

## ÉVÈNEMENTS

- ▷ Histoire de l'informatique
- ▷ Représentation des données
- ▷ Traitement des données
- ▷ [Interactions entre l'homme et la machine sur le Web](#)
- ▷ Architectures matérielles et systèmes d'exploitation
- ▷ Langages et programmation
- ▷ Algorithmique

## 1. JavaScript

### 1.1. Introduction

En seconde, dans le cours de Sciences Numériques et Technologie, les élèves ont étudié HTML et CSS. Ce qui suit n'est donc qu'un rappel#

HTML (ou Hypertext Markup Language) permet de créer la structure d'une page internet.

Grâce à un jeu de balises, il permet de décomposer la page comme un traitement de texte : titre, sous-titre, section,...

On peut par exemple indiquer que l'on utilise un titre avec les balises ouvrante et fermante `<h1></h1>` ou indiquer l'utilisation d'une énumération avec `<ol></ol>` et `<li></li>` pour chaque item.

CSS (ou Cascade Style Sheets) permet la mise en forme des différents éléments HTML. On peut ainsi modifier la couleur ou la police des éléments précédents.

Les pages ainsi créées sont statiques c'est-à-dire que l'interaction avec l'utilisateur est réduite à la possibilité de cliquer sur un lien hypertexte présent dans la page.

On peut dynamiser la page internet de deux manières :

- ▷ soit du côté serveur avec PHP (ou Hypertext Preprocessor) qui peut, par exemple, ajouter le résultat d'une requête à une base de données dans la page qui sera fournie au navigateur;
- ▷ soit du côté client avec JavaScript qui peut, par exemple, faire apparaître des info-bulles contextuelles ou réaliser des animations.

### 1.2. Présentation

JavaScript est un langage créé en 1995 par Brendan Eich qui travaille chez Netscape Communication Corporation principal concurrent à l'époque d'Internet Explorer. JavaScript est alors transmis à l'ECMA (European Computer Manufacturers Association) pour standardisation sous la dénomination d'*ECMAScript* abrégé en ES. Les deux versions 5, sortie en 2009, et 6, sortie en 2014, sont implémentées dans les différents navigateurs présents sur le marché. Les versions suivantes 7, 8 et 9 sont des mises-à-jour "mineures" de l'ES6 et les fonctionnalités ajoutées ne sont pas supportées par tous les navigateurs.

JavaScript dont le nom est une référence à Java de Sun Microsystems, ne doit pas être confondu avec Java. En effet, Java est fortement typé et précompilé alors que JavaScript est à typage dynamique et interprété. JavaScript est un langage orienté objet.

## 2. Programmation

### 2.1. Quelques éléments du langage

Pour tester un code JavaScript en ligne, on peut utiliser le site <https://jsfiddle.net>

Comme tout langage de script présent dans une page HTML, les instructions de JavaScript doivent se trouver entre balises.

```
1 <!DOCTYPE HTML>
2 <html>
3 <body>
4   <script>
5     alert("Hello world!")
6   </script>
7 </body>
8 </html>
```

En cherchant des scripts JavaScript sur Internet, on pourra trouver le code suivant :

```
1 <script type="text/javascript">
2   /* bloc d'instructions */
3 </script>
```

En HTML5, il n'est pas utile d'indiquer le type/MIME du script : c'est d'ailleurs considéré comme redondant. Considérons l'exemple suivant, un peu plus riche.

```
1 <!DOCTYPE HTML>
2 <html>
3   <body>
4     <ol id="liste">
5       <li id="item1" onclick="changetexte(this.id)">Un</li>
6       <li id="item2" onclick="changetexte(this.id)">Deux</li>
7     </ol>
8     <script>
9       function changetexte(ident) {
10        var obj = document.getElementById(ident);
11        alert(obj.firstChild.nodeValue)
12      }
13    </script>
14  </body>
15 </html>
```

Dans un navigateur, cette page html a l'aspect suivant :

1. Un
2. Deux

Un clic sur l'un des deux items provoque l'apparition d'une alerte avec le texte associé.

Deux éléments ont été mis en place pour obtenir ce comportement : une fonction définie dans le script et un appel à cette fonction au niveau des éléments de la liste.

`onclick` est un gestionnaire d'événements : il permet de déclencher une action quand un événement, ici un clic gauche sur l'élément, survient.

Dans la fonction `changetexte` de paramètre `ident`, on commence par définir la variable `obj` et on lui affecte l'objet sur lequel on a cliqué. Le mot clé `document` fait référence à la page HTML.

