



Javascript



JAVASCRIPT

Les événements en JavaScript

La création d'un événement en JavaScript sur une page Web utilise les attributs HTML de type événement auxquels nous attachons le code JavaScript que nous désirons exécuter lors du déclenchement de l'événement associé.

JAVASCRIPT

Les attributs HTML de type événement

Il existe plusieurs attributs HTML de type événement utilisables à différents niveaux sur une page HTML.

Les attributs de niveau fenêtre

Près d'une dizaine d'attributs de niveau fenêtre peuvent être utilisés. Par exemple :

- ***onload*** : se déclenche après le téléchargement complet d'une page HTML.
- ***onresize*** : se déclenche lorsque le navigateur est redimensionné.
- ***onunload*** : se déclenche lorsque le navigateur est fermé.

Pour ne citer que ces quelques exemples.

Les attributs de niveau formulaire

Une dizaine d'attributs de niveau formulaire peuvent être utilisés. Par exemple :

- ***onfocus*** : se déclenche lorsqu'un élément reçoit le focus.
- ***onreset*** : se déclenche lorsque le bouton reset d'un formulaire est cliqué.
- ***onsubmit*** : se déclenche lorsqu'un formulaire est soumis.

JAVASCRIPT

Les attributs de niveau clavier

Il existe trois attributs de niveau clavier qui peuvent être utilisés. Ce sont :

- ***onkeydown*** : se déclenche lorsque l'utilisateur enfonce une touche de son clavier.
- ***onkeypress*** : se déclenche lorsque l'utilisateur enfonce une touche de son clavier. Cet événement n'est pas déclenché avec toutes les touches du clavier. Par exemple, les touches ***ctrl***, ***alt***, ***shift*** et ***esc*** ne sont pas récupérées par cet événement dans tous les navigateurs.
- ***onkeyup*** : se déclenche lorsque l'utilisateur relâche une touche enfoncée.

Les attributs de niveau souris

Il existe d'une dizaine d'attributs de niveau souris qui peuvent être utilisés. Par exemple :

- ***onclick*** : se déclenche lors d'un clic sur un élément.
- ***ondblclick*** : se déclenche lors d'un double-clic sur un élément.
- ***onmousedown*** : se déclenche lorsqu'un bouton de la souris est enfoncé sur un élément.
- ***onmouseup*** : se déclenche lorsqu'un bouton de la souris est relâché sur un élément.

JAVASCRIPT

```
<element onclick="myScript">
```

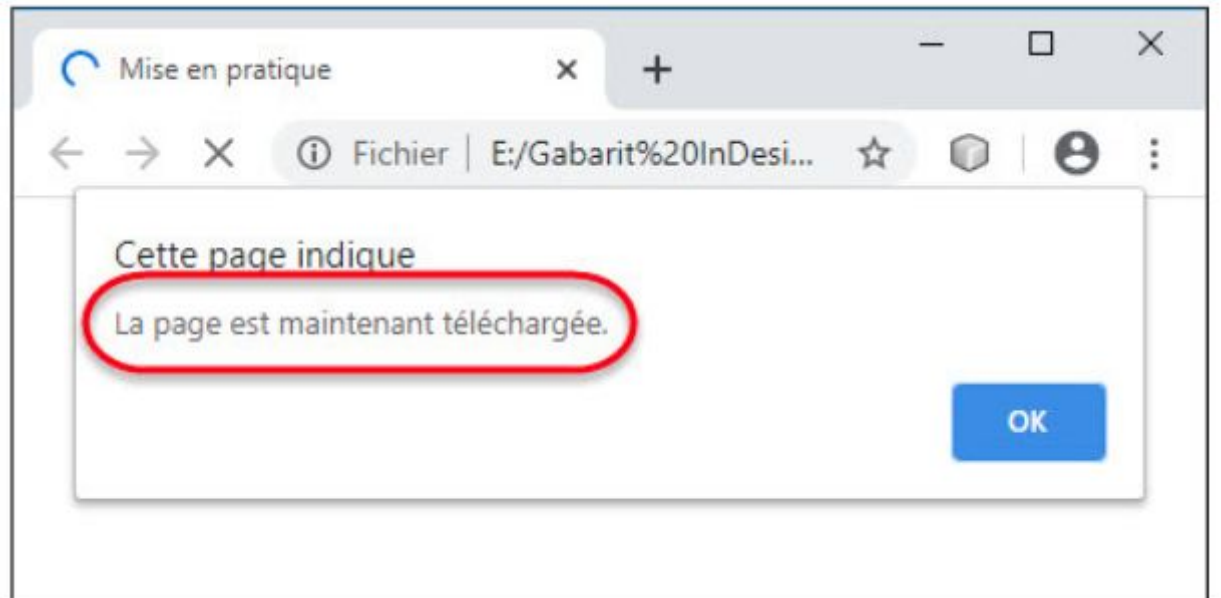
```
object.onclick = function() {myScript};
```

```
object.addEventListener("click", myScript);
```

