

Le web dynamique : JavaScript

Récupérer le fichier compressé `js.zip` sur notre site et le décompresser dans votre répertoire de travail pour avoir accès aux fichiers cités ci-dessous.

.....

Capacités attendues

- ✓ Identifier les événements que les fonctions associées aux composants graphiques peuvent traiter.
 - ✓ Analyser et modifier les méthodes exécutées lors d'un clic sur un bouton d'une page Web.
-

Événements

En informatique, un événement est (une représentation d') une action, en général réalisée par l'utilisateur.

Sur une page HTML, il peut s'agir, par exemple, d'un clic sur un bouton, ou sur un élément quelconque, du survol de la souris au-dessus d'un élément, du changement d'état d'un widget, de l'appui sur une touche du clavier, du chargement de la page, etc. (Liste complète : https://www.w3schools.com/jsref/dom_obj_event.asp)

Pour chacun des éléments d'une page HTML, il est possible de définir une réaction à un événement donné. Cette réaction est codée en JavaScript, qui permet (notamment) de modifier tout élément de la page HTML.

Les instructions JavaScript sont exécutées par le client (le navigateur), ce qui signifie que les modifications ont lieu sur l'appareil de l'utilisateur. Aucune donnée n'est transmise, ni demandée, au serveur (à ce stade).

Exemple minimaliste

→ Voir fichiers `exemple1.html`

Contenu du fichier HTML :

- la ligne 5 indique de charger le fichier `script.js` situé dans le même répertoire que le fichier HTML ;
- dans l'élément `h1` à la ligne 8, l'attribut `onclick` est renseigné pour exécuter la fonction `hello`.

```
1 <!DOCTYPE HTML>
2 <html lang="fr">
3   <head>
4     <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
5     <script src="script1.js"></script>
6   </head>
7   <body>
8     <h1 onclick="hello()">Titre</h1>
9   </body>
10 </html>
```

Contenu du fichier JavaScript :

- la ligne 2 déclenche l'affichage d'une fenêtre de dialogue (window représente l'onglet du navigateur) ;
- la ligne 3 est une instruction qui affiche des informations dans la console de débogage.

```
1 function hello() {
2   window.alert("Hello world!");
3   console.log("on a cliqué sur le titre");
4 }
```

Exemple 2 : modification d'un élément identifié

→ Voir fichiers exemple2.html

Corps du fichier HTML :

- l'attribut `id` permet d'identifier l'élément `h1` par le mot `letitre` ;
- l'attribut `onclick` indique la fonction à exécuter lors du clic sur le bouton.

```
1 <body>
2   <h1 id="letitre">Titre</h1>
3   <button onclick="bleu()">Cliquez ici !</button>
4 </body>
```

Contenu du fichier JavaScript :

- le nom `document` est une référence au fichier HTML ;
- la fonction (méthode) `getElementById` renvoie l'élément dont l'identifiant est passé en paramètre ; la variable `element` est alors une référence au titre `h1` identifié par le mot `"letitre"` ;
- `element.style.color` désigne la couleur du texte dans l'élément ;

```
1 function bleu() {
2   console.log("on a cliqué sur le bouton");
3   let element = document.getElementById("letitre");
4   element.style.color = "#0000F0";
5 }
```

Exemple 3 : modification d'un élément passé en argument

→ Voir fichiers exemple3.html

Corps du fichier HTML :

- l'attribut `onmouseover` indique la fonction à exécuter lorsque la souris survole l'élément ;
- l'attribut `onmouseout` indique la fonction à exécuter lorsque la souris ne survole plus l'élément ;
- le mot-clef `this` désigne l'élément concerné (`h1`), qui est passée en argument des fonctions `bleu` et `noir`.

```
1 <body>
2   <h1 onmouseover="bleu(this)" onmouseout="noir(this)">Titre</h1>
3 </body>
```

Contenu du fichier JavaScript :

- le paramètre `element` de chacune des fonctions `bleu` et `noir` a pour valeur l'élément passé en argument ;
- `element.innerHTML` désigne le contenu de la balise, qui est ici modifié ;
- `element.style.fontSize` désigne la taille du texte dans l'élément.

```
1 function bleu(element) {
2   element.innerHTML = "Titre survolé";
3   element.style.color = "#0000F0";
4   element.style.fontSize = "48px";
5 }
6 function noir(element) {
7   element.innerHTML = "Nouveau titre";
8   element.style.color = "#000000";
9   element.style.fontSize = "32px";
10 }
```

Exemple 4 : utilisation d'une variable

→ Voir fichiers `exemple4.html`

Corps du fichier HTML : un clic sur le bouton appelle la fonction `changeInfo`.

```
1 <body>
2   <h1>Compteur</h1>
3   <button onclick="changeInfo()">Cliquez ici !</button>
4   <p id="info">Vous n'avez pas encore cliqué sur le bouton.</p>
5 </body>
```

Contenu du fichier JavaScript :

```
1 let n = 0;
2
3 function changeInfo() {
4   console.log("La valeur de n est : " + n);
5   n = n+1;
6   console.log("La nouvelle valeur de n est : " + n);
7   let element = document.getElementById("info");
8   element.innerHTML = "Vous avez cliqué " + n + " fois sur le bouton.";
9 }
```

Syntaxe en JavaScript

1 <i>// Alternative</i>	10 <i>// Boucle bornée</i>
2 if (condition) {	11 for (let idx = 0; idx < 10; idx++) {
3 instructions;	12 instructions;
4 }	13 }
5	14
6 <i>// Boucle non bornée</i>	15 lst = [10, 20, 30]
7 while (condition) {	16 for (let el of lst) {
8 instructions;	17 instructions;
9 }	18 }

- Les conditions doivent être parenthésées.
- Chaque instruction doit se terminer par un point-virgule.
- Un bloc d'instructions doit être placé entre accolades {}.
- Les opérateurs logiques sont : && (et) || (ou), ! (non).
- Le renvoi de valeur dans une fonction se fait avec le mot-clef `return`.