Lors de la création du rendu de la page HTML, le DOM (ou Document Object Model) est créé. Il représente la structure de la page HTML sous la forme d'un arbre ramifié. Chaque niveau est un noeud et le niveau hiérarchique le plus faible est le texte. Cela permet de comprendre la manière dont on accède, en JavaScript, au texte associé à chaque élément de la liste en utilisant la propriété en

lecture seule `firstChild` qui renvoie le premier noeud de l'objet puis la propriété `nodeValue` qui permet de lire ou d'écrire la valeur de ce premier noeud.

Lors de l'appel de la fonction, le paramètre transmis utilise le mot clé `this` qui permet de faire référence à l'élément à partir duquel l'appel a été réalisé.

La définition de la fonction permet d'observer deux éléments de la syntaxe de JavaScript :

- ▷ Lors d'une succession d'instructions, celles-ci se terminent par un `;`.
- ▷ Si celles-ci appartiennent au même bloc d'instructions, on les entoure d'accolades.

La partie de script peut être placée dans un fichier externe `fonction.js` et la page html modifiée de la manière suivante :

```
1 <!DOCTYPE HTML>
2 <html>
3   <body>
4     <ol id="liste">
5       <li id="item1" onclick="changetexte(this.id)">Un</li>
6       <li id="item2" onclick="changetexte(this.id)">Deux</li>
7     </ol>
8     <script src="fonction.js" />
9   </body>
10 </html>
```

## 2.2. Gestion d'événements

La manière de programmer précédente, quoique fonctionnelle, n'est pas préconisée. Il est préférable de séparer HTML de JavaScript. Les nouveaux codes sont alors

Code 1 – HTML

```
1 <!DOCTYPE HTML>
2 <html>
3   <body>
4     <ol id="liste">
5       <li id="item1">Un</li>
6       <li id="item2">Deux</li>
7     </ol>
8     <script src="fonction2.js" />
9   </body>
10 </html>
```

Code 2 – fonction2.js

```
1 var objets = document.querySelectorAll('li');
2 function change_texte(e) {
3   alert(e.target.firstChild.nodeValue)
4 }
5 for (var i = 0; i < objets.length; i++) {
6   objets[i].onclick = change_texte
7 }
```

Le code JavaScript commence par récupérer la liste de tous les items de type `<li>`. A chacun de ces objets, on associe le gestionnaire d'événement `onclick` qui réalisera l'appel à la fonction `change_texte`. Une nouveauté dans la définition de la fonction est le paramètre `e` qui représente l'événement lors de l'appel de la fonction. `e.target` permet de récupérer l'objet sur lequel le clic a été effectué.

Une troisième manière d'associer un gestionnaire d'événement à un élément est

Code 3 – fonction2.js

```
objets[i].addEventListener('click', change_texte)
```

`myElement.addEventListener` permet d'associer plusieurs fonctions à l'événement `'click'` alors que la propriété `myElement.onclick` ne retient que la dernière fonction affectée.

De plus, on peut utiliser `myElement.removeEventListener` pour retirer un objet de l'écouteur d'événements.

On peut citer quelques gestionnaires d'événements :

- ▷ `myElement.onfocus` et `myElement.onblur` qui déclenchent un événement lorsque l'objet est activé ou non grâce à l'utilisation de la touche de tabulation.
- ▷ `window.onkeypress`, `window.onkeydown` et `window.onkeyup` qui réagissent si on appuie et relâche une touche, si l'appui est maintenu et si la touche est relâché respectivement. Ces événements sont obligatoirement liés à la fenêtre du navigateur.
- ▷ `myElement.onmouseover` et `myElement.onmouseout` qui réagissent lors du survol de l'élément et de l'arrêt de celui-ci.

Les événements sont munis de propriétés : un événement `MouseEvent` possède des propriétés de position `MouseEvent.screenX`,

↳ `MouseEvent.screenY`, de bouton cliqué `MouseEvent.button`, etc.

### 3. Références

Le programme indique qu'il ne saurait être question de développer une expertise dans le langage JavaScript. Les sites suivants seront utiles pour la mise en œuvre de projets :

- ▷ La syntaxe de JavaScript : <https://developer.mozilla.org/fr/docs/Web/JavaScript>
- ▷ Les préconisations du W3C concernant les événements : <https://www.w3.org/TR/DOM-Level-3-Events/>
- ▷ Un tutoriel pour programmer JavaScript : <https://openclassrooms.com/fr/courses/1916641-dynamisez-vos-sites-web-avec-javascript/>