JAVASCRIPT

```
<body onload="maFonction()">
  <h1 id="Titre">Le DOM HTML</h1>
  <p>Ceci est un premier paragraphe contenant
    <a href="http://monsite.com">un lien fictif</a>.</p>
  <p>Ceci est un deuxième paragraphe contenant
    <a href="http://monsite2.com">un autre lien</a>.</p>

  <script>
    function maFonction()
    {
      alert("La page est maintenant téléchargée.");
    }
  </script>
</body>
```



JAVASCRIPT

L'événement **onload** que nous venons d'utiliser sur notre page n'est pas seulement réservé qu'au chargement d'une page Web dans le navigateur. En effet, il est possible d'utiliser cet événement sur d'autres éléments HTML comme une image. Modifions le code de notre page afin d'effectuer le test. Examinez le prochain bloc de code.

```
<body>
  <h1 id="Titre">Le DOM HTML</h1>
  
  <p id="para"></p>

  <script>
    function maFonction()
    {
      document.getElementById("para").innerHTML =
        "L'image est maintenant téléchargé.";
    }
  </script>
</body>
```



JAVASCRIPT

Récapitulation

Il existe plusieurs attributs HTML de type événement utilisables à différents niveaux sur une page HTML.

Dans cette leçon, nous avons découvert l'existence de ces attributs HTML et nous avons étudié comment se déclenche un événement.

Un événement a besoin de deux choses pour être déclenché sur un élément HTML.

Nous devons déclarer notre événement au niveau de l'élément HTML sur lequel nous désirons déclencher notre événement et lui associer une fonction qui sera exécutée lors du son déclenchement.

JAVASCRIPT

Utiliser des événements courants en JavaScript

L'attribut onclick

```
<body>
  <h1 id=«Titre»>Les événements</h1>
  <p onclick="alert('Bravo !');">Cliquez-moi ! Cliquez-moi !</p>
</body>
```

JAVASCRIPT

Utiliser des événements courants en JavaScript

Il existe deux manières de réagir aux événements en JavaScript :

1. On peut utiliser des propriétés afin d'assigner un gestionnaire d'événement à un élément HTML;
2. On peut utiliser la méthode `addEventListener()`.

```
<body>
  <h1 id="Titre">Les événements</h1>
  <p id="para">Cliquez-moi ! Cliquez-moi !</p>
  <p>Un deuxième paragraphe.</p>

  <script>
    let p1 = document.getElementById("para");
    p1.onclick = function()
    {

    };
  </script>
</body>
```

```
<body>
  <h1 id="Titre">Les événements</h1>
  <p id="para" onclick="bravo()">Cliquez-moi ! Cliquez-moi !</p>
  <p>Un deuxième paragraphe.</p>

  <script>
    function bravo()
    {
      let p1 = document.getElementById("para");
      p1.textContent = "Bravo!";
      p1.style.fontSize = "25px";
      p1.style.color = "green";
      p1.style.fontWeight = "bold";
    }
  </script>
</body>
```

JAVASCRIPT

Utiliser des événements courants en JavaScript

Les attributs onmouseover et onmouseout

```
<body>
  <h1 id="Titre">Les événements</h1>
  

  <script>
    function surImage(img)
    {
      img.src = "Images/smiley-bye.gif";
    }

    function image(img)
    {
      img.src = "Images/bye-bye2.gif";
    }
  </script>
</body>
```

JAVASCRIPT

Utiliser des événements courants en JavaScript

Les attributs onkeydown et onkeyup

```
<body>
  <h1 id="Titre">Les événements</h1>
  <input type="text" id="demo" onkeydown="fonctionKeydown()"
    onkeyup="fonctionKeyup()">
  <p id="saisie" class="para2"></p>

  <script>
    function fonctionKeydown()
    {
      document.getElementById("demo").style.backgroundColor = "lightgreen";
    }
    function fonctionKeyup()
    {
      let texte = document.getElementById("demo");
      document.getElementById("saisie").innerHTML = texte.value;
    }
  </script>
</body>
```

Les événements



JAVASCRIPT

Récapitulation

Lorsque bien compris et utilisés correctement, les attributs HTML de type événement peuvent apporter beaucoup d'interaction entre l'utilisateur et vos site Web.

Dans cette leçon, nous avons découvert comment associer une fonction à un attribut HTML de type événement.

Nous avons aussi découvert que nous sommes en mesure d'obtenir le même résultat en utilisant différentes manières d'associer nos fonctions avec nos attributs HTML de type événement.

