University of Oum El Bouaghi

Advanced Web Programming

JavaScript Reminders

Concerned students

Faculty	Department	Level	Speciality
ESNL	MI	B3	ISSE

Interest

- **Example 1**

```
<html>
<head>
   <title>Static Page</title>
</head>
<body>
<div>
   We are on the 11/01/2024
</div>
</body>
</html>
```

Interest

- **Example 1**

```
<html>
<head>
   <title>Static Page</title>
</head>
<body>
<script type = "text/javascript">
 date = new Date();
 document.writeln(" We are on the", date);
</script>
</body>
</html>
```

Some advice

- **JavaScript code can be challenging to debug.**
  - Use a **debugger**,
  - Pay attention to formatting (**code indention**),
  - Never use global variables (use the '**var**' keyword).

Insertion of JS code

- **In the header of the page**
  - Between the **<head>** and **</head>** tags.
  - Code executed **upon a user event**.
  - The event is located **in the body** of the document.

```
<html>
<head>
<script type="text/javascript">
   functionf () { alert('Goodbye'); }
</script>
</head>
<body onUnload="f();"> // closing the current page
</body>
</html>
```

Insertion of JS code

- **In the body of the HTML page**
  - Between the **<body>** and **</ body>** tags.
  - Executed when **loading the page**.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
   alert('good morning');
</script>
</body>
</html>
```

Insertion of JS code

- **In the header or in the file**
  - File in text format.
  - **Adventage:** Reusability of the script across multiple pages.
  - **Disadvantage:** Additional request to the server

```
<html>
<head>
<script type="text/javascript"  src="fichier.js"></script>
</head>
<body>
<input type="button" onclick="popup()" value="Clic">
</body>
</html>
```

JavaScript console

- **JavaScript console** is present in the development tools of Firefox or Chrome(keyboard shortcut **F12**).

Examlpe:

- x; //Undefined variable
- **var** x; x; // Defined variable, uninitialized
- **var** x = 1; x; // Defined and initialized Variable

- // Display a variable in the **console**

```
console.log(x);
```

- // Display an alert dialog

```
alert("Coucou !");
```

Typing and Variables

- Variables are objects.
- They can be declared at anytime.
- '**var**' keyword can be used for their declaration.
- Every value has a type in JavaScript.
- This type is **not specified during the declaration** of a variable.

```
var x = 1;
typeof x;  // → number
var y = "How was your day?";
typeof y;  // → string
```

- **Types are dynamic**

```
var x;                  // x is undefined
var x = 5;                    // x is a Number
var x = "Ahmed";        // Now x is a String
```

Typing and Variables

- In **JavaScript,** There are six data types including:
  - Five primitives: **Boolean, Number, String, Null, Undefined.**
  - Objects: **Object.**

- **Undefined** is the type of variable that have no value (e.g. uninitilized variable).

```
var x;
typeof x;
// → undefined
```

Operators

- **affectation**

  +=, -=, *=, /=, %=, &=, |=, <<=, >>=

- **comparison**

  ==, !=, <, <=, >, >= ,=== ,!==

- **arithmetic**

  %, ++, --

- **logic**

  &&, ||, !

- **bit**

  &, |, ^ (XOR), <<, >>, >>>

Control structures

- Javascript has control structure similar to those in C language:
  - **if else**
  - **for, while, do.. while**
  - **break, continue**
  - **switch**
- **for … in** (foreach) loop for arrays.
- and the **throw, try, catch** exceptions.

Functions

- Declaration like in Java, C++

```
function square(x) { return x * x; };
```

- Variables **can store** functions!

```
var square = function (x) {  return x * x;};
```

- Functions are '**first-class**' **objects**: they can be manipulated and exchanged like any other JavaScript objects.

```
function square(x) {  return x * x; };
var varfonc = square;  // Variable assignment
//Function execution with the operator()
varfonc(2); // → 4
```

Main Functions

- **eval(string)** : Evaluates the Javascript code.
- **Number(var)** : Convert to a number.
- **String(var)** : Convert to a string.
- **int parseInt(string[,radix])** : Convert to an integer based on the specified radix(base).
- **float parseFloat(string)** : Convert to a real.
- **encodeURI(uri)**
- **decodeURI(uri)**

JavaScript timers

- **timers** allow executing an action after a certain delay, or performing an action every x seconds.
- Activating a timer is a simple operation:
- **setTimeout() and clearTimeout()**

```
window.setTimeout("myfunction()",1000);
```

- This code will call the function myfunction() after one second (time is expressed in milliseconds).
- One can stop a timer before its timeout by using **clearTimeout**() function. It is necessary to "name" timeout, like this:

```
mytimer=window.setTimeout("myfunction()",5000);
```

- This will stop it like this:

```
window.clearTimeout(mytimer);
```