

Introduction à JavaScript

I) Historique

JavaScript est un outil majeur dans le développement d'une page **web interactive**. C'est un langage de programmation **orienté objet** qui permet d'interagir avec les éléments d'une page web. On pourra changer la couleur d'un texte, inclure ou exclure des images, vérifier si un formulaire est bien rempli, ...

JavaScript a été développé 1995 par **Brendan Eich**, il es standardisé sous le nom de **ECMAScript** en 1997. JavaScript est une implémentation du **ECMAScript** par la **fondation Mozilla**.

Il y a eu d'autres implémentations du **ECMAScript**, on peut citer **Microsoft** avec **Jscript** intégrer dans **internet explorer (IE)** jusqu'à sa version 9.

Il existe plusieurs sites internet traitant de JavaScript.

Le site internet de **Mozilla** contient toute la documentation de JavaScript, elle y est mise à jours régulièrement, le lien ici :

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

Il existe aussi des outils de développement sur internet pour visualiser facilement l'implémentation de JavaScript.

II) JS Bin

Le site **JS Bin** propose un outils de développement avec des onglets séparés pour visualiser son code.

Le lien est ici : <https://jsbin.com/hafuzelifo/edit?html,js,console,output>

The screenshot shows the JS Bin online development tool interface. The browser window displays the URL <https://jsbin.com/hafuzelifo/edit?html,js,console,output>. The interface features a top navigation bar with tabs for HTML, CSS, JavaScript, Console, and Output. The 'JavaScript' tab is selected and highlighted with a red box. Below the tabs, there are three main panels: 'Code HTML' on the left, 'Code JavaScript' in the center, and 'Console de développement Affiche les erreurs' on the right. The 'Output' panel on the far right shows the rendered web page. The 'Code JavaScript' panel contains the following code:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <meta name='viewport' content='width=device-width, initial-scale=1'>
  <title>JS Bin</title>
</head>
<body>
</body>
</html>
```

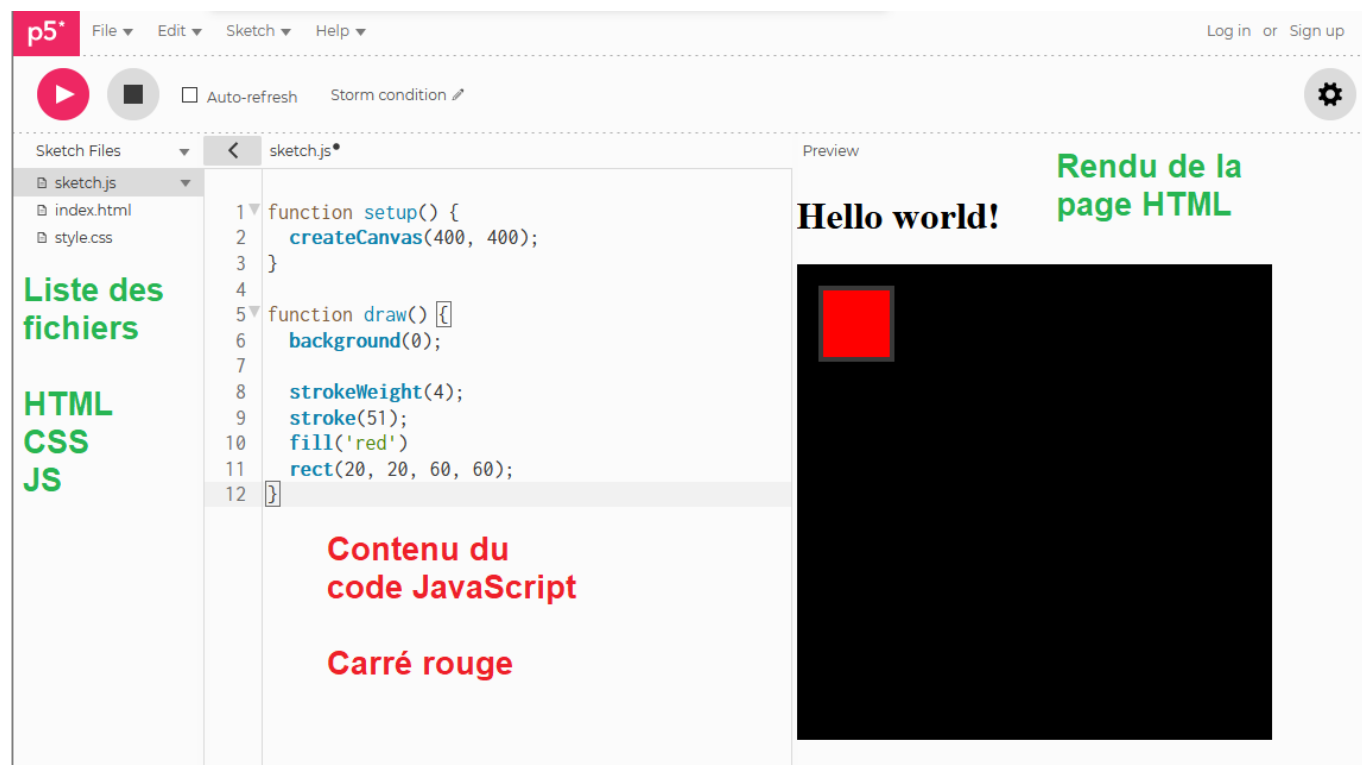
 The 'Console' panel is empty. The 'Output' panel shows the rendered HTML page. The interface also includes a 'New bin' button, a 'Textarea editor mode' checkbox, and a 'Upgrade to pro now' button.

III) Processing

Nous utiliserons aussi le [site P5.js](https://editor.p5js.org/) qui intègre un éditeur de développement. P5 fait référence à une **librairie JavaScript** de fonctionnalités : Processing.

