

# Interaction Homme - Machine sur le WEB

---

## Partie 3 - Le javascript

---

Pour créer des pages web dynamique, le HTML ne suffit pas, le contenu des pages peut être calculé côté serveur (PHP) ou côté client (Javascript). L'interpréteur javascript est intégré dans tous les navigateurs.

Le langage javascript a été créé en dix jours dans les années 1995 par Brendan Eich, un informaticien américain alors employé chez l'éditeur Netscape Communication qui était à l'origine du navigateur Web Netscape Navigator (et qui était le premier à intégrer javascript).

Remarque : Le code du logiciel Netscape Navigator était ouvert et a servi de base au navigateur Firefox encore en usage actuellement.

Malgré son nom, JavaScript n'a rien à voir avec le langage Java, même si Brendan Eich affirme s'être inspiré de nombreux langage, dont Java, pour mettre au point JavaScript.

Notre but ici n'est pas d'apprendre un nouveau langage de programmation, mais juste d'étudier quelques exemples d'utilisation du JavaScript, notamment dans le cas des interactions entre un utilisateur et une page web.

### Premier contact avec javascript

Commençons par mettre en place notre environnement de travail :

#### Activité 1

Dans votre répertoire de travail (à créer), créez 3 fichiers vides : `index.html` , `style.css` et `script.js` en utilisant notepad++ (ou autre).

Remarque : il sera important d'enregistrer vos modifications au fur et à mesure.

#### Activité 2

Ouvrir le fichier `index.html` à l'aide d'un éditeur de texte, saisissez le code ci-dessous :

```

<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Débuter avec javascript</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p>Voici une page web qui ne fait pas grand-chose mais qui
      nous sera utile</p>
  </body>
  <script src="script.js"></script>
</html>

```

Rien de très nouveau dans ce code sauf la ligne ci-dessous qui permet d'exécuter le programme JavaScript contenu dans le fichier `script.js` :

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

Complétons maintenant notre feuille de style `style.css`

### Activité 3

Ouvrir le fichier `style.css` à l'aide d'un éditeur de texte, saisissez le code ci-dessous :

```

/* dans le fichier style.css */

h1
{
  text-align: center;
}

```

Complétons maintenant notre fichier `script.js` qui contiendra notre code javascript :

### Activité 4

Ouvrez le fichier `script.js` à l'aide d'un éditeur de texte, saisissez le code ci-dessous :

```

/* dans le fichier script.js */

alert("Le JavaScript fonctionne !")

```

Observez le résultat en ouvrant le fichier `index.html` dans le navigateur de votre choix.

Comme vous pouvez le constater, la page web s'affiche bien, mais nous avons en plus une fenêtre (souvent qualifiée de surgissante ou pop-up en anglais) qui apparaît. L'apparition de

cette fenêtre est bien évidemment due à l'instruction `alert` présente dans le `javaScript`.

Le but ici n'étant pas d'apprendre à programmer en `javaScript`, nous nous contenterons pour le moment de cette simple instruction `alert`.

Evidemment `javaScript` permet de faire bien plus de choses. En effet, on retrouve en `javaScript` les grands classiques des langages de programmation : variable, condition, boucle, fonction,...

Dans l'exemple ci-dessus, l'instruction `alert` est exécutée dès l'ouverture de la page web, il est tout à fait possible de lier l'exécution de certaines instructions `javaScript` à l'action d'un utilisateur (par exemple un clic sur un bouton).

## Utilisation d'un bouton

### Activité 5

Modifiez le code HTML en ajoutant la ligne commençant par `<button>` comme ci-dessous :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p>Voici une page web qui ne fait pas grand-chose mais qui nous sera utile</p>
    <button onclick="maFonction()">Cliquer ici</button>
  </body>
  <script src="script.js"></script>
</html>
```

Modifiez ensuite le code JavaScript comme suit :

```
/* dans le fichier script.js */

function maFonction() {
  alert("Le JavaScript fonctionne !")
}
```

Testez cette nouvelle page en ouvrant le fichier `index.html` avec le navigateur de votre choix (ou rafraichir la page si elle est encore ouverte dans le navigateur).

Comme vous pouvez le constater, l'instruction `alert` n'est plus exécutée à l'ouverture de la page web, mais uniquement dans le cas où l'utilisateur clique sur le bouton.

On a associé au bouton un événement `onclick`, en cas de clic sur la souris, la fonction JavaScript `maFonction()` est exécutée.

Si on s'intéresse au code JavaScript, on retrouve bien une fonction `maFonction()` qui est définie par l'instruction `function`, suivie du nom choisie `maFonction()`, le contenu de la fonction est entre deux accolades `{...}`

Remarque : cette écriture en JavaScript est équivalent à un `def maFonction():` en Python.

Entre l'accolade ouvrante et l'accolade fermante (qui délimite la fonction), on retrouve uniquement notre instruction `alert`. À l'ouverture de la page web cette instruction `alert` n'est pas exécutée, car elle se trouve dans une fonction. Le clic sur le bouton entraîne l'exécution de la fonction `maFonction()` et donc de l'instruction `alert`.

Il est évidemment possible de faire des choses bien plus complexes que l'affichage d'un simple pop-up avec JavaScript. Il est possible de modifier le style d'une balise, de modifier la classe (CSS) d'une balise ou encore de modifier le contenu d'une balise, ci-après quelques exemples.

## Activité 6

Modifiez le code HTML comme suit :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p id="monPara">Voici une page web qui ne fait pas grand chose</p>
    <button onclick="maFonction()">Cliquer ici</button>
  </body>
<script src="script.js"></script>
</html>
```

Modifiez le code JavaScript comme suit :

```
/* dans le fichier script.js */

function maFonction() { document.querySelector("#monPara").style.color="red";}
```

Testez cette nouvelle page en enregistrant vos fichiers et en ouvrant le fichier index.html

Dans l'exemple ci-dessous, nous avons déjà ajouté un id ("monPara") à la balise "p" dans notre code HTML.

