

# Notion de JavaScript

## Programmation événementielle

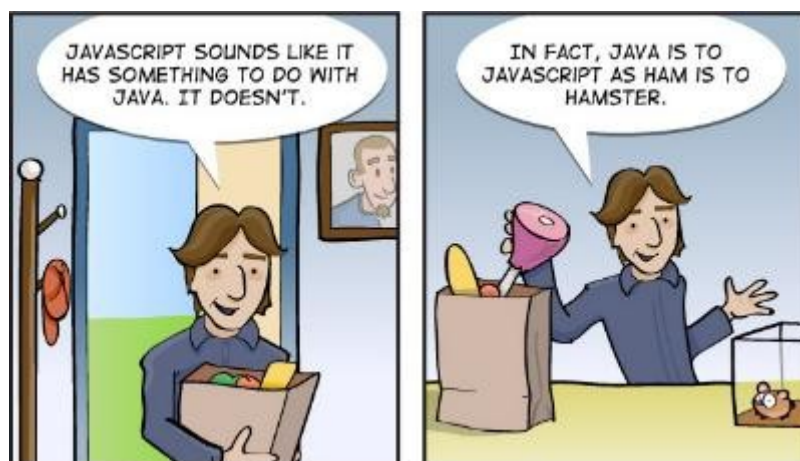
Jusqu'à présent nous avons utilisé comme **paradigme de programmation**, c'est-à-dire comme façon de programmer **la programmation impérative**. Dans cette façon de programmer, les instructions s'enchaînent les unes à la suite des autres avec des tests conditionnels, des boucles et des fonctions.

Le JavaScript est un langage multi-paradigmes. Il peut être utilisé de façon impérative, mais on va l'utiliser pour découvrir **la programmation événementielle**.

Avant cela, voici une courte histoire de ce langage.

Le langage JavaScript a été créé en 1995 par la société Netscape pour être intégré à leur logiciel phare à cette époque : Le navigateur Web Netscape. Le navigateur Firefox en est le lointain descendant. Dès 1996, Microsoft intègre sa propre version du langage à son navigateur Internet Explorer (dont le descendant est le navigateur Edge). Le langage JavaScript devient alors le langage incontournable pour la programmation Web côté client ce qu'il est toujours à l'heure actuelle après de nombreuses évolutions afin de l'améliorer et de le sécuriser. Depuis quelques années, il est aussi utilisé côté serveur.

JavaScript a été conçu en voulant le faire ressembler au langage Java tout en le simplifiant afin qu'il puisse être accessible même aux apprentis programmeurs. Cela ne l'empêche pas d'être un langage très puissant, adapté au Web et qui demande du temps d'apprentissage pour en utiliser toutes les facettes.



JavaScript est très souvent utilisé sur la machine du client web (votre navigateur web, à *distinguer de celle du serveur*) pour rendre la page interactive. Un clic, le survol d'un élément, l'appuie sur une touche ou le chargement d'une image sont des événements qui peuvent déclencher l'appel d'une fonction JavaScript. C'est en ce sens que c'est un langage événementiel.

# Rudiments du langage Javascript

Ce langage est par de nombreux points similaire au Python mais sa syntaxe diffère un peu.

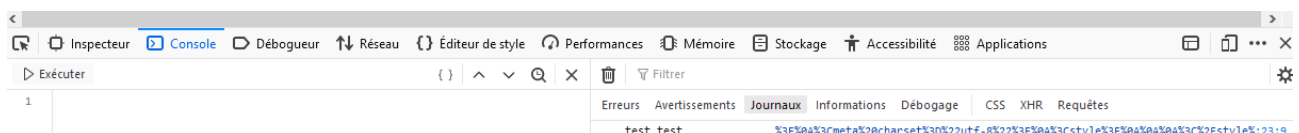
Les types de base :

Nombre	Entier(243) ou flottant(345.4)
Chaînes	"un mot" ou 'un mot'
Booléen	true ou false

Les principales commandes. La grande différence avec Python est que les blocs sont délimités par des accolades au lieu d'indentations et que les lignes doivent se terminer par des points-virgules.

