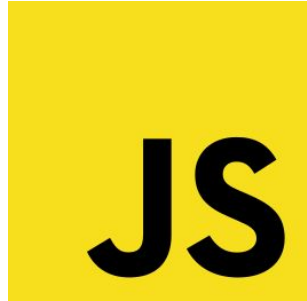


Javascript



## Objectifs

- Comprendre et manipuler les **événements** du DOM

## L'Event

Un événement est une **action qui se déclenche dans le navigateur et à laquelle il est possible de répondre.**

Par exemple, si un utilisateur clique sur un bouton, on pourrait vouloir réagir à **l'événement click** et décider d'afficher une boîte de dialogue.

**C'est un cas très courant.**

→ Il nous faudrait alors ajouter, sur le bouton, un écouteur d'événement **"click"**. Celui-ci pourrait alors exécuter un bloc de code **à chaque fois que l'événement "click" est déclenché.**

## Réagir à un événement : `addEventListener()`

- On utilise `element.addEventListener("événement à écouter", "fonction à déclencher")`
  - Un élément peut avoir **plusieurs** écouteurs d'événements.
  - [Liste des événements à écouter](#)

```
<button id="btn">Cliquer ici 🤖</button>
```

```
let btn = document.querySelector('#btn');

function display(event) {
  alert("Le bouton vient d'être cliqué");
  console.log(event);
}

btn.addEventListener('click', display);
```



- Depuis le `addEventListener`, on invoque la fonction sans ses parenthèses
- La fonction récupère automatiquement un argument, selon le type d'événement écouté

127.0.0.1:5500/index.html

Lorem ipsum dolor, sit amet consectetur ad  
quae harum possimus repellat a aperiam ex  
cumque inventore placeat repellendus elige  
excepturi? Iure?

Cacher

Cliquer ici 🧐

127.0.0.1:5500 indique

Le bouton vient d'être cliqué

OK

Elements

Console

Angular

Sources

2

Filter

Default levels

1 hidden

load enabled.

index.html:46

event-listener.js:5

```
▼ PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, pressure: 0, ...} ⓘ  
  isTrusted: true  
  altKey: false  
  altitudeAngle: 1.5707963267948966  
  azimuthAngle: 0  
  bubbles: true  
  button: 0  
  buttons: 0  
  cancelBubble: false  
  cancelable: true  
  clientX: 80  
  clientY: 124  
  composed: true  
  ctrlKey: false  
  currentTarget: null  
  defaultPrevented: false  
  detail: 1  
  eventPhase: 0  
  fromElement: null  
  height: 1  
  isPrimary: false  
  layerX: 80  
  layerY: 124  
  metaKey: false  
  movementX: 0  
  movementY: 0  
  offsetX: 7  
  offsetY: 17  
  pageX: 80  
  pageY: 124  
  path: (5) [button#btn, body, html, document, Window]  
  pointerId: 1  
  pointerType: "mouse"
```

```
event-listener.js
▼ PointerEvent {isTrusted: true, pointerId: 1, width: 1, height: 1, ...} ⓘ
  isTrusted: true
  altKey: false
  altitudeAngle: 1.5707963267948966
  azimuthAngle: 0
  bubbles: true
  button: 0
  buttons: 0
  cancelBubble: false
  cancelable: true
  clientX: 137
  clientY: 121
  composed: true
  ctrlKey: false
  currentTarget: null
  defaultPrevented: false
  detail: 1
  eventPhase: 0
  fromElement: null
  height: 1
  isPrimary: false
  layerX: 137
  layerY: 121
  metaKey: false
  movementX: 0
  movementY: 0
  offsetX: 64
  offsetY: 14
  pageX: 137
  pageY: 121
  ▶ path: (5) [button#btn, body, html, document, Window]
  pointerId: 1
```

- L'objet **"event"** passé automatiquement en argument de la fonction.
- Ici, comme l'événement écouté est **"click"**, c'est un objet de type **"PointerEvent"**

Pratiquons !