Dans le code JavaScript :

```
document.querySelector("#monPara").style.color="red";
```

Cette ligne permet de modifier le style de la balise ayant pour id "monPara" : la couleur du texte devient rouge. Comme cette modification du style se trouve dans la fonction "maFonction()", cette modification sera effective uniquement si l'utilisateur appuie sur le bouton.

Il est possible de travailler plus "proprement" en utilisant les classes CSS.

## Activité 7

Modifiez le code HTML comme suit :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p id="monPara">Voici une page web qui ne fait pas grand-chose</p>
    <button onclick="foncRouge()">Rouge</button>
    <button onclick="foncVert()">Vert</button>
  </body>
<script src="script.js"></script>
</html>
```

Modifiez le code JavaScript comme suit :

```
/* dans le fichier script.js */

function foncRouge() {
  document.querySelector("#monPara").classList.remove("vert");
  document.querySelector("#monPara").classList.add("rouge");}

function foncVert() {
  document.querySelector("#monPara").classList.remove("rouge");
  document.querySelector("#monPara").classList.add("vert");}
```

Modifiez le code CSS comme suit :

```
/* dans le fichier style.css */

h1
{
    text-align: center;
}

.rouge
{
    color:red;
    font-size:20px;
}

.vert
{
    color:green;
    font-size:30px;
}
```

Après avoir analysé le code ci-dessus, testez cette nouvelle page en ouvrant le fichier `index.html` .

Dans l'exemple ci-dessus, nous avons maintenant 2 boutons, un clic sur le bouton `vert` permet d'exécuter la fonction `foncVert()` , un clic sur le bouton `rouge` permet d'exécuter la fonction `foncRouge()` , jusque là, rien de vraiment nouveau.

La fonction JavaScript `foncVert()` permet de modifier la classe CSS de la balise ayant pour id `monPara` .

Dans un premier temps, la ligne :

```
document.querySelector("#monPara").classList.remove("rouge");
```

permet de supprimer l'association entre la balise d'id `monPara` et la classe CSS `rouge` (si cette association n'existe pas, cette ligne n'a aucun effet).

Dans un deuxième temps, on associe la classe CSS `vert` avec la balise d'id `monPara` avec la ligne :

```
document.querySelector("#monPara").classList.add("vert");
```

Le principe est identique avec la fonction `foncRouge()` .

Il est également possible de modifier le contenu d'une balise HTML.

## Activité 8

Modifiez le code HTML comme suit :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <p id="monPara">Voici une page web qui ne fait pas grand chose</p>
    <button onclick="modifMessage()">Cliquez ici</button>
  </body>
<script src="script.js"></script>
</html>
```

Modifiez le code JavaScript comme suit :

```
/* dans le fichier script.js */

function modifMessage() { document.querySelector("#monPara").innerHTML =
  "Bravo, vous avez cliqué sur le bouton !"
}
```

Après avoir analysé le code ci-dessus, testez cette nouvelle page en ouvrant le fichier

index.html

Le contenu de la balise ayant pour id `monPara` est modifié grâce à la ligne :

```
document.querySelector("#monPara").innerHTML = "Bravo, vous avez cliqué sur le bouton !"
```

Il existe d'autres événements que "onclick", par exemple, il est possible de détecter le "survol" par le curseur de la souris d'un élément HTML.

## Activité 9

Modifiez le code HTML comme suit :

```

<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Le trio</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Le trio : HTML, CSS et JavaScript</h1>
    <div onmouseover="foncEntre()" onmouseout="foncQuitte()" id="maDiv">
  <p>Survolez-moi</p></div>
  </body>
<script src="script.js"></script>
</html>

```

Modifiez le code JavaScript comme suit :

```

/* dans le fichier script.js */

function foncEntre(){
document.querySelector("#maDiv").classList.remove("blanc");
document.querySelector("#maDiv").classList.add("rouge");}

function foncQuitte() {
document.querySelector("#maDiv").classList.remove("rouge");
document.querySelector("#maDiv").classList.add("blanc");}

```

Modifiez le code CSS comme suit :



```
h1
{
    text-align: center;
}

p
{
    text-align : center;
}

#maDiv
{
    width : 200px; height : 100px; margin : 0 auto; border : 2px solid black;
}

.rouge
{
    background-color:red;
}

.blanc
{
    background-color:white;
}
```

Après avoir analysé le code ci-dessus, testez cette nouvelle page en ouvrant le fichier `index.html`

L'événement `onmouseover` correspond bien au survol par le curseur de la souris d'un élément HTML.

L'événement `onmouseout` est lui déclenché quand le curseur de la souris quitte un élément HTML donné.

Il existe beaucoup d'autres événements que nous n'aborderons pas ici. Si vous voulez en savoir plus, vous pouvez consulter [ce site](#).

En résumé, le code HTML permet de générer des éléments graphiques qui seront affichés par un navigateur web, mais pas seulement : il est aussi possible de mettre en place dans le code HTML des événements. Un événement donné pourra déclencher l'exécution d'instructions JavaScript.