Déclaration	let var ou let var = expression;
Affectation	var = expression
Conditionnelle	if (expression) { instructions} if (expression) { instructions;} else {instructions;}
Boucle bornée	for (i = 0 ; i < n ; i += 1) {instructions;}
Boucle non bornée	while (expression) { instructions }
Fonction	function NomFonction(parametres) { instructions;}
Valeur de retour	return
Message dans la console pour le débogage	console.log(chaine)
Commentaires	// Commentaires /* Commentaires */

Lors du débogage, utiliser la fonction console.log qui permet d'avoir un message dans la console développeur (Ctrl + Maj + I) puis sélectionner les onglets afin de n'avoir que les onglets « console » et « journaux » sélectionnés comme dans cette copie d'écran :



## Exemple

Vous allez découvrir trois versions d'une page contenant un programme JavaScript qui incrémente un compteur. La première version est la plus simple et la dernière correspond aux bonnes pratiques actuelles.

**Dans votre navigateur ouvrez l'URL : <https://flaustriat.frama.io/web/compteurJS0.html>**

Cliquez plusieurs fois sur le bouton « Cliquez-moi ! ». Cette page contient un bouton et affiche le nombre de fois où l'utilisateur a appuyé sur ce bouton.

Pour remettre le compteur à zéro rafraichissez la page (Ctrl + R).

Vous pouvez voir le code source de la page avec Ctrl + U. C'est le code qui est affiché ci-dessous

Il s'agit de programmation événementielle car la fonction `incremente` est appelée lors de l'événement « l'utilisateur clique sur le bouton ».

### Explication du fonctionnement du compteur.

Cet exemple se veut simple et ne respecte pas les bonnes pratiques actuelles. Néanmoins vous trouverez encore ce type d'écriture dans de nombreuses pages Web.

```
<html lang="fr">
  <head>
    <meta charset="utf-8"/>
    <meta name="auteur" content="NSI"/>
    <title>Compteur Javascript</title>
    <script type="text/javascript">
      let compteur = 0;
      function incremente() {
        compteur = compteur + 1;
        let d = document.querySelector('#valeur');
        d.innerHTML = compteur;
      }
    </script>
  </head>
  <body>
    <button onclick="incremente()">Cliquez-moi!</button>
    <span id="valeur">0</span>
  </body>
</html>
```

### Le code HTML

La balise HTML `script` sert à mettre du code dans une page HTML. L'attribut `type` précise qu'il s'agit du langage JavaScript. Entre les deux balise `script` se trouve du code JavaScript.

La balise `button` (bouton en anglais) permet d'afficher un bouton. L'attribut `onclick` permet de définir une action lorsque l'utilisateur clique sur le bouton. Ici, cela appelle la fonction `incremente`.

La balise HTML `span` permet d'afficher du texte sur une ligne et n'a pas de signification autre que de permettre d'appliquer une mise en forme particulière au texte entre balises. Cette balise est identifiée de façon unique dans la page par son `id` qui est `valeur`.

### Le code JavaScript

En JavaScript, les lignes de commandes se terminent par un « ; ».

`let compteur = 0;` → Permet de déclarer une variable globale nommée `compteur` que l'on initialise à 0.

La ligne suivante est l'entête de la fonction `incremente`.

`compteur = compteur + 1;` → Incrémente la variable `compteur` de 1

`let d = document.querySelector('#valeur');` → Déclare une variable `d` et lui affecte l'objet qui correspond à la balise d'id `valeur`.

`d.innerHTML = compteur;` → Affecte contenu entre les balises du span d'id `valeur` la valeur de `compteur`.

Le compteur augmente tant que la page n'est pas rechargée. Lorsque la page est rechargée (Ctrl+R) la variable `compteur` est remise à zéro.

Une bonne pratique qui permet de mieux structurer et maintenir le code est de séparer le JavaScript du HTML.

**Ouvrez l'URL <https://flaustriat.frama.io/web/compteurJS1.html>**

Regardez le code source. L'unique changement est que le JavaScript est remplacé par un appel à l'aide de l'attribut `src` au fichier `compteur.js`.

L'attribut `defer` permet d'assurer que le fichier HTML est chargé avant d'exécuter le JavaScript.

```
<script type="text/javascript" src="compteur.js" defer >
```

Cliquez sur le lien `compteur.js`, vous retrouverez le code JavaScript précédent.

**Ouvrez l'URL <https://flaustriat.frama.io/web/compteurJS2.html>**

Regardez le code source. Il n'y a plus d'attribut `onclick` dans la balise `button`

Cliquez sur `compteur2.js`. Il contient deux nouvelles lignes.

```
let b = document.querySelector('#compt');
b.addEventListener("click",incremente);
```

La première, permet de créer un objet `b` qui correspond au bouton d'id `compt` dans le code HTML. La seconde ajoute un gestionnaire d'événements qui à l'événement "click" sur l'objet `b` appelle la fonction `incremente`.

