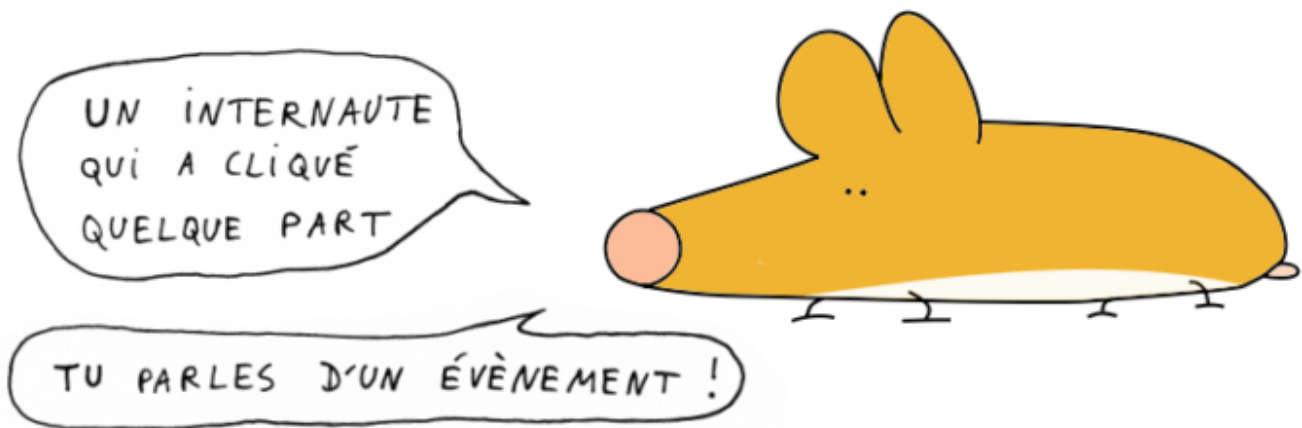


Programmation événementielle

Il est possible de programmer des réactions suite à des actions de l'internaute. Les actions des utilisateurs sont appelées *événement*. Voici quelques exemples :

- Clic sur un bouton.
- Touche du clavier enfoncée.
- Focus dans un champ de texte.
- Survol d'une image avec le pointeur.



La méthode `addEventListener()`

Syntaxe

Le principe est d'attacher un événement à une balise avec la méthode `addEventListener()` :

```
balise.addEventListener(événement, fonction);
```

La balise peut notamment être désignée par la fameuse méthode **`getElementById`**.

Évènement *click*

Dans le code ci-dessous, on capte tous les clics de la souris sur une image, et on déclenche une boîte d'alerte.

```
<!DOCTYPE html>
<html>
<head>
```

```

    <meta charset="utf-8">
</head>
<body>
  
  <script>
    document.getElementById("chapeau").addEventListener('click', function() {
      alert("clic !");
    });
  </script>
</body>
</html>

```

Ci-dessus, l'évènement click appelle une fonction anonyme qui affiche une simple boîte d'alerte. On peut également appeler une fonction par son nom :

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  
  <script>
    document.getElementById("chapeau").addEventListener('click', FaisUnTruc);
    function FaisUnTruc() {
      alert('clic !');
    }
  </script>
</body>
</html>

```

Attention à l'ordre des choses

En JavaScript, il ne faut surtout pas utiliser *document.getElementById()* avant que l'id de la balise ne soit lu par le navigateur qui, je le rappelle, lit le code de haut en bas. De ce fait, si vous souhaitez mettre le code JavaScript dans l'entête, il conviendra d'utiliser l'évènement *window.onload* pour attendre que toutes les balises soient lues avant d'y effectuer des opérations.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script>
    window.onload = function() {
      document.getElementById("chapeau").addEventListener('click', FaisUnTruc);
    }
    function FaisUnTruc() {
      alert('clic !');
    }
  </script>
</head>
<body>
  
</body>
</html>

```

Évènements *mouseover* et *mouseout*

Il est possible de gérer plusieurs évènements sur le même objet. Dans l'exemple ci-dessous, on guette le survol du pointeur sur une image et son éloignement :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  
  <p id="blabla"></p>
  <script>
    var chapeau = document.getElementById("chapeau");
    chapeau.addEventListener("click", function() { message('Vous avez cliqué'); });
    chapeau.addEventListener("mouseover", function() { message('Vous passez'); });
    chapeau.addEventListener("mouseout", function() { message('Vous repartez'); });
    function message(texte) {
      document.getElementById("blabla").innerHTML = texte;
    }
  </script>
</body>
</html>
```

Remarques :

- La ligne `var chapeau = document.getElementById("chapeau");` permet d'éviter les répétitions des `document.getElementById`.
- Si vous souhaitez appeler une fonction avec un paramètre, vous devez l'insérer dans une fonction anonyme `function() { ... }`

Profitons du JavaScript pour animer le survol d'une image. Voici l'exemple du rollover où le survol d'une image change sa source.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  
  <script>
    document.getElementById("photo").addEventListener("mouseover", function() {
ChangeImage("chien.jpeg"); });
    document.getElementById("photo").addEventListener("mouseout", function() {
ChangeImage("chat.jpeg"); });

    function ChangeImage(image) {
      document.getElementById("photo").src = image;
    }
  </script>
</body>
```

</html>

Tableau des évènements

En attachant des évènements à des balises, le JavaScript vous permettra de réaliser vos rêves les plus fous (enfin presque). Voici les évènements les plus courants. Notez que ce tableau est loin d'être exhaustif, car il existe une centaine d'évènements.

Évènements	Actions de l'utilisateur	Balises concernées
blur	Le focus quitte un champ de formulaire	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
change	Le contenu d'un champ de formulaire est modifié. Voir un exemple ici.	<input>, <select> et <textarea>
click	L'utilisateur clique sur l'objet	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
focus	Un champ de formulaire reçoit le focus	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
load	Le chargement du contenu de la balise est terminé	<body>, <frame>, <iframe>, , <input type="image">, <link>, <script>, <style>
mouseout	La souris quitte l'objet	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
mouseover	La souris arrive au-dessus de l'objet	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
reset	Annulation d'un formulaire (clic sur un bouton reset)	<form> et <keygen>
select	Sélection de texte	<input type="file">, <input type="password">, <input type="text">, <keygen> et <textarea>
submit	Envoi d'un formulaire si aucune valeur <i>false</i> n'est retournée. Voir un exemple ici.	<form> et <keygen>
unload	L'internaute quitte la page	<body>
keyup	Une touche du clavier est relâchée	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>
keydown	Une touche du clavier est enfoncée	Toutes les balises sauf : <base>, <bdo>, , <head>, <html>, <iframe>, <meta>, <param>, <script>, <style> et <title>