

Interaction avec l'utilisateur dans une page web: utilisation du JavaScript:

Nous avons déjà pu nous familiariser avec le couple HTML-CSS, en faite, le couple est plutôt un trio, car aujourd'hui un développeur web ne peut pas faire l'impasse sur le 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.

Avant d'entrer dans le vif du sujet, un petit rappel historique : JavaScript a été créé en dix jours par Brendan Eich en 1995. 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. Après des débuts chaotiques, il est, comme déjà dit plus haut, devenu incontournable dans le développement web.

Commençons par mettre en place notre environnement de travail:

Exercice 1: Dans votre répertoire de travail, créez 3 fichiers : index.html, style.css et script.js.

Exercice 2: Après avoir ouvert 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>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</p>
  </body>
  <script src="script.js"></script>
</html>
```

Rien de très nouveau dans ce code, à part le

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

qui permet d'exécuter le programme JavaScript contenu dans le fichier "script.js"

Exercice 3: Après avoir ouvert le fichier style.css à l'aide d'un éditeur de texte, saisissez le code ci-dessous:

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

Exercice 4: Après avoir ouvert le fichier script.js à l'aide d'un éditeur de texte, saisissez le code ci-dessous:

```
alert("Le JavaScript fonctionne !")
```

Exercice 5: Afin d'afficher la page web que nous venons de créer dans un navigateur web, cliquez sur le fichier "index.html"

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,...Si vous voulez en apprendre plus sur la programmation en JavaScript, je vous invite à consulter le site [openclassrooms](https://openclassrooms.com).

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).

Exercice 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>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:

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

Testez cette nouvelle page en cliquant sur le fichier index.html

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()" ("function maFonction()..." 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, voici quelques exemples:

Exercice 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="maFonction()">Cliquer ici</button>
  </body>
  <script src="script.js"></script>
</html>
```

Modifiez le code JavaScript comme suit:

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

Testez cette nouvelle page en cliquant sur 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, la ligne

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

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:

Exercice 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="foncRouge()">Rouge</button>
    <button onclick="foncVert()">Vert</button>
  </body>
  <script src="script.js"></script>
</html>
```

Modifiez le code JavaScript comme suit:

```
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: Après avoir analysé le code ci-dessus, testez cette nouvelle page

```
h1{
    text-align: center;
}
.rouge {
    color:red;
    font-size:20px;
}
.vert {
    color:green;
    font-size:30px;
}
```

en cliquant sur 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:

Exercice 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>
    <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:

```
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 cliquant sur 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.

Exercice 10: 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:

```
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 cliquant sur le fichier index.html

”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.