P5 est donc la 5ième version de **Processing**, elle permet entre autre d'intégrer plusieurs éléments graphiques.

Le lien vers l'éditeur : <https://editor.p5js.org/>



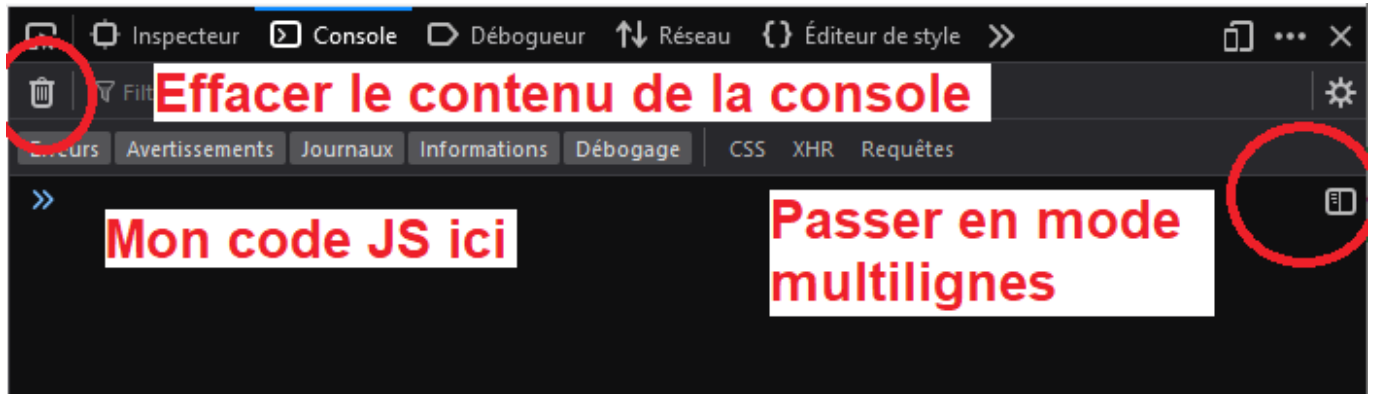
Le contenu du fichier HTML est le suivant :



IV) La console navigateur.

Pour exécuter du JavaScript nous pouvons aussi utiliser la console de développement des navigateurs web. Pour cela on utilise la commande **ctrl+maj+i** ou l'inspecteur de code html (**clik droit + examiner l'élément ..**) ou la touche **F12**.

Sous Firefox on obtient cette console.



Voici un petit code qui calcule la somme de 2 nombres



V) Le code JavaScript

Le JavaScript va nous permettre de rendre nos pages web dynamique, c'est-à-dire que l'on va pouvoir interagir sur les différents éléments du code HTML pour changer son contenu.

a) Inclure notre code JavaScript

Pour inclure un fichier de script, on utilise la balise HTML `<script src = ... > </script>`.

On peut placer cette balise dans différente partie du fichier HTML. : en-tête ou corps. Il est préférable de la placer à **la fin de la page web** juste avant de fermer la balise body car à l'exécution de la page, cette dernière ne s'affiche pas si le script contient des bugs ou si le script prend du temps à exécuter une instruction.

Ligne de commande pour y inclure un fichier : `<script src="scripts/main.js"></script>`

Remarque :

1) L'**indentation** ne joue pas le même rôle que en Python. On peut mettre des indentations différentes sur un même bloc d'instructions.

2) Quand on écrit une instruction JavaScript, il doit généralement mettre un **point virgule ;** à la fin de l'instruction. L'oubli d'un point virgule est souvent source d'erreur à l'exécution.

Mettre des commentaires : Il y a 2 manières de commenter son code.

Commentaires sur une ligne : on utilise `// mon commentaire`

Commentaires sur une ligne : on utilise `/* mon commentaire sur plusieurs lignes */`

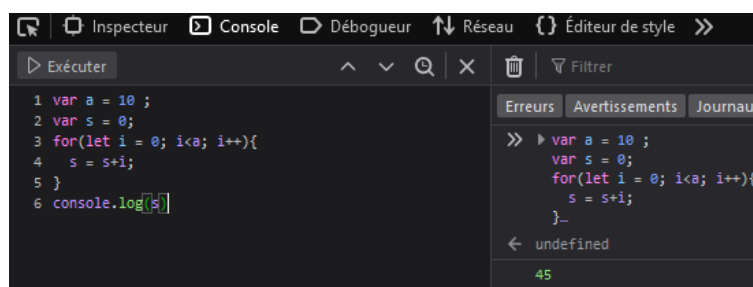
b) Déclarer une variable

Pour déclarer une variable on utilise le mot clé `var` et le signe `=` pour indiquer une affectation.

On peut aussi utiliser le mot clé `let` mais la variable sera alors accessible que dans le bloc ou elle a été déclarée. On pourra utiliser par exemple plusieurs fois le même nom de variable dans des blocs séparés. Attention si on utilise une déclaration avec `let`, la variable n'est pas accessible en dehors du bloc.

Exemple :

