

# RÉSUMÉ 4 - RAJOUTER DU JAVASCRIPT

Lien vers l'activité : [Rajouter du javascript](#)  
www.infoforall.fr - Dernière modif. : 25 01 2021



## 1 - INTÉGRATION DU JAVASCRIPT

### 1.1 - INTÉGRÉ À HTML OU FICHER À PART ?

- Méthode 1 (à éviter sauf essai) : on le place dans une balise HTML `<script>` :

```
1 <script>
2 alert("Hello World !");
3 </script>
```

- Méthode 2 (à privilégier) : on le place dans un fichier .js indépendant et on indique son existence au navigateur via une balise `<script>`.

```
1 <script src="mes scripts.js"></script>
```

### 1.2 - ÉVÉNEMENT ONLOAD

Le navigateur Web applique le code lu dans le fichier HTML de façon séquentielle : dès qu'il tombe sur la balise `script`, il lance donc la requête pour récupérer et exécuter le fichier javascript. Il faut donc :

- soit placer un événement "attente du téléchargement complet" si on utilise `script` dans la balise `head`

```
1 function demarrage() {
2     alert("Hello World !");
3 }
4
5 window.addEventListener("load", demarrage);
```

L5 : surveillance d'événement avec `addEventListener` : lorsque la page est chargée, cela lance automatiquement la fonction `demarrage`.

L2 : `alert` permet de lancer une fenêtre pop-up.

- soit placer la balise `script` juste avant la balise de fermeture de `body`.

### 1.3 - DIFFÉRENCE PYTHON ET JAVASCRIPT

- Chaque instruction finit nécessairement par un **point-virgule** (ligne 2 et ligne 5)
- On déclare les fonctions avec le mot-clé **function**
- La tabulation n'a aucun sens de structuration réelle de blocs.
- C'est l'accolade d'ouverte `{` et l'accolade de fermeture `}` qui construit les blocs.
- Les commentaires sur une ligne se font avec `//`
- Les commentaires multilignes se font avec `/*` et `*/`

## 2 - VARIABLES AVEC JAVASCRIPT

### 2.1 - DÉCLARATION DE VARIABLES LOCALES AVEC VAR ET LET

- En **Javascript**, les variables sont globales en lecture et en écriture par défaut !
- Seule exception : les **paramètres des fonctions**, qui sont de base des variables locales.
- Préfixer la **première affectation** d'une variable par **var** la rend locale à la fonction en cours
- Préfixer la **première affectation** d'une variable par **let** la rend locale au bloc `{ }` en cours.

```
1 function calcul(a, b) {
2     var reponse = a + b;
3     return reponse
4 }
5
6 function demarrage() {
7     var rep = calcul(6,4);
8     alert(rep);
9 }
10
11 window.addEventListener("load", demarrage);
```

## 2.2 TYPAGE DES VARIABLES

JavaScript possède moins de types natifs de variables que Python

- **number** (qui regroupe dans les integers et les floats) ,
- **boolean**,
- **string** et
- **undefined** (qu'on peut voir comme une sorte de None).

```
1  function calcul(a, b) {
2      /* Fonction qui renvoie l'addition des arguments a et b */
3      var reponse = a + b;
4      return reponse
5  }
6
7  function demarrage() {
8      var rep = calcul(2, "5");
9      alert(rep);
10 }
11
12 window.addEventListener("load", demarrage);
```

## 2.3 - TENTER DE CONVERTIR

`parseInt("2")` va ainsi renvoyer 2.

`parseFloat("2.5")` va ainsi renvoyer 2.5.

## 3 - DEBUG

On peut afficher des choses dans la **console (cachée)** (Développement Web - Console Web dans Firefox) du navigateur. Cela permet de voir le contenu des variables ... Bref, de faire le debug.

On pourra alors y afficher des choses en utilisant la méthode

**log** sur l'objet **console** : `console.log(rep)` ;

On peut également utiliser Réseau pour observer l'ordre des téléchargements.

www.infoforall.fr

