M2202 – Algorithmique
TD 2 : Interactivité utilisateur et menus

Les actions de l'utilisateur constituent des événements : clic, survol d'un élément, etc. Jquery permet de traiter de manière simple ces événements, et leur associer des actions.

Quelques rappels et exemples
Gestion du clic sur tous les éléments div (affichage d'une boite d'alerte avec le mot clic !!) :

```
$("div").click(function(){
     alert("Clic !!");
});
```

Gestion du survol sur tous les éléments div (modification de la couleur d'arrière-plan de la div en vert):

```
$("div").mouseover(function(){
     $(this).css("background-color", "green");
});
```

1. **Exercice 1** : améliorer l'ergonomie

Reprenons notre page réalisée lors du dernier exercice du TD1 : 2 boutons permettent d'afficher et d'effacer des paragraphes selon leur classe.

```
<!DOCTYPE html>
<html>
<head>
    <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'>
    </script>
    <script>
    $(document).ready(function(){
        $('#cache').click(function(){
        $('.test').hide();
        });
```

```
        $('#affiche').click(function(){
        $('.test').show();
      });
    });
    </script>
</head>
<body>
    <h2 class='test autre'>Ent&ecirc;te</h2>
    <p class='test'>Un paragraphe.</p>
    <p>Un autre paragraphe.</p>
    <button id='cache'>Effacer les &eacute;l&eacute;ments de classe test</button>
    <button id='affiche'>Afficher les &eacute;l&eacute;ments de classe test</button>
</body>
</html>
```

On remarque que, contrairement à ce que recommandent les règles d'ergonomie, le survol des éléments de type **button** ne provoquent pas l'affichage du curseur 'main' (*pointer*), indiquant l'interactivité sur l'objet. Nous allons pour corriger cela utiliser la propriété CSS2 **cursor**.
Cela peut être facilement géré avec à jQuery, et ce de plusieurs manières.

Méthode 1 : affecter une classe existante

Il s'agit d'abord de créer une classe dont la propriété CSS *cursor* aura la valeur *pointer*, nous nommerons cette classe *hand*.

Modifiez l'élément **head** pour ajouter cette propriété CSS pour la classe ***hand*** (on utilisera un élément **style**).
Modifiez la fonction $(document) .ready pour y ajouter le de code suivant : `$('button').addClass('hand') ;`

Cette ligne de code ajoute la classe *hand* à tous les éléments de type *button*. De ce fait, le curseur est modifié au survol des éléments **button**.

Enregistrez votre page et vérifiez le bon fonctionnement de votre code.

Méthode 2 : utiliser une fonction anonyme
Dans ce cas, il n'est pas utile de définir le comportement d'une classe, nous n'avons plus besoin de l'élément *style*.
Enregistrez votre page sous un nouveau nom.
Supprimez l'élément style de l'en-tête. Il s'agit maintenant d'utiliser la méthode **addClass()** en lui passant non pas un nom de classe, mais un comportement via une fonction anonyme.
Dans cette fonction, on appellera la méthode css de l'élément pour en modifier la propriété cursor. L'écriture du code est la suivante :

```
$('button').addClass(function(){$(this).css('cursor','pointer');});
```

Explications :

```
$('button').addClass(function(){$(this).css('cursor','pointer');});
//pour tous les éléments de type button

$('button').addClass(function(){$(this).css('cursor','pointer');});
//On ajoute un comportement de classe

$('button').addClass(function(){$(this).css('cursor','pointer');});
//auquel on passe une fonction anonyme

$('button').addClass(function(){$(this).css('cursor','pointer');});
//On récupère l'élément concerné - c'est un objet jQuery → $(this)!!

$('button').addClass(function(){$(this).css('cursor','pointer');});
//dont on modifie une propriété CSS : la propriété cursor prend la valeur pointer
//syntaxe de css() pour modifier une propriété : $('filtre').css('propriété', 'valeur')
```

N.B. : en utilisant **addClass()** de cette façon, les éléments **button** ne se voient affectés aucune classe particulière, mais un attribut de style, comme on peut le vérifier en inspectant les éléments (mais pas en affichant le code source).

Modifiez la ligne de code suivant ce qui vient d'être expliqué, vérifiez-en le bon fonctionnement.

2. **Exercice 2** : Une introduction aux menus

Il s'agira de modifier l'apparence d'une liste au survol ; bien que cette fonctionnalité soit accessible dans une certaine mesure avec la pseudo-classe **:hover**, notez que :

- :hover ne gère que l'élément auquel il est appliqué (pas de survol distant) ;
- le comportement css géré par:hover ne peut pas être modifié ;
- :hover enfin ne gère que des propriétés css.

JavaScript, et donc jQuery, permet d'outrepasser facilement ces limitations.

Gérer un survol avec jQuery

Nous savons déjà comment répondre à une action sur le bouton de la souris grâce à la méthode **click()**. D'autres méthodes permettent de gérer le survol d'éléments :

- lorsque l'on entre dans l'élément : `mouseenter()` ;
- lorsque l'on sort de l'élément : `mouseleave()` ;

Comme pour la méthode click(), on peut passer en argument une fonction anonyme :

```
$('li').mouseenter(function(){
     //action lorsque le curseur entre dans l'élément
}) ;
```

Nous sommes prêts maintenant à mettre en place le code html/css et jQuery.

Créez et enregistrez la page css suivante (mapagejq4.css):

```
@charset "utf-8";
/* CSS Document */
#navigation {
     width: 200px;
}
ul {
     list-style-type: none;
     margin: 4px;
     font-size: 16px;
     font-weight: bolder;
     font-family:Arial, Helvetica, sans-serif;
}
.bleu {
     color: blue;
}
.souligne {
     text-decoration: underline;
}
.stabilo {
     background: yellow;
}
```

Créez et enregistrez ensuite la page html suivante :

```
<!DOCTYPE html>
<html>
<head>
    <meta charset='utf-8'>
    <link rel='stylesheet' href='mapagejq4.css'>
</head>
<body>
    <div id='navigation'>
        <ul>
            <li class='bleu'>Menu1</li>
            <li class='bleu stabilo'>Menu2</li>
```

```
            <li class='bleu souligne'>Menu3</li>
            <li class='bleu souligne'>Menu4</li>
        </ul>
    </div>
</body>
</html>
```

Nous allons maintenant ajouter le code jquery nécessaire à la gestion des survols des éléments li ; pour cela, ajoutez le code suivant :

```
<head>
    <meta charset='utf-8'>
    <link rel='stylesheet' href='mapagejq4.css'>
    <script src='http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js'>
    </script>
    <script>
    $(document).ready(function(){
          $('li:even').mouseenter(function(){
               $(this).removeClass('bleu');
          });
    });
    </script>
</head>
```

Explications

Au chargement de la page, on affecte à tous les élément li pairs - **`$('li:even')`** - un gestionnaire de survol qui provoque le retrait de la classe bleu à l'élément survolé - **`$(this).removeClass('bleu')`** – lorsque le curseur entre dans l'élément - **`.mouseenter().`**

Enregistrez et vérifier le fonctionnement : seuls les textes Menu1 et Menu 3 (li pairs) perdent leur effet de style → couleur bleu !!

Pour aller plus loin, ajoutez le code qui restaure la classe lorsque le curseur sort de l'élément.