```
var a = 10;
var s = 0;
for (let i = 0 ; i<a ; i++) {
    s = s+i ;
}
console.log(s) ;
```



On a créé ici 2 variables `a` et `s` qui sont accessibles sur l'ensemble du code. Dans la **boucle** `for` on déclare la variable `i` mais uniquement dans la boucle avec la commande `let`. On ne pourra pas y accéder en dehors de cette boucle. Ici la boucle `for` parcourt les éléments `i` tant que `i < a`. Le symbole `i++` est un code d'incrément de la variable `i`. `i++` signifie que l'on ajoute 1 à la valeur de `i`.

L'instruction `console.log(s)` ; permet de visualiser la valeur de `s`. Elle joue le même rôle que la commande `print` en Python.

Application : Reproduire ce scripte dans la console (touche F12 ou inspecter ou `ctrl+maj+i`)

VI) Exemples de code

a) Astuces de programmer

Dans la suite on présente plusieurs exemples. On rappelle qu'une instruction JavaScript se termine généralement par un point virgule ; .

Remarque :

1) Attention il y a des cas particuliers pour lesquelles on ne met pas de point virgule : fonction, boucle et instruction conditionnelle. Les instructions de ces dernières sont balisées par des accolades.

2) Pour nommer les variables ou les fonctions, elles commencent par une lettre minuscule et on n'utilise pas des caractères spéciaux (?/*) . Elles peuvent contenir des majuscules et des chiffres.

3) Le choix du nom de variable et de fonction reste important. Il doit refléter l'utilisation que l'on en fait. Il doit être court et sans ambiguïté. De plus en JavaScript, la pratique veut que le nom de variable ou fonction commence par une lettre minuscule et si il y a un nouveau mot on le fait commencer par une lettre majuscule.

Exemples :

```
var monNom = 'Bond' ;
var monPrenom = 'James' ;
var monMetier = 'agent secret' ;
```

b) Exemples de code

Les différents types de données

Variable	Explications	Exemples
Chaîne de caractères	Du texte entre guillemets	<code>let name = 'bob' ;</code>
Nombre	Nombre, entier ou flottant	<code>let nb = 10 ;</code> <code>let nb_pi = 3,1416 ;</code>
Booléen	Valeur true / false	<code>let cond = true ;</code>
Tableau	Identique à Python	<code>Let tab = [1,3, 'jack', [4,5]] ;</code> <code>tab[0] et tab[2]</code>
Objet	Tout est un objet en JavaScript il peut être stocké dans une variable.	<code>Let myVar = document.queySelector('h1')</code>

Les opérateurs

Opérateur	Explication	Symbole	Exemple
Addition	Ajouter 2 nombres Concaténation de deux chaînes	+	var a = 6+9 ; var b = 'Le beau ' + 'Jack' ;
Soustraction Multiplication Division puissance Modulo	Opération de base en maths	- * / ** %	let nb = 10 ; let nb_pi = 3,1416 ; let k = 25%4 ;
Assignment	Affecter une valeur à un variable	=	let a = true ;
Égalité	Renvoie true/false	===	15 === 15
Négation et inégale		! !==	!false 15 !== 5

La structure conditionnelle :

L'instruction `prompt` sert à demander une valeur à l'utilisateur.

L'instruction `alert` sert à afficher une boîte de dialogue avec le texte indiqué.

```
var iceCream = prompt('Aimes tu la glace au chocolat: oui /non');
```

```
if (iceCream === 'oui') {
    alert("J'adore la glace au chocolat !");
} else {
    alert("Oooh, mais j'aurais préféré au chocolat.");
}
```

Les fonctions :

On code ici la multiplication de 2 nombres. l'instruction `return` marche comme sur Python.

```
function multiply(num1,num2) {
    let result = num1 * num2;
    return result;
}
```

Les boucles for :

On affiche les carrés parfaits de 0 à 100

```
var k = 11;
var s = s;
for (let i = 0 ; i<k ; i++) {
    s = i**2 ;
    console.log(i) ;
}
```