

A. Comprendre les événements

Tout le code JavaScript que nous avons vu dans les chapitres précédents ne s'exécute qu'une seule fois, au chargement de la page. C'est donc une utilisation extrêmement limitée, il n'y a aucune interaction avec l'utilisateur de la page. Un des objectifs du JavaScript est de dynamiser la page, de proposer des interactions avec les utilisateurs.

Or, comment interagir avec ces utilisateurs ? Eh bien, par l'intermédiaire d'événements. Les événements se produisent à tout moment dans une page, par exemple lorsque le pointeur de la souris se déplace sur la page, quand il passe au-dessus d'un élément et quand il en sort, quand l'utilisateur clique sur un bouton. Mais vous avez aussi des événements qui sont déclenchés avec le clavier, comme appuyer sur une touche ou relâcher cette touche du clavier. Donc, il y a en permanence des événements qui se produisent dans une page web.

Aussi, pour nous, la seule question qu'il faut se poser est : à quels événements nous devons réagir et de quelle manière ? Pour réagir à un événement, il faudra être à son écoute, il faudra gérer un écouteur spécifique à cet événement.

B. Utiliser un bouton déclencheur d'action

1. Réagir au clic sur un bouton

Dans cet exemple très simple, nous allons réagir au clic de l'utilisateur sur un bouton de la page.

Voici la structure de la page et celle du bouton :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ma page</title>
  </head>
  <body>
    <p>Integer posuere erat a ante venenatis...</p>
    <button id="monBouton">Cliquez-moi</button>
  </body>
</html>
```

Le bouton possède l'identifiant `id="monBouton"`.

Voilà le code JavaScript utilisé dans cette page :

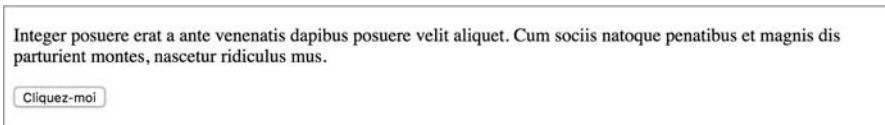
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ma page</title>
```

```
</head>
<body>
  <p>Integer posuere erat a ante venenatis...</p>
  <button id="monBouton">Cliquez-moi</button>
  <script>
    var leBouton = document.getElementById("monBouton") ;
    leBouton.onclick = function(){
      alert('Vous avez cliqué sur le bouton.') ;
    } ;
  </script>
</body>
</html>
```

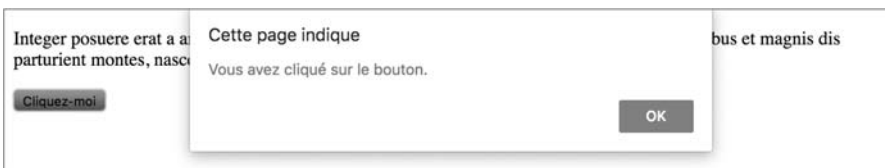
Détaillons ce script simple :

- Dans la variable `leBouton`, nous identifions le bouton.
- Ensuite, nous associons à ce bouton l'événement `onclick`. Cela permet de savoir si l'utilisateur clique sur ce bouton.
- Si l'utilisateur clique sur le bouton, nous exécutons une fonction anonyme : `function(){...}`. Cette fonction est dite anonyme, car elle ne possède pas de nom. Ce n'est pas nécessaire, car cette fonction n'est appelée qu'une seule fois, qu'à un seul moment, lorsque l'utilisateur clique sur le bouton. Cette fonction n'a pas besoin d'être appelée ailleurs.
- Cette fonction affiche un simple message dans un `alert()`.

Voici la page initiale :



Voici l'affichage obtenu lorsque l'utilisateur clique sur le bouton :



Nous avons bien l'affichage du message d'alerte lorsque l'utilisateur clique sur le bouton.

L'exemple à télécharger est dans le dossier **Chapitre-04-A/exemple-01.html**.

2. Utiliser un écouteur d'événement

Dans ce deuxième exemple, nous allons utiliser la fonction JavaScript écouteur d'événements : `addEventListener()`, <https://developer.mozilla.org/fr/docs/Web/API/EventTarget/addEventListener>

Nous allons utiliser la même structure de page que dans l'exemple précédent :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ma page</title>
  </head>
  <body>
    <p>Integer posuere erat a ante venenatis...</p>
    <button id="monBouton">Cliquez-moi</button>
  </body>
</html>
```

Voilà le code JavaScript utilisé dans cet exemple :

