

JS TP1- Hello World

Introduction à JavaScript

Introduction


JavaScript n'est pas Java

Pour commencer, tordons le cou à une erreur classique : **JavaScript** et **Java** ne sont en aucune manière apparentés. Il s'agit de deux langages différents. En effet, dans le cadre d'un site Web :

- ❖ **JavaScript** est généralement exécuté côté client, sur la machine de l'utilisateur où tourne le navigateur qui analyse et affiche la page Web.
- ❖ **JavaScript** peut également être exécuté côté serveur dans un environnement comme **Node.js**.
- ❖ **Java**, quant à lui, est un langage compilé qui peut être exécuté sur diverses plateformes, y compris côté serveur, mais aussi sur des appareils mobiles ou des applications de bureau.
- ❖ Un code **JavaScript** est appelé dans une page Web différemment d'une applet **Java**.
- ❖ Il est nécessaire de déclarer les variables en **Java**, alors que cette déclaration peut être implicite en **JavaScript**.

Javascript Hello World

Javascript est un langage interactif dynamique de page Web utilisé dans le développement Web.

 Créez un fichier helloworld.html avec le Bloc-notes et ouvrez-le dans votre navigateur pour voir la page Web

hello-world.html

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>First Web Page</title>
    <script type="text/javascript">
      document.write("HelloWorld");
    </script>
  </head>
  <body></body>
</html>
```

<html> : la balise indique au navigateur qu'il s'agit d'un document HTML.

<head> : tag est un conteneur de métadonnées sur le contenu et le jeu de caractères.

<title> : la balise définit le titre du document.

<body> : la balise définit le corps du document.

<script> : la balise est utilisée pour définir un script côté client (JavaScript).

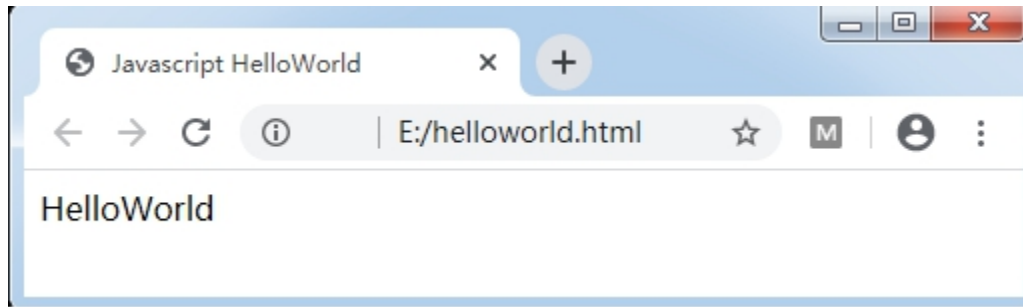
Tout votre code **javascript** devra être placé entre balises **<script>**

```
<script type="text/javascript">
  document.write("HelloWorld");
</script>
```

Il faut bien noter qu'une instruction en JavaScript se termine par un point-virgule ";", comme pour le CSS.


document.write est une **notation pointée**, elle est appelée via l'objet **document**.

Résultat :



Vous pouvez le placer partout dans la page, mais attention, chaque position aura une conséquence sur l'interprétation du JavaScript.

Ceci, nous le verrons quand nous étudierons le DOM ;).

 Ajouter un espace entre "Hello" et "World"

Ce qu'on ne voit pas

Modifier le script **hello-world.html**

```
<script type="text/javascript">
  // Ecrire dans le document
  document.write("HelloWorld");

  /*
    document.write("Hello World");
  */
</script>
```

Les **commentaires** en JavaScript sont des annotations dans le code qui ne sont pas exécutées par le moteur JavaScript. Ils permettent aux développeurs d'expliquer leur code, de le rendre plus lisible et de désactiver temporairement certaines parties du code sans les supprimer. Il existe deux types de commentaires en JavaScript :

- les commentaires sur une seule ligne, qui commencent par `//`, et
- les commentaires sur plusieurs lignes, qui sont encadrés par `/* */`.

Utiliser des commentaires aide à maintenir un code propre et compréhensible, surtout dans des projets collaboratifs ou complexes.

Ajouter dans le script **hello-world.html**

```
console.log("Bonjour tout le monde");
```

`console.log` est une fonction en JavaScript qui permet d'afficher des messages dans la console du navigateur ou de l'environnement d'exécution. Elle est souvent utilisée pour déboguer du code en affichant des informations, comme des variables, des objets ou des messages, afin de comprendre ce qui se passe à différents moments de l'exécution.

Le contenu passé à `console.log` s'affiche directement dans la console, ce qui en fait un outil très utile pour suivre l'état des données et repérer d'éventuelles erreurs.

`console.log` est une **notation pointée**, elle est appelée via l'objet `console`.

Raccourci : Ctrl + Shift +K : pour afficher la console (Firefox)

On peut aussi utiliser des simples quotes :

```
console.log('Bonjour tout le monde');
```

Insertion de code JavaScript dans un document HTML

Trois méthodes d'insertion

Il existe trois manières d'insérer du code **JavaScript** dans un document HTML :

Dans la balise

La manière la plus simple à mettre en œuvre est l'utilisation des attributs prévus dans les recommandations HTML pour cela : **onclick**, **onmouseover**, etc. Cette méthode présente **un inconvénient majeur**, celui de mêler le code destiné à donner la structure du document (le **HTML**) au code destiné à le mettre en quelque sorte en mouvement, le **JavaScript**. La mise à jour d'une telle page est difficile, de la même manière que gérer la mise en forme **CSS** à l'aide de l'attribut **style** est une pratique à déconseiller. Elle sera parfois utilisée dans le cas d'exercice pour aller vite.

