

## TP 3 : LES EVENEMENTS ET LES EFFETS

---

### 1. LES EVENEMENTS

Les événements sont les déclencheurs de l'interactivité apportée par le JavaScript. Ils permettent d'associer à des actions de l'utilisateur, des méthodes. Avec jQuery, le principe reste le même mais certains gestionnaires d'événements sont adaptés.

#### 1.1. LES GESTIONNAIRES D'ÉVÉNEMENT

##### 1.1.1. AU CLIC DE LA SOURIS

**click()** Associe une fonction à l'événement clic des éléments de la sélection.

```
$( "p" ).click(
    function (event) {
        console.log(event.type); /* type d'événement */
        console.log(event.which) ; /* bouton pressé */
        console.log(event.pageX) ; /* coordonnée X */
        console.log(event.pageY) ; /* coordonnée Y */
    }
);
```

Exercice : Au clic d'un paragraphe, celui-ci disparaît.

##### 1.1.2. AU DOUBLE CLIC

**dblclick()** Associe une fonction à l'événement double clic.

```
$( "div" ).dblclick();
```

Exercice : Au double-clic d'un titre, celui-ci est entouré d'une bordure.

##### 1.1.3. LE FOCUS

**focus()** Associe une fonction au focus des éléments spécifiés.

```
$( "p" ).focus();
```

Exercice : Au focus de la box input, lui ajouter la valeur *Mention obligatoire*.

##### 1.1.4. LA PERTE DU FOCUS

**blur()** Associe une fonction au dé-focus des éléments spécifiés.

```
$( "input" ).blur();
```

Exercice : Après l'écriture du champ relatif au nom (au-moins un caractère) et la perte du focus de celui-ci, son arrière-plan est colorié et "OK" est ajouté.

##### 1.1.5. LA BARRE DE DÉFILEMENT

**scroll()** Associe une fonction à la barre de défilement d'un élément.

```
$(window).scroll();
```

Exercice : Un texte apparaît lorsque l'utilisateur utilise la barre de défilement de la textarea.

### 1.1.6. LE BOUTON DE LA SOURIS

**mousedown()** Associe une fonction pour les éléments sélectionnés lorsque l'utilisateur presse une touche de la souris.

```
$( "p" ).mousedown() ;
```

**mouseup()** Associe une fonction pour les éléments de la sélection lorsque l'utilisateur relâche le bouton de la souris.

```
$( "p" ).mouseup() ;
```

L'événement clic est activé lorsqu'un `mousedown` et un `mouseup` est détecté.

*Exercice :* Au clic sur le bouton, détailler les événements `mousedown`, `mouseup` et `click` dans la div `#sortie`. Utilisez l'objet 'event'.

### 1.1.7. LE DÉPLACEMENT DU CURSEUR

Détecter le moindre mouvement du curseur est un événement intéressant mais il faut être conscient que cela entraîne une forte consommation des ressources du système. En effet, les actions associées au mouvement de la souris se déclenchent dès que le curseur bouge d'un pixel.

**mousemove()** Associe une fonction lorsque l'utilisateur bouge le curseur.

```
$( "div" ).mousemove() ;
```

*Exercice :* Retrouver les coordonnées du curseur de la souris à chacun de ses mouvements au-dessus de la div et les afficher dans le span.

### 1.1.8. L'ENTRÉE ET LA SORTIE DU CURSEUR

**mouseover()** Associe une fonction lorsque le pointeur entre dans un élément ou un des enfants de l'élément.

```
$( "div" ).mouseover() ;
```

**mouseenter()** Associe une fonction lorsque le pointeur entre dans un élément.

```
$( "div" ).mouseenter() ;
```

**mouseout()** Associe une fonction lorsque le curseur de la souris quitte un élément ou un des enfants de l'élément.

```
$( "div" ).mouseout() ;
```

**mouseleave()** Associe une fonction lorsque le pointeur quitte un élément.

```
$( "div" ).mouseleave() ;
```

Le fonctionnement de ces gestionnaires d'événements est différent pour jQuery. En effet, `mouseover` propage l'événement à la hiérarchie des objets de la page. Ce phénomène est connu sous le terme propagation des événements ou débordement des événements. Au contraire, `mouseenter` évite cette propagation de l'événement.