```
<script>
  var leBouton = document.getElementById("monBouton") ;
  leBouton.addEventListener("click", monAlerte, false) ;
  function monAlerte() {
    alert('Vous avez cliqué sur le bouton.') ;
  } ;
</script>
```

Analysons ce code :

- Nous retrouvons la même variable pour identifier le bouton.
- Sur ce bouton, nous utilisons la méthode `addEventListener()`.
- Le premier argument est l'événement que nous souhaitons écouter. Cet argument est une chaîne de caractères, il est donc placé entre guillemets. C'est l'événement "click", c'est le fait que l'utilisateur clique sur un élément identifié. Voici l'URL des événements que nous pouvons écouter : <https://developer.mozilla.org/fr/docs/Web/Events>
- Le deuxième argument est le nom de la fonction qui doit être appelée lorsque cet événement est déclenché : `monAlerte`.
- Le troisième argument concerne des options, nous n'en avons pas besoin, donc nous indiquons `false`.
- Ensuite, nous avons la définition de la fonction `monAlerte()` qui doit être exécutée.

Les affichages sont strictement identiques aux précédents.

Cette deuxième solution est nettement plus propre et plus technique. Elle distingue bien tous les paramètres liés à l'écoute d'un événement.

L'exemple à télécharger est dans le dossier **Chapitre-04-A/exemple-02.html**.

C. Utiliser d'autres événements

1. Les événements à écouter

Nous avons de nombreux événements à écouter à notre disposition :

<https://developer.mozilla.org/fr/docs/Web/Events>

Voici ceux liés au pointeur de la souris :

<https://developer.mozilla.org/fr/docs/Web/API/MouseEvent>

Nous n'allons bien sûr pas les passer tous en revue, ils sont simples à mettre en œuvre.

2. Le pointeur "au-dessus"

Le premier événement que nous allons aborder, c'est le simple fait de passer le pointeur de la souris au-dessus d'un élément identifié. C'est l'événement `mouseover`.

Voici l'élément ciblé, un simple paragraphe ayant l'identifiant `para` :

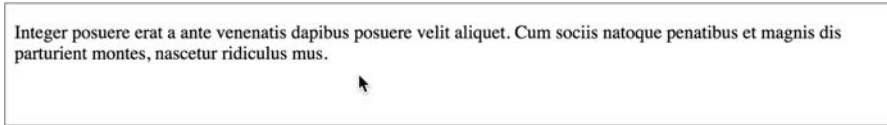
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Ma page</title>
  </head>
  <body>
    <p id="para">Integer posuere erat a ante venenatis...</p>
  </body>
</html>
```

Et voici le code JavaScript utilisé :

```
<script>
  var monElement = document.getElementById("para") ;
  monElement.addEventListener("mouseover",monAlerte,false) ;
  function monAlerte(){
    alert('Événement détecté.') ;
  } ;
</script>
```

Nous avons le même code que précédemment. Le seul élément qui change est l'événement écouté, `mouseover` dans cet exemple.

Voici l'affichage initial, avec la souris qui n'est pas au-dessus du paragraphe :



Voici l'affichage obtenu lorsque le pointeur de la souris passe au-dessus du paragraphe `<p id="para">`:



L'événement `mouseout` est déclenché lorsque le pointeur sort de la zone de l'élément identifié. L'exemple à télécharger est dans le dossier **Chapitre-04-C/exemple-01.html**.

3. Événement lié aux formulaires

Les formulaires ont aussi leurs événements. Nous allons dans cet exemple utiliser l'événement `onblur`, qui est déclenché lorsque le point d'insertion quitte un champ.

Voici la structure du formulaire que nous allons utiliser :

```
<form id="inscription" name="formulaire" method="post" action="#">
  <p>
    <label for="nom">Nom : </label>
    <input id="nom" name="nom" type="text">
  </p>
  <p>
    <label for="prenom">Prénom : </label>
    <input id="prenom" name="prenom" type="text">
  </p>
  <p>
    <label for="motDePasse">Mot de passe : </label>
    <input id="motDePasse" name="motDePasse" type="password">
  </p>
  <p>
    <input type="submit" name="envoyer" id="envoyer" value="Envoyer">
  </p>
</form>
```

C'est le champ du mot de passe qui nous intéresse, il est identifié par `id="motDePasse"`. L'objectif est d'afficher un message d'alerte si ce champ contient moins de 8 caractères.

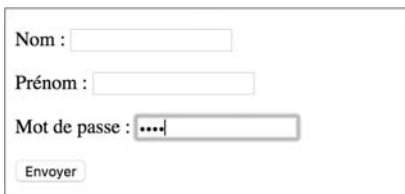
Voici le code utilisé :

```
<script>
  var champMDP = document.getElementById("motDePasse") ;
  champMDP.onblur = function () {
    var nbCar = champMDP.value.length ;
    if ( nbCar < 8 ) {
      alert ("Le mot de passe doit contenir au moins 8 caractères.")
    }
  };
</script>
```

Analysons ce code :

- Nous créons une variable nommée `champMDP` pour identifier le champ du mot de passe.
- Lorsque le point d'insertion quitte, `onblur`, ce champ `champMDP`, nous exécutons une fonction anonyme.
- Dans cette fonction, nous créons une variable `nbCar` qui compte le nombre de caractères saisis dans le champ du mot de passe : `champMDP.value.length`.
- Ensuite, nous testons si ce nombre est inférieur à 8 : `if (nbCar < 8)`.
- Si c'est le cas, nous affichons une alerte.

Voici le champ du mot de passe avec moins de 8 caractères saisis :



Nom :

Prénom :

Mot de passe :

Envoyer

Si nous passons au champ suivant, le message d'alerte est affiché :



Nom :

Prénom :

Mot de passe :

Envoyer

Cette page indique

Le mot de passe doit contenir au moins 8 caractères.

OK

Les formulaires permettent d'utiliser d'autres événements :

- `onfocus` : lorsque le pointeur rentre dans un champ.
- `focus` : lorsqu'un champ possède le focus, c'est-à-dire qu'il est actif.
- `focusin` : lorsqu'un champ va recevoir le focus.
- `focusout` : lorsqu'un champ va perdre le focus.