```
<button onclick="console.log('Bonjour !');">Bonjour</script>
```

Dans la balise script

Une deuxième méthode, analogue à l'utilisation d'une feuille de style interne en **CSS**, consiste à faire appel à l'élément **script** et à insérer le code **JavaScript** à l'intérieur de cet élément. Par exemple,

```
<script type="text/javascript">
  console.log("Bonjour !");
</script>
```

L'attribut **type** n'est pas obligatoire. Par défaut, le navigateur considère que tout code à l'intérieur d'une balise script sans attribut type est du javascript (depuis HTML 5) Donc, si vous écrivez du JavaScript, vous pouvez omettre cet attribut.

```
<script>  
  console.log("Bonjour !");  
</script>
```

Cet élément peut être présent dans l'entête du fichier **HTML** tout comme l'élément **style** des feuilles de style internes. Il est aussi possible de l'insérer au cœur de la page **HTML**, comme descendant de l'élément **body**, mais cela complique encore une fois la maintenance de la page.

Dans un fichier .js

Une troisième méthode fait elle aussi appel à l'élément **script**, mais cette fois-ci le code **JavaScript** se trouve dans un fichier externe. Dans ce cas, l'élément **script** précise à l'aide de l'attribut **src** l'emplacement du fichier **JavaScript** :

```
<script src="../../scripts/aideSaisie.js">
```

Cette dernière méthode est la plus commode à mettre en œuvre si un script doit être utilisé sur **plusieurs pages**, mais elle doit être maniée avec précaution afin d'éviter toute perte d'informations. À la fin de ce cours, nous reviendrons sur de bonnes pratiques de codage à utiliser afin d'éliminer ce genre de soucis.

ES6 : type="module"

Cependant, si vous utilisez d'autres types de scripts, comme `type="module"` pour un module JavaScript, ou un langage de script différent, il devient important de préciser le type.

Exemple avec `type="module"` :

```
<script type="module">
  import { exampleFunction } from './module.js';
  exampleFunction();
</script>
```

Le type module donne accès aux importations et exportations, entre autres.

En résumé, pour le JavaScript standard, l'attribut `type` est facultatif, mais pour d'autres types de scripts ou des modules, il peut être nécessaire. (On y reviendra plus tard)

Présence d'une alternative

Il n'y a pas que des navigateurs graphiques supportant **JavaScript** qui soient susceptibles de consulter un document Web. Il existe des outils de consultation de pages qui ne supportent pas **JavaScript**, ou, ce qui est pire, n'en ont qu'un support partiel.

C'est la raison pour laquelle, à chaque fois que du contenu ou une fonctionnalité est ajouté à une page en faisant appel à **JavaScript**, il est nécessaire de prévoir une alternative sous un format accessible à ce genre d'outils de consultation (le plus souvent simplement du **HTML**), et fournissant à ces outils une information équivalente et des fonctionnalités permettant d'atteindre les mêmes buts.

Cette alternative doit être présente dans un élément **noscript**, **enfant de script** lorsqu'il est **descendant de l'élément body**.

Par exemple :

```
<script type="text/javascript">
  (...)
  <noscript>Information équivalente au contenu produit par le
script</noscript>
</script>
```

Parenthèse SEO

De plus, les robots indexeurs des moteurs de recherche **ne sont pas faits pour** parcourir le contenu JS.

En général, le js est désactivé pour les outils de référencement. Le contenu qui serait produit en **JavaScript est alors invisible**.

Le robot google peut indexer du JS, mais le contenu étant plus lent à être rendu côté navigateur, le **budget du crawl** peut être atteint plus rapidement sur du contenu rendu en JS.

De plus, si le contenu met plus de temps à charger car le JS doit contacter une API pour le contenu, le robot **peut quitter la page plus rapidement** ou **attribuer un mauvais score**.

Responsable SEO : “ C’est bizarre, quand on désactive le JavaScript, le site ne fonctionne plus”

Boîte de dialogue

En JavaScript, une boîte de dialogue permettent d'interagir avec l'utilisateur en affichant des messages

alert() : Affiche un message simple à l'utilisateur avec un bouton "OK". Elle est généralement utilisée pour informer l'utilisateur d'un événement ou d'une action.

```
alert("Bonjour !");
```



Exercice :

Ecrire un script qui au click d'un bouton afficher le message d'alert "Hello World"

prompt() : Affiche une boîte avec un champ de texte permettant à l'utilisateur de saisir une valeur, accompagnée d'un bouton "OK" et "Annuler". La valeur saisie est renvoyée si l'utilisateur clique sur "OK", ou **null** s'il clique sur "Annuler".

```
let nom = prompt("Quel est votre nom ?");
```

MEMO

Les commentaires

<code>//</code> : les commentaires uniques ne sont pas exécutés par javascript
--

<code>/* */</code> : les commentaires sur plusieurs lignes
--

Quotes

<code>""</code> or <code>' '</code> : est une séquence de caractères <code>""</code> comme : <code>"Apple"</code> or <code>'Apple'</code>

Affichage

<code>document.write()</code> : écrit le contenu HTML dans le document dans le navigateur

<code>console.log()</code> : Permet d'afficher un message dans la console. Elle n'est pas visible sur la page HTML

<code>alert()</code> : Affiche un message simple à l'utilisateur avec un bouton "OK".

Solution exercice :

```
<button onclick="alert('Hello World')">Cliquez ici</button>
```