


Les événements

Les événements sont les déclencheurs de l'interactivité apportée par le JavaScript. Ceux-ci permettent d'associer des fonctions ou des méthodes aux actions de l'utilisateur. Avec la librairie jQuery, le principe reste le même, mais certains gestionnaires d'événements sont adaptés et de nouvelles méthodes introduites.

Passons en revue quelques gestionnaires d'événement :

<code>click</code>	Associe une fonction à l'événement clic de souris.
<code>focus</code>	Associe une fonction à l'événement focus.
<code>blur</code>	Associe une fonction à l'événement blur.
<code>scroll</code>	Associe une fonction à l'utilisation de la barre de défilement.
<code>mousedown</code>	Associe une fonction lorsque l'utilisateur presse la touche de la souris.
<code>mouseup</code>	Associe une fonction lorsque l'utilisateur relâche la touche de la souris.
<code>mousemove</code>	Associe une fonction lorsque le curseur de la souris survole un élément.
<code>mouseover</code>	Associe une fonction lorsque le curseur de la souris survole un élément.
<code>mouseout</code>	Associe une fonction lorsque le curseur de la souris quitte un élément.
<code>mouseenter</code>	Associe une fonction lorsque le curseur de la souris entre dans un élément mais sans propagation de l'événement.
<code>mouseleave</code>	Associe une fonction lorsque le curseur de la souris entre dans un élément mais sans propagation de l'événement.
<code>submit</code>	Associe une fonction à l'envoi d'un formulaire.
<code>change</code>	Associe une fonction à la modification d'un contrôle de formulaire.

 Globalement, jQuery reprend les événements du JavaScript sans le préfixe `on`. Ainsi le `onclick` du JavaScript traditionnel devient `click` en jQuery.

Exemple

Au clic sur un paragraphe, celui-ci disparaît.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>jQuery</title>
<meta charset="UTF-8">
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>
<script>
$(document).ready(function(){
$("p").click(function () {
$(this).slideUp();
});
});
</script>
<style>
p {cursor: pointer;}
</style>
</head>
<body>
```

```
<p>Paragraphe</p>
<p>Paragraphe</p>
<p>Paragraphe</p>
</body>
</html>
```



Après le clic sur le dernier paragraphe :



on(événement, [sélecteur], [données], fonction)

Affecte l'événement à un élément donné.

- **événement** (chaîne de caractères) : l'événement associé. Si plusieurs événements sont spécifiés, ils seront simplement séparés par un espace.
- **sélecteur** (optionnel) : un filtre pour sélectionner les enfants de l'élément sur lequel on applique l'événement.
- **données** (optionnel) : des données éventuellement fournies à la fonction.
- **fonction** : le code à exécuter au déclenchement de l'événement.

```
$("#button").on("click", function() {
    alert($(this).text());
});
```

La méthode `on()` est une alternative à l'écriture simplifiée de la gestion des événements que nous venons de découvrir. Elle permet une action plus évoluée par ses nombreux paramètres.

La méthode `off()` supprime les actions associées à un événement par la méthode `on()`.

Exemple

Ajoutons un arrière-plan de couleur au survol du paragraphe. Cet arrière-plan disparaît lorsque le curseur quitte celui-ci.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title>jQuery</title>
<meta charset="UTF-8">
<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script><script>
$(document).ready(function(){
$("p").on("mouseover mouseout", function(e){
$(this).toggleClass("over");
});
});
</script>
<style>
p { cursor: pointer;
padding-left: 20px;}
p.over { background: rgb(195,215,235);}
</style>
</head>
<body>
<h3>Paragraphe</h3>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed
non risus. Suspendisse lectus tortor, dignissim sit amet,
adipiscing nec, ultricies sed, dolor.</p>
</body>
</html>
```



Au survol du paragraphe :

