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 versionoù 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  <img onclick="fonction_qu_on_veut_activer()"
2       src="https://doc.infoforall.fr/commun/images/accueil_HTML5_150.png">
```

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  <img width="100px"
2       onclick="alert(this.src);"
3       src="https://doc.infoforall.fr/commun/images/accueil_HTML5_150.png">
```

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 `class` devient le JS `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.

```
this.width = 200px;
this.height = 200px;
this.style.display = 'block';
this.style.display = 'inline';
this.style.display = 'inline-block';
this.style.color = 'red';
this.style.backgroundColor = 'yellow';
this.src = 'image2.png';
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 :

```
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).

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

```
<ol>
    <li>Premier élément</li>
    <li>Deuxième élément</li>
</ol>
```

Exemple de lecture d'un innerHTML :

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

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

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

6 - EXEMPLE D'UTILISATION

Voir le site pour d'autres exemples.

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8" />
    <title>Gestion des autres événements</title>
    <style>
        .s {
            display: inline-block;
            text-align: center;
            width: 20px;
            height: 20px;
            color: white;
            background-color: #BB00BB;
            margin: 5px;
        }
    </style>
</head>
<body>
    <main>
        <h1>Qui se cache derrière le nom mystère ?</h1>
        <p>
            <span onclick="affiche(this,'L');" class="s">*</span>
            <span onclick="affiche(this,'o');" class="s">*</span>
            <span onclick="affiche(this,'v');" class="s">*</span>
            <span onclick="affiche(this,'e');" class="s">*</span>
            <span onclick="affiche(this,'l');" class="s">*</span>
            <span onclick="affiche(this,'a');" class="s">*</span>
            <span onclick="affiche(this,'c');" class="s">*</span>
            <span onclick="affiche(this,'e');" class="s">*</span>
        .</p>
    </main>
    <script>
function affiche(reference, caractere) {
    reference.innerHTML = caractere;
}
    </script>
</body>
</html>
```