JAVASCRIPT

La méthode `addEventListener()`

```
<script>
  let para = document.querySelector("p");
  para.addEventListener("click", modifierTexte);

  function modifierTexte()
  {
    para.innerHTML = "<b>Bravo !</b>";
    para.style.fontSize = "25px";
    para.style.color = "green";
  }
</script>
```

Les événements

Bravo!

Un deuxième paragraphe.

JAVASCRIPT

Autre exemple

```
<body>
  <h1 id="Titre">Les événements</h1>
  

<script>
  let image = document.querySelector("img");

  image.addEventListener("mouseover", modifierImage);
  image.addEventListener("mouseout", imageInitiale);

  function modifierImage()
  {
    image.src = "Images/smiley-bye.gif";
  }

  function imageInitiale()
  {
    image.src = "Images/bye-bye2.gif";
  }
</script>
```

Évaluez vos connaissances nouvellement acquises

1. Les événements correspondent à des _____ effectuées soit par l'utilisateur, soit par le navigateur.
 - a. méthodes
 - b. actions
 - c. fonctions
 - d. intentions

2. Parmi les affirmations suivantes, laquelle est vraie concernant les attributs HTML de type événement ?
 - a. Le nom des attributs débutent par les lettres on.
 - b. Ce type d'attribut est associé à une action spécifique effectuée sur une page Web.
 - c. Ce type d'attribut exécute une série d'instructions lors de son déclenchement.
 - d. Toutes ces réponses

3. Quel est le premier paramètre que doit recevoir la méthode `addEventListener()` de JavaScript ?
 - a. Le nom de l'événement déclencheur.
 - b. La fonction qui doit être exécutée lors du déclenchement de l'événement.
 - c. Cette méthode ne reçoit aucune valeur en paramètre.
 - d. Les réponses a et b.

Évaluez vos connaissances nouvellement acquises

4. Examinez attentivement le prochain bloc de code.

```
<body>
  <h1 id="Titre">Les événements</h1>
  <input type="text" id="demo">
  <p id="saisie"></p>
</body>

<script>
  let saisi = document.getElementById("demo");
  let para = document.getElementById("saisie");

  saisi.addEventListener("keydown", modifierTexte());

  function modifierTexte()
  {
    para.innerHTML = saisi.value;
  }
</script>
```

Quel sera le résultat affiché lors d'une saisie dans la zone de texte ?

- a. Rien, l'événement `keydown` n'est pas un événement reconnu en JavaScript.
- b. Rien, la méthode `addEventListener()` reçoit une valeur incorrecte en paramètre.
- c. Le contenu textuel de notre élément `p` sera modifié à chaque fois que l'utilisateur saisira une lettre dans la zone de saisie de la fenêtre.
- d. Aucune de ces réponses

Évaluez vos connaissances nouvellement acquises

5. Examinez attentivement le prochain bloc de code.

```
<body>
  <h1 id="Titre">Les événements</h1>
  <p id="para">Cliquez-moi !</p>
  <p id="saisie"></p>
</body>

<script>
  let titre = document.getElementById("Titre");
  let para = document.getElementById("para");

  para.addEventListener("click", modifierTexte);

  function modifierTexte()
  {
    titre.innerHTML += " en JavaScript";
    titre.style.color = "red";
  }
</script>
```

Quel sera le résultat de l'exécution de ce script ?

- a. Rien, l'événement click n'est pas attaché sur un bon élément sur la page Web.
- b. Tout le contenu textuel de la page sera modifié.
- c. Le contenu textuel et la couleur du texte de l'élément h1 sera modifié.
- d. Le contenu textuel et la couleur du texte du deuxième paragraphe sera modifié.

Exercice sur les actions de formulaire

Vous allez créer un formulaire d'inscription avec les champs suivants :

- Nom (obligatoire, au moins 3 caractères)
- Prénom (obligatoire, au moins 3 caractères)
- Email (obligatoire, doit être une adresse email valide)
- Mot de passe (obligatoire, au moins 6 caractères)
- Confirmation du mot de passe (doit correspondre au mot de passe)
- Genre (choix entre "Homme" et "Femme")
- Bouton d'envoi ("S'inscrire")



Lorsque l'utilisateur soumet le formulaire :

- Vérifier que tous les champs obligatoires sont remplis.
- Vérifier que le champ email contient une adresse valide.
- Vérifier que le mot de passe fait au moins 6 caractères et que la confirmation correspond.
- Si une validation échoue, afficher un message d'erreur sous le champ correspondant.
- Si tout est correct, afficher un message de succès.

Exigences supplémentaires :

- Utiliser l'attribut `onsubmit` pour déclencher la validation en JavaScript.
- Empêcher la soumission du formulaire si une validation échoue.

Vérifier qu'une chaîne de caractère est un email

```
const emailRegex = /^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$/;  
emailRegex.test(chaine);
```