Il en est de même pour `mouseout` qui se déclenche à chaque fois que le pointeur se déplace vers ou à partir d'un élément enfant. À l'inverse, `mouseleave` ne se déclenche qu'une fois, soit lorsque le pointeur quitte l'élément en cours. Dans

certaines situations de programmation, cette propagation peut se révéler gênante et jQuery y apporte une solution.

***Exercice :** Lorsque le curseur de la souris survole (mouseover et mouseenter) la division colorée en bleu, les compteurs « mouseovercount » et « mouseentercount » doivent être incrémentés.*

## 1.2. MÉTHODES OU GESTIONNAIRES D'ÉVÉNEMENTS AVANCÉS

Vous pouvez programmer en jQuery pendant de longues années en utilisant simplement les gestionnaires d'événements passés en revue précédemment. Cependant, pour faire de la délégation d'événements ainsi que pour gérer des espaces de noms d'événement, il va être nécessaire d'aller plus loin.

### 1.2.1. LIER UN ÉVÉNEMENT À UN OBJET

La méthode `on()` est plus flexible et plus puissante que les événements spécifiques comme `click()` ou `mouseover()` parcourus ci-avant.

#### **on(événement, 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.
- fonction : fonction à associer à l'événement.

```
$( "p" ).on( "mouseover mouseout", function() {  
    $( this ).text( "-- " + $( this ).text() + " --" );  
});
```

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

***Exercice :** Relier les événements `mouseenter` et `mouseleave` à la balise de paragraphe. Cette nouvelle association a comme action de mettre ou non un arrière-plan de couleur (utilisez CSS et `toggle`).*

### 1.2.2. EXÉCUTER UNE FONCTION UNE SEULE FOIS

#### **one(événement, fonction)**

Associe une fonction à un événement donné. La différence avec la fonction `bind()` est que la fonction associée à l'événement ne sera exécutée au maximum qu'une seule fois pour chaque élément de la sélection.

- événement (chaîne de caractères) : type d'événement concerné.
- fonction : fonction à associer à l'événement.

```
$( "p" ).one( "click", function() {  
    alert( $( this ).text() );  
});
```

***Exercice :** Soit un bouton que l'on ne peut cliquer qu'une fois. Au premier clic, un message est affiché.*

### 1.2.3. PREVENTION DE L'ACTION PAR DEFAULT

**event.preventDefault()** Empêche l'action par défaut du navigateur pour ce type d'événement (si elle existe).

```
$("#a").on("click", function(event){
    event.preventDefault(); /* empêche d'ouvrir le lien */
});
```

*Exercice :* Au clic de la checkbox, la checkbox n'est PAS cochée. Au clic du lien, la checkbox est cochée `$("#..").prop("checked",true)`

### 1.2.4. DELEGATION D'ÉVÉNEMENTS

Soit un événement `click` sur un élément de liste `<li>` qui génère un nouvel élément de liste. L'événement `click` doit continuer à fonctionner pour tous ces éléments de liste nouvellement créés.

**on(événement, élément, fonction)** Attache une fonction à un événement donné pour tous les éléments existants et futurs trouvés.

```
$("#ul").on("click", li, function(){
    alert("test");
});
```

Sans cette manière de faire, il faut savoir que jQuery fonctionne sur les éléments existant au chargement du DOM. Si un événement génère un nouvel élément, votre fonction n'agira pas sur celui-ci.

*Exercice :* Le clic génère un nouveau paragraphe. En cliquant sur le paragraphe initial ou sur le paragraphe nouvellement créé, la fonction ajoute à nouveau un paragraphe. Et ainsi de suite...

## 1.3. MÉTHODES PROPRES À JQUERY

### 1.3.1. AU PASSAGE DE LA SOURIS

Cette fonction propre à jQuery reprend `mouseover` et `mouseout`, soit lorsque le curseur survole un élément et quitte celui-ci.

