



# Java Script

## Table des matières

1.1 - Présentation du document.....	2
1.2 - Présentation de JavaScript.....	3
2 - Les variables.....	4
2.1 - JavaScript Console : la structure séquentielle.....	6
2.2 - JavaScript Console : la structure conditionnelle.....	8
2.3 - JavaScript Console : la structure itérative.....	9
2.4 - JavaScript Console : les tableaux.....	10
2.5 - JavaScript Console : les fonctions.....	11
2.6 - Page web : Authentification.....	12
2.7 - Page web : Inscription.....	14

## **1.1 - PRÉSENTATION DU DOCUMENT**

### **Présentation du document :**

Le document qui suit est une préparation à la programmation.

Il nécessite une connaissance minimale en HTML.

Il peut être lu et pratiqué en parallèle d'un document sur l'algorithmique.

La première partie présente du JavaScript à la console (pour illustrer les principes de base de la programmation).

La deuxième présente du JavaScript client en liaison avec du HTML.

Cette présentation est simplifiée au maximum.

### **Contenu :**

Présentation de JavaScript.

Les variables.

Les structures de base de la programmation (séquence, condition, boucle).

Les tableaux.

Les fonctions.

Les formulaires.

Les listes déroulantes.

## 1.2 - PRÉSENTATION DE JAVASCRIPT

JavaScript est un langage client et un langage serveur.

Côté client il est interprété par un navigateur, côté serveur il est interprété par Node.js.

Cette présentation concerne JavaScript côté client.

Côté client c'est du DHTML (Dynamic HTML) c'est-à-dire qu'il nous permet de récupérer les valeurs des attributs des éléments HTML ou de modifier les textes des documents HTML ou même de modifier le DOM – Document Object Model, le modèle de document HTML - (ajouter, suppression ou modification des éléments HTML).

JavaScript est un langage orienté objet (les éléments HTML – <label>, <button>, <select> - sont des objets informatiques) , procédural et événementiel (pensez au clic sur un bouton, c'est un événement au sens informatique du terme).

Le langage JavaScript est sensible à la casse c'est-à-dire que l'écriture des instructions doit respecter les majuscules et des minuscules.

Il est fortement conseillé de n'utiliser que les lettres non accentués, les chiffres et le tiret de soulignement (\_ tiret du 8) pour nommer une variable. Les variables sont aussi sensibles à la casse.

Le code suivant génère une erreur parce que la variable est nommée **age** et non pas **Age**.

```
> var age = 18;
< undefined
> console.log(Age)
✖ ▶ Uncaught ReferenceError: Age is not defined
  at <anonymous>:1:13
```

Le code suivant génère une erreur parce que l'instruction de boucle est **for** et non pas **For**.

```
> For(var i=1; i<=12; i++) console.log(i)
✖ Uncaught SyntaxError: Unexpected token var
```

## 2 - LES VARIABLES

Un programme est une suite d'instructions qui manipule des variables.

Une variable est un espace mémoire qui permet de stocker une valeur. Une variable de type tableau permet de stocker plusieurs valeurs (cf plus loin la section 2.4).

Une variable est caractérisée par un nom, un type – JavaScript est faiblement typé - et une valeur.

On déclare une variable ainsi :

```
| var a = 3 ;  
| var b = 5 ;
```

On affecte une valeur à une variable ainsi :

```
| a = 5 ;
```

On récupère la valeur d'une variable ainsi :

```
| var c = a ;
```

ou

```
| var r = a + b ;
```

Un objet HTML pour JavaScript est modélisé ainsi :

Objet
(-) attributs
(+) méthodes()

(-) signifie « privé ».

(+) signifie « public ».

label
(+) id (+) innerHTML

input
(+) id (+) type (+) value

| <label id='lbl\_nom'>Nom : </label>

| <input type='text' id='nom' value='Tintin' />

Ces 2 instructions affiche ceci :

Nom :

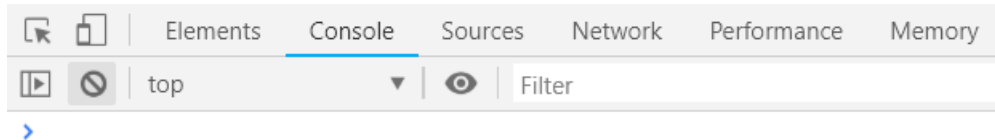
La modification des attributs value et innerHTML avec du code JavaScript permettra de modifier le visuel.

## 2.1 - JAVASCRIPT CONSOLE : LA STRUCTURE SÉQUENTIELLE

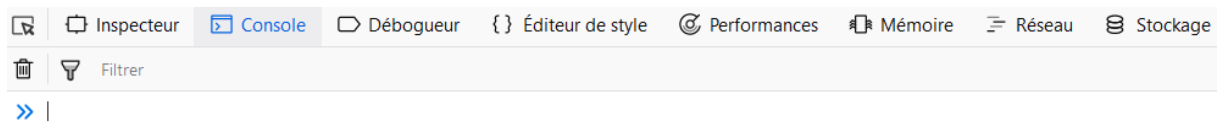
La structure séquentielle signifie que toutes les instructions sont exécutées en séquence, les unes après les autres, toujours dans le même ordre.

