

ECRIRE UN SCRIPT

Les scripts se placent dans `<head>`
mais on préférera les mettre juste avant `</body>`.

```
<script>  
  "use strict"; // ça c'est cadeau  
  alert("Hello, world!");  
</script>  
<script src="path/to/script.js"></script>
```

L'attribut `language` de la balise `script` est déprécié depuis HTML5.

LA CONSOLE

Pour le debug.

```
var message = "Hello, world!";  
  
alert(message);  
console.log(message);  
console.info(message);  
console.warn(message);  
console.error(message);
```

QUELQUES RAPPELS ?

Pour voir (ou revoir) les bases de Javascript et de la programmation.

QUELQUES RAPPELS ?

- COMMENTAIRES
 - variables & types
 - instructions
 - structures de contrôle
 - fonctions
-

```
// Commentaire  
alert(); // Commentaire  
  
/* Commentaire  
sur  
plusieurs  
lignes  
alert(); */
```

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

Sert à stocker une donnée en mémoire vive.

```
"Hello, world!";  
var message = "Hello, world!"; // affectation  
console.log(message);
```

Se déclare avec `var`, en **CAMELCASE**.

```
var customerName = "Marie";  
var size = 167,  
    lastAction,  
    isHappy = true;
```

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

Leur portée (scope) est la fonction parente, et la `window` par défaut.

```
var customerName = "Marie";  
customerName; // "Marie"  
window.customerName; // "Marie"
```

```
function doSomething() {  
    var customerName = "Marie";  
    customerName; // "Marie"  
    window.customerName; // undefined  
}  
customerName; // undefined  
window.customerName; // undefined
```

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

On peut mettre de tout dans une variable. Types de données primitifs :

```
"Hello, world!"; // string  
2.5; // number  
true; // boolean  
null; // null  
undefined; // undefined
```

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

Tout ce qui n'est pas primitif est object.
Tout, c'est par exemple :

- screen
- navigator
- window
- document
- console
- history
- ...

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

Même les tableaux sont des object.

```
// Tableau simple
var letters = ["a", "b"];
letters.push("c");
letters[2];
```

```
// Tableau associatif
var letters = {
    a: "A", // Les clés sont nécessairement des strings
    "b": "B",
};
letters.c = "C";
letters.c;
```

QUELQUES RAPPELS ?

- commentaires
 - **VARIABLES & TYPES**
 - instructions
 - structures de contrôle
 - fonctions
-

On peut combiner tableaux et tableaux associatifs :

```
var authors = [  
  {  
    "name": "Jules Verne",  
    "books": ["Voyage au Centre de la Terre"]  
  },  
  {  
    "name": "Honoré de Balzac"  
    "books": ["La Peau de Chagrin", "Le Père Goriot", "Gobseck"]  
  }  
];
```

... à l'infini !

QUELQUES RAPPELS ?

- commentaires
- **VARIABLES & TYPES**
- instructions
- structures de contrôle
- fonctions

JS est faiblement typé :

```
var a;  
a = "Hello, world!";  
a = 2.5;  
a = true;  
a = null;  
a = undefined;
```

QUELQUES RAPPELS ?

- commentaires
- variables & types
- **INSTRUCTIONS**
- structures de contrôle
- fonctions

Une instruction dicte une action.

```
var message;  
message = 2 * 5;  
alert(message);  
alert(3 * 2);  
message = "Hello, world!";  
console.log(message.length);  
document.getElementById("demo").innerHTML = message;
```

Elles sont séparées par un point-virgule et un retour à la ligne.

QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - **STRUCTURES DE CONTRÔLE**
 - fonctions
-

Servent à contrôler les instructions
pour écrire des **ALGORITHMES**.

```
var age = 10; // instruction

// Structure de contrôle : condition
if (age >= 18) {
    console.log("adulte"); // instruction
} elseif (age >= 0) {
    console.log("enfant"); // instruction
} else {
    console.log("c'est possible ça ?"); // instruction
}
```

Le contenu d'un `if` est toujours interprété comme un boolean.

QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - **STRUCTURES DE CONTRÔLE**
 - fonctions
-

```
var age = 0; // instruction

// Structure de contrôle : boucle (de répétition)
while (age < 18) {
    console.log("âge : " + age); // instruction
    age++; // instruction
}
console.log("âge : " + age); // instruction
```

QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - **STRUCTURES DE CONTRÔLE**
 - fonctions
-

Elles suffisent à presque tout faire :

- conditions : `if`, `elseif`, `else` et `switch`
- boucles : `for`, `while`

Les boucles (`for`) sont le plus souvent utilisées pour parcourir des tableaux.

QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - structures de contrôle
 - **FONCTIONS**
-

Une fonction est une procédure avec **DES ENTRÉES** (paramètres) et **UNE SORTIE** (valeur de retour).
On trouve un traitement entre les deux.

```
function add(a, b) {  
    var result = a + b;  
    return result;  
}  
  
console.log(add(1, 8)); // affiche 9
```


QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - structures de contrôle
 - **FONCTIONS**
-

Deux façons de déclarer une fonction :

```
// Fonction classique
function add(a, b){
    // ...
}
add(1, 1);
```

```
// Fonction dite "anonyme" stockée dans une variable
var add = function(a, b){
    // ...
};
add(1, 1);
```

QUELQUES RAPPELS ?

- commentaires
 - variables & types
 - instructions
 - structures de contrôle
 - **FONCTIONS**
-

Javascript embarque déjà quelques fonctions ou bibliothèques prédéfinies :

`Math`, `Date`, `JSON`, `parseInt`, `setTimeout`, `alert`, ...
et `jQuery` encore plus !

OPÉRATEURS

- + (addition et concaténation)
- -
- *
- /
- %
- ++
- --

```
var a = 1;  
a = a + 1; // 2  
a++; // 3
```

```
"Hello," + " world!";
```

AFFECTATIONS

- =
- += (addition et concaténation)
- -=
- *=
- /=

```
var a = 1;  
a = a + 5; // 6  
a += 2; // 8  
a *= 2; // 16
```

```
var a = "Hello,";  
a += " world!";
```

COMPARAISONS

- == et ===
- != et !==
- > et <
- >= et <=
- typeof
- instanceof
- ?

```
var a = 1;
typeof a; // number

a == true; // true
a === true; // false

a == "1"; // true
a === "1"; // false

a == 1; // true
a === 1; // true
```

LIENS UTILES

- http://www.w3schools.com/js/js_conventions.asp
- http://www.w3schools.com/js/js_best_practices.asp

SÉLECTION D'ÉLÉMENTS DOM

En Javascript brut :

```
// Les éléments "label"
document.getElementsByTagName("label");

// Les éléments qui ont l'id "user-form"
document.getElementById("user-form");

// Les éléments qui ont la class "submit"
var submits = document.getElementsByClassName("submit");

// Le deuxième élément qui a la classe "submit"
submits[1];
```

Bonne pratique : stocker les résultats dans des variables s'ils doivent être réutilisés.

ÉVÈNEMENTS

Déclenchés sur des éléments DOM.

ÉVÈNEMENTS

Souris

- `click`
- `dblclick`
- `mousedown`
- `mouseup`
- `mouseenter` ou `mouseover`
- `mouseleave` ou `mouseout`
- `mousemove`
- `scroll`

ÉVÈNEMENTS

Clavier

- `keydown`
- `keyup`
- `keypress` (maintien d'une touche enfoncée)

ÉVÈNEMENTS

Autres

- focus
- blur
- resize
- change
- ...

Doc : http://www.w3schools.com/jsref/dom_obj_event.asp

ÉCOUTEURS D'ÉVÈNEMENTS

```
<a href="#">Click to say hello</a>
```

En Javascript brut, version inline (old school) :

```
<a href="#" onclick="alert('Hello, world!'); return false;">Click to say
```

En Javascript brut, avec la fonction `addEventListener` :

```
var links = document.getElementsByTagName("a"),
    link = links[0];

link.addEventListener("click", function(e){
    e.preventDefault();
    alert("Hello, world!");
}, false);
```

SUITE

Partie 2 : jQuery