**hover(fonction 1, fonction 2)** La méthode `hover()` de jQuery lie deux événements fréquemment utilisés à l'élément sélectionné ; l'un lorsque le pointeur survole et l'autre lorsqu'il le quitte.

Cette méthode s'effectue en deux temps. Quand le curseur se situe au-dessus d'un élément déterminé, la première fonction passée en paramètre est exécutée. Lorsque le curseur sort du cadre de l'élément, la seconde fonction est exécutée.

```
$("#p").hover(
    function(){ $(this).addClass("hover"); },
    function(){ $(this).removeClass("hover"); }
);
```

Au survol des paragraphes, la classe `hover` est ajoutée. Lorsque le curseur quitte la zone, la classe est retirée. Cette méthode retourne un objet jQuery.

*Exercice* : classique, une image qui change au survol de la souris.

## 1.4. APPLICATIONS

*Exercice (1.4.1)* : Au survol d'un item, celui-ci change de couleur d'arrière plan.

*Exercice (1.4.2)* : Concevoir un script qui permet d'afficher une image en taille réelle au survol d'une vignette de celle-ci.

## 2. LES EFFETS

### 2.1. AFFICHER ET CACHER

Les méthodes `show()` et `hide()` de jQuery permettent de faire apparaître et disparaître des éléments.

**show(vitesse, fonction de rappel)** Affiche un élément sélectionné (pour autant que celui-ci soit caché). L'animation modifie dynamiquement la hauteur, la largeur et l'opacité de l'élément. Depuis la spécification jQuery 1.3, les marges externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet. `$('p').show('slow');`
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet

```
$('p').show('slow',  
            function(){ alert("Fin"); }  
);
```

**hide(vitesse, fonction de rappel)** Cache un élément sélectionné (pour autant que celui-ci soit visible). L'animation modifie dynamiquement la hauteur, la largeur et l'opacité de l'élément. Depuis la spécification jQuery 1.3, les marges externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet. `$('p').hide('fast');`
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet.

```
$('p').hide('normal',  
            function(){alert("Fin");}  
);
```

*Exercice (2.1.1)* : proposer au lecteur d'afficher la suite d'un article (le deuxième paragraphe), après avoir cliqué sur le lien.

*Exercice (2.1.2)* : permettre de dérouler des listes imbriquées (ayant un ul) par un clic.

## 2.2. GLISSER VERTICALEMENT

Les fonctions `slideDown()` et `slideUp()` permettent de jouer dynamiquement sur la hauteur d'un élément, généralement une division `<div> ... </div>`.

**slideDown(vitesse, fonction de rappel)** Fait glisser vers le bas (down) un élément sélectionné. L'animation modifie seulement la hauteur.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet.
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet

```
$('div').slideDown('fast',  
    function(){alert("Fin");}  
);
```

**slideUp(vitesse, fonction de rappel)** Fait glisser vers le haut (up) un élément sélectionné. L'animation modifie seulement la hauteur. Depuis la spécification jQuery 1.3, les marges verticales, externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet. `$('div').slideUp('fast');`
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet

```
$('div').slideUp('fast',  
    function(){alert("Fin");}  
);
```

Exercice (2.2) : Au clic sur le bouton, dérouler une division.

## 2.3. RÉALISER UN EFFET DE FONDU

**fadeIn(vitesse, fonction de rappel)** Fait apparaître l'élément sélectionné selon un effet de fondu. Cette animation est obtenue en ajustant uniquement l'opacité. L'élément sélectionné doit ainsi déjà avoir une largeur et une hauteur spécifiée.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet.
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet

```
$('p:first').fadeIn(4000,  
    function(){alert("Fin");}  
);
```

**fadeOut(vitesse, fonction de rappel)** Fait disparaître l'élément sélectionné selon un effet de fondu. Cette animation est obtenue en ajustant uniquement l'opacité. L'élément sélectionné doit ainsi déjà avoir une largeur et une hauteur spécifiée.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses

prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet.

- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet.

```
$('#p:first').fadeOut(4000,  
    function(){alert("Fin");}  
);
```

**fadeTo(vitesse, opacité, fonction de rappel)** Modifie l'opacité de l'élément sélectionné jusqu'à une valeur fournie dans le script.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet. `$('#p:first').fadeTo('slow', 0.33);`
- opacity(nombre) : détermine la valeur de l'opacité à atteindre (nombre compris entre 0 et 1).
- fonction de rappel (callback) (optionnel) : s'exécute à la fin de l'effet

```
$('#p:first').fadeTo('slow', 0.33,  
    function(){alert("Fin");}  
);
```

Exercice (2.3.1) : Apparition et disparition progressive sur une image.

Exercice (2.3.2) : Au chargement de la page, faire disparaître les images. Ensuite, lorsque l'on repasse dessus, les rendre moins opaques.

## 2.4. BASCULER D'UN EFFET À L'AUTRE

**slideToggle(vitesse, fonction de rappel)** Cette fonction fait glisser vers le bas un élément qui est en état "Up" et fait glisser vers le haut un élément qui est en état "Down".

L'animation modifie seulement la hauteur. Depuis la spécification jQuery 1.3, les marges verticales, externes et internes sont également modifiées pour obtenir un effet plus fluide.

- vitesse (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('slow', 'normal' ou 'fast') ou le nombre en millisecondes correspondant à la durée de l'effet.
- fonction de rappel (callback) (optionnel) : fonction à exécuter à la fin de l'effet.

```
$('#div').slideToggle('fast',  
    function(){alert("Fin");}  
);
```

Exercice (2.4) : Au clic sur l'élément d'identifiant 'start', faire disparaître une div lentement, la faire réapparaître lentement au clic suivant.

## 2.5. DIFFERER UN EFFET

**delay(durée)** Permet de retarder le déroulement d'un effet

- **durée** : durée est un entier exprimant en millisecondes le retard à apporter pour l'exécution de l'effet suivant.

*Exercice (2.5) : faire apparaître des divisions en deux temps avec un délai de 2 secondes pour la deuxième. Au clic du bouton, seule la division 1 s'affiche. La division 2 apparaît avec un retard de 2 secondes.*

## 2.6. CRÉER UNE ANIMATION

La fonction `animate()` permet de créer et de paramétrer vos propres animations au gré de votre créativité.

**animate(paramètres, vitesse, easing, fonction de rappel)** L'aspect clé de cette fonction est l'objet composé des propriétés de style sur lesquelles sera basée l'animation. Chaque paramètre de l'objet représente une propriété sur laquelle portera l'animation (exemple: `height`, `top` ou `opacity`).

La valeur associée à la clé indique comment la propriété sera animée. Si la valeur est un nombre, le style de la propriété passera de sa valeur actuelle à la valeur spécifiée. Si la valeur `hide`, `show` ou `toggle` est spécifiée, une animation par défaut sera construite pour cette propriété.

- **paramètres** : conteneurs d'attributs de style que vous souhaitez animer et à quelle valeur.
- **vitesse** (optionnel) : chaîne de caractères représentant une des trois vitesses prédéfinies ('`slow`', '`normal`' ou '`fast`') ou le nombre en millisecondes correspondant à la durée de l'effet.
- **easing** (optionnel) : nom de l'effet que vous souhaitez utiliser (plugin requis).
- **fonction de rappel** (optionnel) : fonction à exécuter à la fin de l'animation.

```
$('#div').animate({
    "font-size":"24px",
    "left":"300",
    "width":"200px",
    "opacity":0.5},
1000 );
```

*Exercice (2.6) :*

*Au clic sur le bouton Lancer l'animation.*

*La boîte `box` est déplacée vers la droite de 150 pixels (`left: 150`) , sa largeur est portée à 200 pixels et son opacité diminuée de moitié. L'animation dure 1500 millisecondes.*

*Le clic sur le bouton Reset remet la boîte dans sa position initiale en annulant toutes les modifications réalisées sur la largeur, la position, l'opacité ainsi que la largeur de la bordure gauche.*