Lancer votre navigateur préféré (Chrome ou Firefox ou Edge ...), appuyez sur la touche F12. Sélectionnez « Console ».

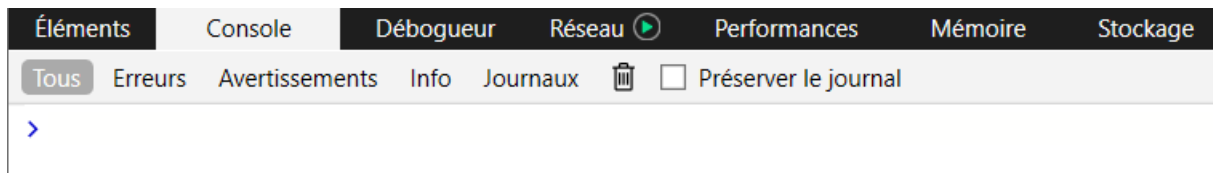
Chrome :



Firefox :



Edge :



Tapez une à une les commandes suivantes et validez à chaque fois avec la touche ENTRÉE :

```
var a = 3 ;  
var b = 5 ;  
var r = a + b ;  
console.log(r) ;  
  
> var a = 3 ;  
< undefined  
-----  
> var b = 5 ;  
< undefined  
-----  
> var r = a + b ;  
< undefined  
-----  
> console.log(r) ;  
8
```

Le point-virgule (;) à la fin de chaque instruction est facultatif.

```
var nom_de_variable
```

définit une variable (espace mémoire).

```
var nom_de_variable = valeur ;
```

définit une variable et lui affecte une valeur.

```
console.log(nom_de_variable)
```

affiche le contenu de la variable à la console.

## 2.2 - JAVASCRIPT CONSOLE : LA STRUCTURE CONDITIONNELLE

La structure conditionnelle signifie qu'une instruction (ou plusieurs) sera exécutée si une condition est remplie.

Tapez une à une les commandes suivantes :

```
var age = 18
if(age >= 18) { console.log("Majeur") ; } else { console.log("Mineur") ; }

> var age = 18
  if(age >= 18) { console.log("Majeur") ; } else { console.log("Mineur") ; }
  Majeur
```

et puis ceci :

```
var age = 17
if(age >= 18) { console.log("Majeur") ; } else { console.log("Mineur") ; }

> var age = 17
  if(age >= 18) { console.log("Majeur") ; } else { console.log("Mineur") ; }
  Mineur
```

La syntaxe est donc la suivante :

```
if(condition) {
  instructions ;
} else {
  instructions ;
}
```



## 2.3 - JAVASCRIPT CONSOLE : LA STRUCTURE ITÉRATIVE

La structure itérative signifie qu'une instruction sera exécutée plusieurs fois tant que la condition est satisfaite.

Tapez la ligne de code suivante.

```
| for(var i = 1; i <= 12; i++) console.log(i);
```

```
> for(var i = 1; i <= 12; i++) console.log(i);
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

```
11
```

```
12
```

La syntaxe est donc la suivante :

```
for(var i = début ; i <= fin ; i++) {  
  instructions ;  
}
```

## 2.4 - JAVASCRIPT CONSOLE : LES TABLEAUX

Un tableau est une variable à laquelle on associe plusieurs valeurs.

```
var jours = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche'];  
for(var i = 0 ; i < jours.length ; i++) {  
    console.log(jours[i]) ;  
}
```

```
> var jours = ['Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi', 'Samedi', 'Dimanche'] ;  
for(var i = 0 ; i < jours.length ; i++) {  
    console.log(jours[i]) ;  
}
```

Lundi

Mardi

Mercredi

Jeudi

Vendredi

Samedi

Dimanche

La syntaxe pour la déclaration et le remplissage d'un tableau est donc la suivante :

```
var nom_du_table = ['valeur_1', 'valeur_2', etc] ;
```

La syntaxe pour la récupération du nombre d'éléments d'un tableau est donc la suivante :

```
nom_du_tableau.length
```

La syntaxe pour la récupération d'un élément d'un tableau est donc la suivante :

```
nom_du_tableau[indice]
```

## 2.5 - JAVASCRIPT CONSOLE : LES FONCTIONS

(facultatif)

Une fonction est un bloc d'instructions défini par un nom.  
Elle peut posséder des paramètres.  
L'exécution d'une fonction se fait par son nom.

