

# JS : Manipuler des documents HTML

## L2 MPCIE - UE Développement Web

David Lesaint

david.lesaint@univ-angers.fr



Janvier 2019

# Les évènements

# Les évènements

## Sont

- Générés par l'utilisateur (eg. clic souris) ou une API (eg. fin d'une vidéo).
- Communiqués au code source lorsqu'ils surviennent.

## Plusieurs types d'évènements

- Standard (normalisés dans le DOM), non-standard, ou spécifique à un type de navigateur.
- Animation, batterie, appel, CSS, base de données, document, glisser-déposer, élément, focus, formulaire, frame, entrée (clavier/pad/souris), média (audio/vidéo), menu, réseau, notification, popup, impression, ressource, script, capteurs (compas/lampe/gyroscope/...), session, carte, SMS, SVG, table, texte, touché, mise à jour, champ, vue, websocket, fenêtre, ...

## Principes

- Les évènements sont matérialisés en JS par des objets prédéfinis de différents types.
- Chaque évènement a pour cible un objet tel un élément du DOM ou un objet d'API.
- Chaque évènement se propage en 2 phases successives - *capture* et *bouillonnement* - aux éléments situés sur la branche du DOM reliant l'objet cible à l'élément racine.
- Un gestionnaire d'évènements (event handler) est une fonction de rappel qui s'exécute lorsqu'un type d'évènements survient sur un objet donné.
- L'association/enregistrement d'un gestionnaire à l'une des phases de propagation d'un type d'évènements sur un objet donné se programme.

# Les interfaces d'évènements

Tout évènement est un objet implémentant l'interface `Event`

Fournit une information contextuelle au(x) gestionnaire(s) :

- `type` : son type (eg. `"click"`)
- `target` : son objet cible.
- `currentTarget` : l'objet courant sur lequel il est propagé.
- `bubbles` : s'il est en phase de bouillonnement ou de capture.
- ...

... Ou implémentant une sous-interface spécifique

E.g. `MouseEvent`

- Dérive de `UIEvent` qui dérive de `Event`.
- A des propriétés relatives à la position de la souris, etc.

# Les principaux évènements

Nom de l'évènement	Action pour le déclencher
<code>click</code>	Cliquer (appuyer puis relâcher) sur l'élément
<code>dblclick</code>	Double-cliquer sur l'élément
<code>mouseover</code>	Faire entrer le curseur sur l'élément
<code>mouseout</code>	Faire sortir le curseur de l'élément
<code>mousedown</code>	Appuyer (sans relâcher) sur le bouton gauche de la souris sur l'élément
<code>mouseup</code>	Relâcher le bouton gauche de la souris sur l'élément
<code>mousemove</code>	Faire déplacer le curseur sur l'élément
<code>keydown</code>	Appuyer (sans relâcher) sur une touche de clavier sur l'élément
<code>keyup</code>	Relâcher une touche de clavier sur l'élément
<code>keypress</code>	Frapper (appuyer puis relâcher) une touche de clavier sur l'élément
<code>focus</code>	« Cibler » l'élément
<code>blur</code>	Annuler le « ciblage » de l'élément
<code>change</code>	Changer la valeur d'un élément spécifique aux formulaires ( <code>input</code> , <code>checkbox</code> , etc.)
<code>input</code>	Taper un caractère dans un champ de texte ( <b>son support n'est pas complet sur tous les navigateurs</b> )
<code>select</code>	Sélectionner le contenu d'un champ de texte ( <code>input</code> , <code>textarea</code> , etc.)

<code>submit</code>	Envoyer le formulaire
<code>reset</code>	Réinitialiser le formulaire

# Enregistrement de gestionnaires (DOM-2)

Avec `addEventListener(type, listener[, options])`  
sur tout objet implémentant `EventTarget`

- `type` : type d'évènements (`string`).
- `listener` : fonction de rappel (ou objet implémentant `EventListener`) dont le 1er argument est l'évènement.
- `options` :
  - soit un booléen indiquant si l'écouteur sera notifié en phase de capture ou de bouillonnement.
  - soit un objet à propriétés booléennes
    - `capture` : notification en phase capture ou bouillonnement.
    - `once` : unique notification suivie d'un désenregistrement automatique de l'écouteur.
    - `passive` : interdiction pour l'écouteur de bloquer l'action par défaut de l'évènement.

# Gestionnaires

evenement-addeventlistener.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Enregistrement de gestionnaire d'évènements</title>
4 </head>
5 <body>
6   <span id="clickme">Cliquez-moi !</span>
7   <script>
8     var element = document.querySelector('#clickme');
9     element.addEventListener('click', function() {
10       alert("Vous m'avez cliqué !");
11     }, false);
12     element.addEventListener('click', function() {
13       alert("Je confirme : vous m'avez bien cliqué !");
14     }, false);
15   </script>
16 </body></html>
```



# Gestionnaires : accès à l'objet cible

evenement-addeventlistener-2.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Enregistrement de gestionnaire d'évènements</title>
4 </head>
5 <body>
6   <span id="clickme">Cliquez-moi !</span>
7   <script>
8     var element = document.querySelector('#clickme');
9     element.addEventListener('click', function(e) {
10       e.target.innerHTML = "Vous avez cliqué !";
11     }, false);
12   </script>
13 </body></html>
```

