

Réagir aux actions de l'utilisateur (événements)

Objectif : écouter les actions de l'utilisateur (clic, survol, saisie clavier...), exécuter des fonctions quand un événement se produit, utiliser `addEventListener`, créer des pages plus interactives et dynamiques

Sommaire

1. C'est quoi un événement ?

2. Comment écouter un événement ?

3. Exemples de base

4. L'objet Event

5. Attacher plusieurs événements à un même élément

6. Exercice

7. Ressources

C'est quoi un événement ?

Un événement, c'est quelque chose que l'utilisateur fait sur la page :

Action de l'utilisateur	Événement
Il clique quelque part	click
Il passe la souris sur un élément	mouseover
Il appuie sur une touche du clavier	keydown
Il tape dans un champ de formulaire	input
Il soumet un formulaire	submit
<i>Etc...</i>	

Comment écouter un événement ?

On utilise la méthode **addEventListener()** :

```
let bouton = document.getElementById("monBouton");  
  
bouton.addEventListener("click", function () {  
  alert("Tu as cliqué !");  
});
```

Ici, on écoute le **clic** sur un bouton, et on réagit en affichant une alerte.

⚠ **Le nom de l'événement est une chaîne ("click", "keydown", etc.)**

Et le deuxième argument est une fonction qui sera appelée quand l'événement se produit.

Exemples de base

```
document.getElementById("demo").addEventListener("click", function () {  
  this.innerText = "Tu m'as cliqué !";  
});
```

```
document.getElementById("demo").addEventListener("mouseover", function () {  
  this.style.backgroundColor = "lightblue";  
});
```

```
document.addEventListener("keydown", function (e) {  
  console.log("Touche appuyée :", e.key);  
});
```

L'objet event : ce qu'il contient et pourquoi il est utile

Quand tu ajoutes un écouteur d'événement, tu peux récupérer l'**objet event** automatiquement dans la fonction.

```
document.addEventListener("keydown", function () {  
  console.log(event); // affiche plein d'infos !  
});
```

Propriété	Ce qu'elle donne
<code>event.key</code>	La touche appuyée ("a", "Enter", etc.)
<code>event.target</code>	L'élément HTML sur lequel l'action a eu lieu
<code>event.type</code>	Le type d'événement déclenché ("click", "keydown"...)
<code>event.clientX, event.clientY</code>	Position de la souris au moment du clic

L'objet event : ce qu'il contient et pourquoi il est utile

Exemple pratique avec event.key :

```
document.addEventListener("keydown", function (e) {  
  document.getElementById("sortie").innerText = `Tu as appuyé sur :  
  ${event.key}`;  
});
```

Attacher plusieurs événements à un même élément

Oui, c'est possible et courant !

```
let bouton = document.getElementById("demo");
bouton.addEventListener("click", function () {
  console.log("Clic !");
});
bouton.addEventListener("mouseover", function () {
  bouton.style.backgroundColor = "yellow";
});
```

Chaque `addEventListener()` est indépendant. Tu peux avoir :

- un click
- un mouseover
- un dblclick
- un mouseout

Certains événements ne fonctionnent que sur certains éléments

Tous les éléments HTML ne peuvent pas réagir à tous les événements.

Élément HTML	keydown fonctionne ?
document	✓ Oui (général)
<input> ou <textarea>	✓ Oui (utile pour les formulaires)
<div> ou <p>	✗ Non, sauf si focusé (rarement utile)

```
// Global : on écoute n'importe quelle touche tapée
document.addEventListener("keydown", function (e) {
  console.log("Touche tapée :", e.key);
});
```

Ressources

Événements JavaScript

https://developer.mozilla.org/fr/docs/Learn_web_development/Core/Scripting/Events

removeEventListener JavaScript

<https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/removeEventListener>

(Vidéo) Événements Js

<https://www.youtube.com/watch?v=XbQVrWYZiXw>