```
// Une fonction
function addition(a, b) {
    var r ;
    r = a + b ;
    return r ;
}

// Appel de la fonction
console.log(addition(3, 5)) ;
```

Version simplifiée :

```
function addition(a, b) {
    return a + b ;
}

console.log(addition(3, 5)) ;

> function addition(a, b) {
    return a + b ;
}

console.log(addition(3, 5)) ;
8
```

## 2.6 - PAGE WEB : AUTHENTICATION

Créez un dossier nommé `prepa_algo_js`.  
Dans ce dossier créez un sous-dossier nommé `html`.  
Créez aussi un autre sous-dossier nommé `js`.  
Dans le dossier `html` créez un fichier nommé `authentification.html`.

# Authentification

Pseudo :

MDP :

Message

```
<!DOCTYPE html>
<html>
  <head>
    <title>authentification.html</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
  </head>
  <body>
    <div>
      <h1>Authentification</h1>
      <form>
        <label for="pseudo">Pseudo : </label>
        <br>
        <input type="text" name="pseudo" id="pseudo" value="p" />
        <br>
        <label for="mdp">MDP : </label>
        <br>
        <input type="password" name="mdp" id="mdp" value="b" />
        <br>
        <input type="button" value="Valider" name="btValider" id="btValider" />
        <br>
        <label id="lblMessage">Message</label>
      </form>
    </div>
    <script src="../js/authentification.js"></script>
  </body>
</html>
```

Dans le dossier js créez le fichier nommé authentication.js.

```

/*
authentication.js
*/

/**
 *
 * @returns {undefined}
 */
function init() {
    // Quand l'utilisateur clique sur le bouton "valider"
    // On sollicite la fonction valider
    document.getElementById("btValider").onclick = valider;
} /// init

/**
 *
 * @returns {undefined}
 */
function valider() {
    // Déclaration d'une variable et affectation d'une valeur
    var lsMessage = "OK";
    // Récupération d'une saisie de l'utilisateur
    var pseudo = document.getElementById("pseudo").value;
    var mdp = document.getElementById("mdp").value;
    // Test des valeurs saisies
    // trim() enlève les espaces avant et après
    // Si le pseudo est vide ou si le mdp est vide alors
    if (pseudo.trim() === "" || mdp.trim() === "") {
        // Affectation de "KO" à la variable lsMessage
        lsMessage = "KO";
    }
    // Affichage d'une valeur (OK ou KO) dans le <label>
    document.getElementById("lblMessage").innerHTML = lsMessage;
} /// init

// Au chargement de la page HTML on sollicite la fonction init
window.onload = init;

```

### Exercices :

- 1) pré-initialiser pseudo et mdp à d'autres valeurs
- 2) effacer le texte du <label> du bas de page
- 3) message d'erreur : un pour chaque cas
- 4) internationaliser la page (2 langues) à partir d'un clic sur un drapeau

## 2.7 - PAGE WEB : INSCRIPTION

Dans le dossier nommé html créez un fichier nommé inscription.html.  
Dans le dossier nommé js créez un fichier nommé inscription.js.

# Inscription

Pseudo :

MDP :

Mot de passe visible : ☐

Email :

Ressaisissez votre Email :

Ville :

Date de naissance :

Message

Le <head> est le même que précédemment.

### Exercice :

A vous de coder la page HTML !

Les listes sont pré-remplies avec du JavaScript.

La première liste affiche des valeurs allant de 1 à 31.

La deuxième liste affiche des valeurs allant de 1 à 12.

La troisième liste affiche des valeurs allant de 1900 à 2019.

Dans un deuxième temps vous afficherez dans la deuxième les mois des mois (Janvier, Février, ...) qui auront été stockés au préalable dans un tableau.

Une partie du code JavaScript

```

/*
inscription.js
*/

/**
 *
 * @returns {undefined}
 */
function init() {
    // Quand l'utilisateur clique sur le bouton "valider"
    // On sollicite la fonction valider
    document.getElementById("btValider").onclick = valider;
    // Création d'un tableau
    var villes = new Array();
    // Remplissage du tableau
    villes[0] = "Sélectionnez";
    villes[1] = "Paris";
    villes[2] = "Lyon";
    villes[3] = "Marseille";
    // Remplissage de la liste déroulante
    var lsOptions = "<option>Sélectionnez</option>";
    for (var i = 0; i < villes.length; i++) {
        lsOptions += "<option value=\"" + villes[i] + "\">" + villes[i] + "</option>";
    }
    document.getElementById("villes").innerHTML = lsOptions;

    // jours, mois et années
    var lsOptions = "<option>Sélectionnez</option>";
    for (var i = 1; i <= 31; i++) {
        lsOptions += "<option>" + i + "</option>";
    }
    document.getElementById("jours").innerHTML = lsOptions;

    // Complétez

} /// init

/**
 *
 * @returns {undefined}
 */
function valider() {
    // Complétez
} /// valider

// Au démarrage on sollicite la fonction init
window.onload = init;

```

### Exercice :

terminez le code de la fonction init()  
 écrivez le code de la fonction valider().