# Gestionnaires d'évènements souris

evenement-souris.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Evènements souris : mousemove</title>
4 </head>
5 <body>
6   <div id="position"></div>
7   <script>
8     var position = document.getElementById('position');
9     document.addEventListener('mousemove', function(e) {
10       position.innerHTML = 'Position X : ' + e.clientX
11       + 'px<br />Position Y : ' + e.clientY + 'px';
12     }, false);
13   </script>
14 </body></html>
```

# Gestionnaires d'événements clavier

evenement-clavier.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Evénements clavier : keypress, keyup, keydown</title>
4 <link rel="stylesheet" href="evenement-clavier.css"/>
5 </head><body>
6   <p><input id="field" type="text" /></p>
7   <table>
8     <tr><td>keydown</td><td id="down"></td></tr>
9     <tr><td>keypress</td><td id="press"></td></tr>
10    <tr><td>keyup</td><td id="up"></td></tr>
11  </table>
12  <script>
13    var field = document.getElementById('field'), down = document
14      .getElementById('down'), press = document
15      .getElementById('press'), up = document.getElementById('up');
16    document.addEventListener('keydown', function(e) {
17      down.innerHTML = e.keyCode; //code ASCII du caractère
18    }, false);
19    document.addEventListener('keypress', function(e) {
20      press.innerHTML = e.keyCode;
21    }, false);
22    document.addEventListener('keyup', function(e) {
23      up.innerHTML = e.keyCode;
24    }, false);
25  </script>
26 </body></html>
```

# Désenregistrement de gestionnaires (DOM-2)

Avec `removeEventListener(type, listener[, opti])`

En repassant exactement les paramètres passés à `addEventListener()`.

evenement-removeeventlistener.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8" />
3 <title>Désrengistrer un gestionnaire d'évènements</title>
4 </head>
5 <body>
6   <span id="clickme">Cliquez-moi !</span>
7   <script>
8     var element = document.querySelector('#clickme');
9     var fa1 = function() {alert("Vous m'avez cliqué !");}
10    var fa2 = function() {alert("Je confirme : Vous m'avez bien cliqué !");}
11    element.addEventListener('click', fa1, false);
12    element.addEventListener('click', fa2, false);
13    element.removeEventListener('click', fa1, false);
14  </script>
15 </body></html>
```





# Phases de capture et de bouillonnement

Cliquer sur `video` lance la lecture mais cache aussi le `div`.

evenement-propagation.html

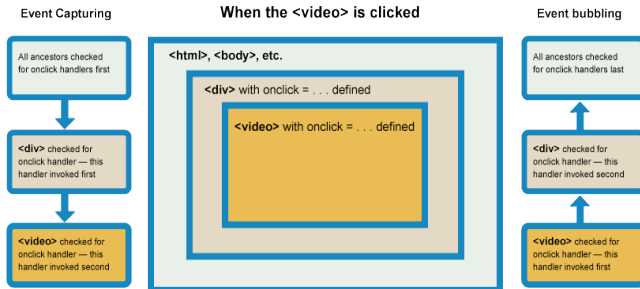
```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8">
3 <title>Capture et bouillonnement</title>
4 <link rel="stylesheet" href="evenement-propagation.css"/>
5 </head>
6 <body>
7   <button>Display video</button>
8   <div class="hidden">
9     <video><source src="rabbit320.mp4" type="video/mp4" /></video>
10  </div>
11  <script>
12    var btn = document.querySelector('button');
13    var div = document.querySelector('div');
14    var video = document.querySelector('video');
15    btn.addEventListener("click", function() {
16      div.setAttribute('class', 'showing');
17    }, false);
18    div.addEventListener("click", function() {
19      div.setAttribute('class', 'hidden');
20    }, false); // phase Bouillonnement
21    video.addEventListener("click", function() {
22      video.play();
23    }, false); // phase Bouillonnement
24  </script>
25 </body></html>
```

# Phases de capture et de bouillonnement

2 phase successives de propagation d'un évènement :

- 1 Capture : de la racine vers l'élément cible.
- 2 Bouillonnement : de l'élément cible vers la racine du DOM.

Chaque gestionnaire est à enregistrer pour l'une ou l'autre phase (bouillonnement par défaut).





# Contrôle de la propagation

## Avec `Event.stopPropagation()`

evenement-stoppropagation.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8">
3 <title>Bloquer la propagation d'évènements</title>
4 <link rel="stylesheet" href="evenement-propagation.css"/>
5 </head>
6 <body>
7   <button>Display video</button>
8   <div class="hidden">
9     <video><source src="rabbit320.mp4" type="video/mp4" /></video>
10  </div>
11  <script>
12    var btn = document.querySelector('button');
13    var div = document.querySelector('div');
14    var video = document.querySelector('video');
15    btn.addEventListener("click", function() {
16      div.setAttribute('class', 'showing');
17    }, false);
18    div.addEventListener("click", function() {
19      div.setAttribute('class', 'hidden');
20    }, false);
21    video.addEventListener("click", function(e) {
22      e.stopPropagation(); // blocage
23      video.play();
24    }, false); // phase Bouillonnement
25  </script>
26 </body></html>
```

# Délégation d'évènements basée sur le bouillonnement

## Créer un gestionnaire sur un parent plutôt que sur ses enfants

evenement-delegation.html

```
1 <!DOCTYPE html>
2 <html><head><meta charset="utf-8">
3 <title>Délégation d'évènements</title>
4 </head>
5 <body>
6   <ul id="parent-list">
7     <li id="post-1">Item 1</li>
8     <li id="post-2">Item 2</li>
9     <li id="post-3">Item 3</li>
10  </ul>
11  <script>
12    document.querySelector("#parent-list").addEventListener(
13      "click",
14      function(e) {
15        // !! balise en majuscules
16        if (e.target && e.target.nodeName == "LI") {
17          // e.target est l'item cliqué de la liste
18          alert("Item " + e.target.id.replace("post-", "") +
19            " a été cliqué!");
20        }
21      });
22  </script>
23 </body></html>
```