

RÉSUMÉ 5 - INTRODUCTION AUX ÉVÉNEMENTS

Lien vers l'activité : [Introduction aux événements](#)
www.infoforall.fr - Dernière modif. : 31 01 2021



1 - DEUX FAÇONS DE SURVEILLER L'ÉVÉNEMENT "CHARGEMENT"

1.1 - DANS LA BALISE HTML

On insère la surveillance directement dans les balises HTML via un attribut **onload** qui permet d'activer une fonction javascript lors du chargement de l'élément :

```
HTML | <body onload="demarrage()" >
```

1.2 - DANS LE CODE JAVASCRIPT

On déclenche une surveillance depuis Javascript en insérant un code de ce type :

```
JS | window.addEventListener("load",demarrage);
```

ATTENTION : dans les deux cas, il faudra disposer ici d'une fonction **demarrage**.

2 - SURVEILLER LES CLICS

Travaillons avec la version où on place l'attribut **onclick** dans la balise voulue.

Exemple 1 : on veut lancer **fonction_qu_on_veut_activer** en cliquant sur une image :

```
1 | 
```

Exemple 2 : On veut afficher un Pop-up en cliquant sur un paragraphe.

```
1 | <p onclick="alert('CLIC sur le paragraphe détecté !')">Clique sur moi !</p>
```

3 - OBTENIR LA RÉFÉRENCE DE LA BALISE HTML AVEC LE MOT-CLÉ THIS

3.1 PRINCIPE DE THIS

Le mot-clé **this** contient automatiquement la référence de la balise qui a déclenché l'événement. D'où le nom **this** pour dire : c'est CA le déclencheur.

Exemple : on veut obtenir l'adresse-source de l'image sur laquelle on vient de cliquer :

```
1 | 
```

3.2 RÉCUPÉRATION D'UN ATTRIBUT À L'AIDE DE THIS

Il suffit de donner la référence de la balise (this) suivi d'un point et du nom de l'attribut voulu.

- Récupération de la source : `this.src`
- Récupération de la largeur : `this.width`
- Récupération de la hauteur : `this.height`

Pour les propriétés liées au CSS, on peut utiliser l'attribut **style**.

- Récupération de la couleur d'écriture : `this.style.color`
- Récupération de la couleur du fond : `this.style.backgroundColor` ...

3.3 DIFFÉRENCE DE NOTATION CSS ET JS

En CSS, la propriété du fond coloré est `background-color`.

En JS, le tiret - n'est pas autorisé : on faut le remplacer par une majuscule : `backgroundColor`.

C'est vrai avec tous les noms composés d'un tiret.

On notera également que le CSS `className` ou encore le CSS `float` qui devient `styleFloat`.

3.4 MODIFIER LES ATTRIBUTS D'UNE BALISE

Il faut faire une simple affectation depuis le JS.

```
1 this.width = 200px;
2 this.height = 200px;
3 this.style.display = 'block';
4 this.style.display = 'inline';
5 this.style.display = 'inline-block';
6 this.style.color = 'red';
7 this.style.backgroundColor = 'yellow';
8 this.src = 'image2.png';
9 this.className = 'maClasseCSSQueJeVeux';
```

ATTENTION : Les exemples ont été écrits avec **this** mais il faut juste une variable qui contiennent la référence de la balise voulue. Si **reference** contient cette référence, on peut écrire :

```
1 reference.width = 200px;
```

4 - AUTRES ÉVÉNEMENTS

Attention, ces noms sont ceux qu'il faut indiquer lorsqu'on veut les surveiller depuis un déclenchement JS :

```
JS reference.p.addEventListener("mouseover", action a faire);
```

Si vous voulez lancer la surveillance depuis une balise, il faut rajouter on devant le nom de l'attribut.

```
HTML <p onmouseover="action a faire()" >
```

- click : on clique et on relâche sur la balise
- dblclick : pareil mais en clic double.
- mousedown : on clique gauche sans relâcher sur la balise.
- mouseup : on relâche gauche sur la balise.
- mouseover : on survole la balise.
- mousemove : on déplace la souris sur l'élément.
- mouseout : on fait sortir la souris de la balise.
- keydown : on appuie sans relâcher sur l'une des touches.
- keyup : on relâche une touche.
- keypress : on appuie et relâche une touche.
- focus : on vient de mettre le focus ou cibler l'élément, par exemple en cliquant sur l'élément.
- blur : on annule ce ciblage, par exemple en cliquant ailleurs.
- load : la balise est correctement chargée.

5 - INNERHTML

Le innerHTML correspond à ce qui est situé entre la balise d'ouverture et la balise de fermeture d'une balise. Les balises orphelines n'ont donc pas de innerHTML.

Cela peut être du texte (exemple 1 ci-dessous), mais cela peut également être tout un ensemble d'autres balises (exemple2).

```
1 <p>Voici un innerHTML composé d'un texte</p>
```

```
1 <ol>
2   <li>Premier élément</li>
3   <li>Deuxième élément</li>
4 </ol>
```

Exemple de lecture d'un innerHTML :

```
1 var texte = reference.innerHTML
2 var entier = parseInt(reference.innerHTML)
3 var flottant = parseFloat(reference.innerHTML)
```

Si vous désirez au contraire **redéfinir ce contenu interne**, il suffit d'utiliser ceci :

```
1 reference.innerHTML = "Je suis le <em>nouveau</em> contenu"
```

6 - EXEMPLE D'UTILISATION

Voir le site pour d'autres exemples.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8" />
5   <title>Gestion des autres événements</title>
6   <style>
7     .s {
8       display: inline-block;
9       text-align: center;
10      width: 20px;
11      height: 20px;
12      color: white;
13      background-color: #BB00BB;
14      margin: 5px;
15    }
16  </style>
17 </head>
18 <body>
19   <main>
20     <h1>Qui se cache derrière le nom mystère&nbsp;?</h1>
21     <p>
22       <span onclick="affiche(this,'L');" class="s">*</span>
23       <span onclick="affiche(this,'o');" class="s">*</span>
24       <span onclick="affiche(this,'v');" class="s">*</span>
25       <span onclick="affiche(this,'e');" class="s">*</span>
26       <span onclick="affiche(this,'l');" class="s">*</span>
27       <span onclick="affiche(this,'a');" class="s">*</span>
28       <span onclick="affiche(this,'c');" class="s">*</span>
29       <span onclick="affiche(this,'e');" class="s">*</span>
30     </p>
31   </main>
32   <script>
33     function affiche(reference, caractere) {
34       reference.innerHTML = caractere;
35     }
36   </script>
37 </body>
38 </html>
```

www.infoforall.fr

