

Les Évènements d'une page web

Définition

Dans le cas du web, les événements sont des actions qui se produisent à l'intérieur de la fenêtre du navigateur et tendent à être rattachés à un élément spécifique qui s'y trouve.

Ils possèdent deux caractéristiques essentielles :

- On peut les « écouter »: on peut les détecter car le système va nous informer qu'elle se produit ;
- On peut leur « répondre »: on va pouvoir attacher un code à cette action qui va s'exécuter dès qu'elle va se produire.

(On détecte un évènement (action ou un état), afin de produire une nouvelle action)

Définir des gestionnaires d'évènements

Aujourd'hui, en JavaScript, il existe trois grandes façons d'implanter un gestionnaire d'évènements :

- On peut utiliser des attributs HTML de type évènement (non recommandé) ;
- On peut utiliser des propriétés JavaScript liées aux évènements ;
- On peut utiliser la méthode `addEventListener()` (recommandé).

Utiliser les attributs HTML

C'est la méthode la plus ancienne, cette façon de faire ne devrait plus être utilisée aujourd'hui. Cependant, de nombreux sites utilisent encore ce type de syntaxe ce qui nous force à l'étudier

Voici 3 exemples d'action qui peuvent être utilisés directement dans le code HTML :

- onclick : l'action se déroulera lorsque la cible sera cliquée.
 - ⇒ Ici, une boîte d'alerte affichant « Bouton Cliqué »
- onmouseover : lorsque la cible sera survolée
 - ⇒ Ici, la couleur d'arrière plan de la cible sera orange
- onmouseout : lorsque la cible ne sera pas survolée
 - ⇒ ici, la couleur d'arrière plan de la cible sera blanche

```
<button onclick="alert('Bouton cliqué')">Cliquez moi !</button>  
<div onmouseover="this.style.backgroundColor='orange'"  
      onmouseout="this.style.backgroundColor='white'">
```

Les propriétés JavaScript

- Pour commencer il faut sélectionner la cible sur laquelle on veut effectuer l'action:

```
//On sélectionne le premier button et le premier div du document  
let b1 = document.querySelector('button');  
let d1 = document.querySelector('div');
```

- Ensuite y relie l'action qui entrainera la modification
- Et on définit l'action à effectuer (elle peut être appliqué sur la même cible ou bien une autre)

```
//On utilise les propriétés gestionnaires d'évènement avec nos éléments  
b1.onclick = function(){alert('Bouton cliqué')};  
d1.onmouseover = function(){this.style.backgroundColor = 'orange'};  
d1.onmouseout = function(){this.style.backgroundColor='white'};
```

La méthode addEventListener()

Cette dernière façon de gérer les évènements est la manière recommandée aujourd'hui car c'est la plus flexible et la plus performante.

On va passer deux arguments à cette méthode :

- le nom d'un évènement qu'on souhaite prendre en charge
- le code à exécuter (qui prendra souvent la forme d'une fonction) en cas de déclenchement de cet évènement.

Cette méthode permet :

- de réagir plusieurs fois et de façon différente à un même évènement
- de réagir à différents évènements à partir de différents ou d'un même objet Element.

```
b1.addEventListener('click', function(){alert('Bouton cliqué')}});  
d1.addEventListener('mouseover', function(){this.style.backgroundColor='orange'});  
d1.addEventListener('mouseover', function(){this.style.fontWeight='bold'});  
d1.addEventListener('mouseout', function(){this.style.backgroundColor='white'});
```

Quel type d'évènement peut-on détecter?

Le chargement du document:

- Définir le moment d'activation d'une fonction au moment de son chargement

Un clic sur un bouton effectué par un utilisateur:

- Les données d'un formulaire peuvent être envoyées à un serveur

Le survol (ou non) d'un élément par la souris d'un utilisateur:

- Une ombre peut apparaitre sur un lien lorsque la souris passe dessus afin de le repérer

La sélection d'un champs de formulaire:

- Le fond d'un champs peut changer de couleur lorsque le curseur est dedans

Quelles actions peut-on réaliser

- On peut modifier, enlever, rajouter du texte dans un élément ou même un nœud dans la page HTML
- On peut modifier le style d'un élément (sa couleur, sa taille, sa position, le cacher et le montrer, etc...
- On peut afficher une alerte ou une nouvelle fenêtre
- On peut effectuer des opérations de toutes sorte et en renvoyer le résultat
- Et bien d'autres choses encore...