

II Événements

II.1 Sélectionner et modifier un élément HTML

Dans cette partie, nous utiliserons la page Web suivante :

```

0 <!doctype html>
1 <html lang="fr">
2   <head>
3     <meta charset="utf-8">
4     <link rel="stylesheet" href="style.css">
5   </head>
6   <body>
7     <h1>Page Test</h1>
8     <p>Quelques liens</p>
9     <ul>
10      <li><a href="https://fr.wikipedia.org">Premier lien</a></li>
11      <li><a href="https://www.mozilla.org" id="moz">Deuxième lien</a></li>
12      <li><a href="https://www.example.com">Troisième lien</a></li>
13    </ul>
14  </body>
15  <script src="script.js"></script>
16 </html>

```

ex1_index.HTML

Page Test

Quelques liens

- [Premier lien](#)
- [Deuxième lien](#)
- [Troisième lien](#)

Aspect initial de la page

Pour sélectionner un élément on peut utiliser un sélecteur CSS avec la fonction `document.querySelector`, s'il y a plusieurs éléments correspondant au sélecteur, c'est le premier qui est sélectionné :

```

0 let lien = document.querySelector("a");
1 lien.href = "https://lapiecode.info/";
2 lien.innerHTML = "Lien modifié";

```

(1)script.js

Page Test

Quelques liens

- [Lien modifié](#)
- [Deuxième lien](#)
- [Troisième lien](#)

On a modifié l'attribut `href` du premier lien, on peut faire de même avec les autres attributs des éléments HTML. Pour modifier, le contenu de l'élément, on utilise la propriété `innerHTML`.

La fonction `document.querySelectorAll` renvoie une liste contenant tout les éléments correspondant au sélecteur.

```

0 let liens = document.querySelectorAll("a");
1 for(let i = 0; i < liens.length; i++) {
2   liens[i].href = "https://lapiecode.info/";
3   liens[i].innerHTML = "Le même lien";
4 }

```

(2)script.js

Page Test

Quelques liens

- [Le même lien](#)
- [Le même lien](#)
- [Le même lien](#)

La fonction `document.getElementById` renvoie l'élément qui a `id` correspondant :

```

0 let lien2 = document.getElementById("moz");
1 lien2.style.background = "gray";
2 lien2.style.color = "white";

```

(3)script.js

Page Test

Quelques liens

- [Premier lien](#)
- [Deuxième lien](#)
- [Troisième lien](#)

Il est possible de modifier les propriétés CSS d'un élément en passant par l'attribut **style**.

II.2 Événements

Les événements permettent de déclencher une fonction selon qu'une action s'est produite ou non. Par exemple, on peut faire apparaître une fenêtre **alert()** lorsque l'utilisateur survole un certain élément HTML.

II.3 Lien fonction/événement

Pour lier une fonction à un événement, il y a deux méthodes :

La première, où le lien est fait dans le HTML directement, qui est donc à éviter :

```

0 <p onmouseover="msg()">
1   Survoler
2 </p>

```

```

0 function msg() {
1   alert("ok");
2 }

```

La deuxième à privilégier :

```

0 <p>
1   Survoler
2 </p>

```

```

0 function msg() {
1   alert("ok");
2 }
3
4 let texte = document.querySelector("p");
5 bouton.addEventListener('mouseover', msg);

```

Dans les deux cas, lorsqu'on passe la souris au dessus du paragraphe de texte un pop-up affiche ok.

II.4 Exemples d'événements

Il existe de nombreux événements, tous plus ou moins utiles. Nous allons donner quelques événements liés à la souris :

Type de l'événement	Attribut correspondant	A quoi il correspond
click	onclick	Cliquer (appuyer puis relâcher) sur l'élément
mouseover	onmouseover	Faire entrer le curseur sur l'élément
mouseout	onmouseout	Faire sortir le curseur de l'élément
mousemove	onmousemove	Faire déplacer le curseur sur l'élément

II.5 L'objet Event

Lorsqu'une fonction est liée à un événement, il est possible de lui passer en paramètre, un objet Event permettant différentes interactions.

```

0 <button id="clickme">
1   Cliquez-moi !
2 </button>

```

```

0 function chgTexte(ev) {
1   ev.target.innerHTML = "A cliqué !";
2 }
3
4 let bouton = document.getElementById('clickme');
5 bouton.addEventListener('click', chgTexte);

```

Par exemple, `<evenement>.target` permet de récupérer l'élément ciblé par l'événement.