JavaScript permet de gérer une centaine d'événements, comme le double-clic, le fait de passer la souris au dessus d'un élément, de mettre le focus sur un élément d'un formulaire.

Pour en savoir plus:

<https://developer.mozilla.org/fr/docs/Web/API/EventListener>

Cette dernière façon de faire permet de séparer au mieux le code HTML de celui du JavaScript. Un intérêt est que l'on peut ajouter un gestionnaire d'événements sur un élément du code HTML sans modifier le code HTML.

Pour tester du code les développeurs « frontend » utilisent des interfaces permettant de modifier en parallèle les codes HTML, css et JavaScript. C'est ce que nous allons utiliser maintenant.

Consigne →



Fenêtres pour le code →

Rendu du code →

## Exercice 1

Ouvrez : <https://capytale2.ac-paris.fr/web/c-auth/list?returnto=/web/code/6e32-29554>

Le but est de compléter le code de la fonction `remise_a_zero` afin que le bouton de remise à zéro du compteur fonctionne.

La fonction `remise_a_zero` devra :

- Affecter la valeur 0 à la variable `compteur` ;
- Remplacer le contenu de la balise `span` d'id `valeur` par le contenu de la variable `compteur`.

Vous pourrez prendre pour exemple la fonction `incremente`.

## Exercice 2

Ouvrez l'URL <https://capytale2.ac-paris.fr/web/c-auth/list?returnto=/web/code/a0c9-29868>

Il s'agit de faire un convertisseur entre les degrés Celsius et les degrés Fahrenheit.

Compléter le script afin que ce programme fonctionne.

Vous devrez compléter la fonction `celsiusFahrenheit` afin qu'elle renvoie la température donnée en degré Celsius en degré Fahrenheit. On rappelle que si  $t$  est une température donnée en degré Celsius alors  $t * 9/5 + 32$  est la température en degré Fahrenheit.

Vous devrez aussi modifier la ligne `text2` afin qu'elle affiche la température convertie en Fahrenheit à l'aide de la fonction `celsiusFahrenheit`.

### Exercice 3

Ouvrez l'URL <https://capytale2.ac-paris.fr/web/c-auth/list?returnto=/web/code/f132-29912>

Il s'agit de rendre actif les boutons qui permettent de mettre en avant les citations.  
Comme sur cette copie d'écran:

#### Quelques citations

1. La révolution informatique fait gagner un temps fou aux hommes, mais ils le passent avec leur ordinateur !
2. Quelle prétention de prétendre que l'informatique est récente : Adam et Eve avaient déjà un Apple !
3. Informatique : Alliance d'une science inexacte et d'une activité humaine faillible.
4. L'informatique, ça fait gagner beaucoup de temps... à condition d'en avoir beaucoup devant soi !



Dans la consigne, les instructions sont données et à remettre dans le bon ordre.

### Exercice 4 - devinez le nombre

Il s'agit de programmer le jeu devine nombre en HTML et JavaScript.

Ouvrez l'URL: <https://capytale2.ac-paris.fr/web/c-auth/list?returnto=/web/code/b0f2-30010>

Vous pouvez bien sûr personnaliser l'apparence en modifiant le fichier CSS et ajouter des fonctionnalités ( compteur de nombre total de parties jouée, nombres déjà jouer, imposer de trouver le nombre en moins de 10 essais, etc.)

### Exercice 5 - Tic tac toe

Ouvrez l'URL <https://capytale2.ac-paris.fr/web/c-auth/list?returnto=/web/code/869c-30051>

Vous devez remettre les instructions dans le bon ordre afin que lorsque vous cliquez sur les cases elles soient complétées successivement par X et O.

Vous pouvez ajouter des fonctions pour que le programme détermine le gagnant ou s'il y a match nul.

#### Sources:

[https://framagit.org/patrice.thibaud/nsi\\_1ere\\_lmdf/-/blob/master/docs/11\\_ihm\\_web/c11\\_p2\\_evenementiel/c11\\_evenementiel.md](https://framagit.org/patrice.thibaud/nsi_1ere_lmdf/-/blob/master/docs/11_ihm_web/c11_p2_evenementiel/c11_evenementiel.md)

#### Pour aller plus loin:

- Tutoriel pour commencer:  
<https://developer.mozilla.org/fr/docs/Web/JavaScript/Guide>
- Référence du langage Javascript:  
<https://developer.mozilla.org/fr/docs/Web/JavaScript>
- "Cheat Sheet" JavaScript:  
<https://htmlcheatsheet.com/js/>