! x	Not	True if x is false, otherwise false.

Boolean Conditions

```
<script>
    var rabbitName = 'archimedes',
        age = 1;


    if (rabbitName == 'archimedes' && age == 1) {
        alert('hello Archie, welcome home!');
    }

    if (age == 0 || age == 1) {
        alert('hello junior rabbit!');
    }

    var isYoung = age < 5;

    if (!isYoung) {
        alert(rabbit is old!');
    }

</script>
```

Math

Math class encapsulate a lot of usefull methods:

- Math.abs(x) absolute value of a Decimal number.
- Math.max(x1,x2) **&** Math.min(x1, x2) Return the number with the highest/lowest value
- Math.pow(x, y) - x^y
- Math.sqrt(x) square root of a number
- Math.random() random number between 0 and 1
- Math.PI π - 3.14159

for loop

Loops can execute a block of code a number of times.

Syntax

```
for(<initial>;  <condition> ; <update>) {
    // code goes here
}
```

Example

```
for (var i = 0; i < 10 ; i++) {
    document.write('this is row ' + i + '<br />');
}
```

for loop

Code

```
for (var i = 0; i < 10 ; i++) {
    document.write('this is row ' + i + '<br />');
}
```

Output

this is row 0
this is row 1
this is row 2
this is row 3
this is row 4
this is row 5
this is row 6
this is row 7
this is row 8
this is row 9

While loop

The while loop loops through a block of code as long as a specified condition is true.

Syntax

```
while (condition) {
    // code to repeat
}
```

While loop

Code

```
var count = 0;
while (count < 10) {
    document.write('Count: ' + count + '<br />');
    count++;
}
```

Output

```
Count: 0
Count: 1
Count: 2
Count: 3
Count: 4
Count: 5
Count: 6
Count: 7
Count: 8
Count: 9
```

Functions

A function is a block of code that will be executed when it is called, Both of the following functions declarations are exactly the same, functions are indeed variables:

```
function clickMe() {
    alert('clicked!');
}

var clickMe = function () {
    alert('clicked!');
```

Functions

Function can have parameters & return value with the `return` keyword.

```
function sum(x, y) {
    return x * y;
}

var six = sum(2, 3);
alert(sum(5, 10));
sum(5, sum(5, 5));
```

Arrays

- The Array object is used to store multiple values in a single variable.
- Array can add & remove values
- Array can store diffrent data types
- Array are Zero-based
- Examples:

Arrays

Declaring Arrays & Initialization

```
var myArray1 = new Array(10, 22);
var myArray2 = new Array();
var myArray3 = [];
var myArray4 = [1, 2, 3];
var myArray5 = new Array("cat", "dog", "mouse", "lion");
var myArray6 = new Array(10); // predefined size array
var myArray7 = [1, "hello world!", 1.24, function () { }, [1, 2, 3],
                null, undefined, { a: 1, b: 2 }, document.body];
```

Arrays

Arrays can be accessed via index:

```
var animals = new Array("Cat", "Dog", "Mouse", "Lion");
```

Get the first value of the array:

```
var cat = animals[0];
```

Assign value to the third index of the array:

```
animals[2] = 'Giraffe';
```

Arrays

Get the current items in the array with the `length` property:

```
var animals = new Array("Cat", "Dog", "Mouse", "Lion");
var animalsCount = animals.length;
// animalsCount = 4
```

Push a new item to the array:

```
Animals.push('Kangaroo‘);
```

Checking the length again:

```
animalsCount = animals.length;
// animalsCount = 5
```

Iterate over the values of the array and use alert to show them;

```
for (var i = 0; i < animals.length; i++) {
    alert(animals[i]);
}
```

Objects

Objects are a special kind of data in javascript.

Objects can be used with properties to assign data:

Example of an object:

```
var person = {
    name: 'Barack Hussein Obama II',
    age: '51',
    title: '44th President of the United States'
}
```

Objects

Access to Object properties:

```
var person = {
    name: 'Barack Hussein Obama II',
    age: '51',
    title: '44th President of the United States'
}

alert(person.name); // Barack Hussein Obama II
alert(person['name']); // Barack Hussein Obama II

person.age = 51;
person['age'] = 51;
```

Objects

Often in web development we Get JSON data and need to manipulate it:

```
{
    'employees': [
            { "firstName": "John", "lastName": "Doe" },
            { "firstName": "Anna", "lastName": "Smith" },
            { "firstName": "Peter", "lastName": "Jones" }
```

Then we can dynamiccaly create html from our data object.