# Evènements JS

## Définition :

Les événements sont des actions ou des occurrences qui se produisent dans le système que vous programmez et dont le système vous informe afin que vous puissiez y répondre d'une manière ou d'une autre si vous le souhaitez.

Dans le cas du Web, les événements sont déclenchés à l'intérieur de la fenêtre du navigateur et tendent à être rattachés à un élément spécifique qui s'y trouve

En JavaScript, un évènement est une action qui se produit et qui possède deux caractéristiques essentielles :

* C’est une action qu’on peut « écouter », c’est-à-dire une action qu’on peut détecter car le système va nous informer qu’elle se produit ;
* C’est une action à laquelle on peut « répondre », c’est-à-dire qu’on va pouvoir attacher un code à cette action qui va s’exécuter dès qu’elle va se produire.

Par exemple, on va pouvoir détecter le clic d’un utilisateur sur un bouton d’un document et afficher une boite de dialogue ou un texte suite à ce clic. On parlera donc « d’évènement clic ».

### Exemple :

* L'utilisateur clique avec la souris sur un certain élément ou en place le curseur sur un certain élément.
* L'utilisateur appuie sur une touche du clavier.
* L'utilisateur redimensionne ou ferme la fenêtre du navigateur.
* Une page web finissant de se charger.
* Un formulaire en cours de soumission
* Une vidéo en cours de lecture, en pause ou en fin de lecture.
* Une erreur qui survient.

## Définir des gestionnaires d’évènements

Aujourd’hui, en JavaScript, il existe trois grandes façons d’implémenter un gestionnaire d’évènements :

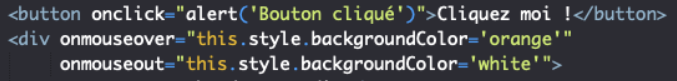
* On peut utiliser des attributs HTML de type évènement (non recommandé) ;
* On peut utiliser des propriétés JavaScript liées aux évènements ;
* On peut utiliser la méthode addEventListener() (recommandé).

## 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 utiliser directement dans le code HTML :

* onclick : l’action se déroulera lorsque la cible sera clickée.
  + 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



## Les propriétés JavaScript

Pour commencer il faut sélectionner la cible sur laquelle on veut effectuer l’action



## 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.

## Quelques Exemples d’évènements

Il existe de nombreux évènements répertoriés en JavaScript (plus d’une centaine). Les liés au Web peuvent être très différents les uns des autres :

Le chargement du document;

Un clic sur un bouton effectué par un utilisateur;

Le survol (ou non) d’un élément par la souris d’un utilisateur;

La sélection d’un élément de formulaire

Etc.

Les évènements peuvent représenter toute forme d’action:

On peut modifier, enlever, rajouter du texte dans un élément ou même un noeud 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 operations de toutes sorte et en renvoyer le résultat

Et bien d’autres choses encore…