

TP 1 : INSTALLATION DE JQUERY ET SELECTEURS

INSTALLATION DE L'ENVIRONNEMENT DE DEVELOPPEMENT

Il y a deux manières de procéder :

(1) Vous pouvez télécharger le fichier `jquery.js` pour le charger localement. La dernière version de la librairie jQuery est téléchargeable sur : <http://code.jquery.com>
Ensuite, il vous reste à intégrer la librairie dans le `<head>` de votre page HTML.

```
/* Appel de la librairie */  
<script type="text/javascript" src="jquery.js"></script>
```

(2) Vous pouvez aussi utiliser le fichier hébergé par des serveurs web comme Google. Regardez sur <https://developers.google.com/speed/libraries/> pour toutes les librairies hébergées par Google.

```
/* Appel de la librairie */  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery  
.min.js"></script>
```

Une fois jQuery chargé, vous pouvez écrire votre code soit entre balises `<script>`, soit dans un fichier javascript externe.

```
/* entre balises */  
<script type="text/javascript"> . . . </script>  
/* appel fichier externe */  
<script src="chemin/vers/fichier.js"></script>
```

LA FONCTION `$()`, CHARGEMENT DOM ET EXEMPLE

La librairie jQuery fournit la fonction du même nom, aussi nommée `$()`, qui permet de sélectionner les éléments d'une page HTML et de les manipuler selon un paradigme de programmation orientée objet.

```
var listeParagraphes = jQuery("p");  
var listeParagraphes = $("p");
```

En sélectionnant l'objet `"document"`, `$()` permet de vérifier si le DOM (Document Object Model) de votre document HTML a correctement été chargé avant de le manipuler. La méthode `.ready()` prend une fonction anonyme qui est exécutée après le chargement.

```
$(document).ready(function() {  
    console.log("Document chargé");  
});
```

Par simplicité, il est possible de la réécrire comme ci-dessous. Attention aux parenthèses.

```
$(function() {  
    console.log("Document chargé");  
});
```

Une fois le DOM chargé, il est possible de sélectionner les éléments du document et de leur appliquer des fonctions. Imaginons un exemple simple d'un menu déroulant dynamique.

```
<body>
<ul>
  <li class="mychapter">Chapitre 1
    <ul class="mysection">
      <li>Section 1.1</li>
      <li>Section 1.2</li>
    </ul>
  </li>
  <li class="mychapter">Chapitre 2
    <ul class="mysection">
      <li>Section 2.1</li>
      <li>Section 2.2</li>
    </ul>
  </li>
</ul>
</body>
```

Une première opération consiste à ne pas afficher les sous-menus, soit les divisions dont la classe est "mysection".

```
$(".mysection").hide();
```

Ensuite, pour faire défiler les sections lors d'un clique de l'utilisateur, nous lions l'action 'click' à une fonction anonyme qui va (1) faire descendre les sections du chapitre cliqué, et (2) va faire remonter les sections de l'autre chapitre.

```
$(".mychapter").click(function(){
  console.log("Click");
  /* fait descendre les éléments enfants en 300 ms */
  $(this).children().slideDown(300);
  /* fait remonter les éléments voisins */
  $(this).siblings().children().slideUp("slow");
});
```

Il est aussi possible de modifier les propriétés CSS des éléments du document HTML :

```
$(".mychapter li").css({
  "background-color":"yellow",
  "font-size":"10px"
});
```

Dans un premier temps, reproduisez le code ci-dessus dans un fichier nommé *exemple_menu.html*. Pour aller plus loin, considérez utiliser `slideToggle()`.

LES SELECTEURS

Dans la suite de ce TP, nous allons considérer les différentes manières de sélectionner des éléments du document HTML.

Téléchargez les fichiers d'exercice à l'adresse https://github.com/fauconnier/ITW_TP
Pour chaque exercice (ex. 1.2), complétez le fichier HTML correspondant (1.2.html).

1. LES SELECTEURS DE BASE

1.1 SELECTION PAR L'IDENTIFIANT

#identifiant

Sélectionne l'élément (unique) spécifié par l'attribut `id="identifiant"`.

`$("#box")` : sélectionne l'élément dont l'id est `box`.

Exercice : Entourer d'une bordure le paragraphe dont l'identifiant est "deux".

1.2 SELECTION PAR LE NOM DE LA BALISE

Elément

Sélectionne tous les éléments (ou balises) dont le nom est spécifié.

`$("div")` : sélectionne toutes les divisions `<div>` de la page.

Exercice : Entourer d'une bordure tous les paragraphes du document.

1.3 SELECTION PAR LA CLASSE

.classe

Sélectionne tous les éléments (ou balises) avec la classe spécifiée.

`$(".texte")` : sélectionne tous les éléments dotés de l'attribut `class="texte"`.

Exercice : Entourer d'une bordure le paragraphe doté de la classe *gras*.

2. LES SELECTEURS HIERARCHIQUES

2.1 SELECTION DES DESCENDANTS

Ascendant Descendant

Sélectionne tous les descendants de l'élément noté "Descendant" par rapport à l'élément parent noté "Ascendant".

`$("#box p")` : sélectionne tous les descendants de `<p>` contenus dans l'élément parent identifié par `box`.

Les descendants peuvent être les enfants, les petits-enfants, etc.

Exercice : Entourer d'une bordure tous les descendants de l'élément identifié par `id="box"`.

2.2 SELECTION DES ENFANTS

Parent > Enfant

Sélectionne tous les éléments notés par "Enfant" qui sont les enfants directs de l'élément parent noté "Parent".

`$("#box > p")` : sélectionne tous les enfants immédiats `<p>` contenus dans l'élément parent identifié par `box`.

Exercice : Entourer d'une bordure les enfants (et uniquement les enfants) de l'élément identifié par `"box"`.

2.3 SELECTION DES FRERES SUIVANTS

Précédent ~ Frères

Cible les éléments frères situés après l'élément identifié par le sélecteur "Précédent" et qui répondent au sélecteur "Frères".

`$("#prev ~ div")` trouve toutes les divisions `<div>` qui sont frères après l'élément avec `#prev` comme identifiant.

Exercice : Entourer d'une bordure les éléments frères du premier item de la liste.

2.4 SELECTION DE L'ELEMENT SUIVANT

Précédent + Suivant

Cible l'élément immédiatement suivant situé après l'élément identifié par le sélecteur "Précédent" et qui répond au sélecteur "Suivant".

`$("#prev + div")` trouve la division `<div>` qui suit l'élément avec `#prev` comme identifiant.

Exercice : Mettre un arrière-plan à l'élément suivant au premier item de la liste.

3. LES FILTRES JQUERY DE BASE

3.1 LE PREMIER ELEMENT

:first

Sélectionne la première instance d'un élément.

`$("li:first")` : sélectionne le premier élément de liste `` du document.

Exercice : Mettre un arrière-plan de couleur à la première ligne du tableau.

3.2 LE DERNIER ELEMENT

:last

Sélectionne la dernière instance d'un élément.

`$("li:last")` : sélectionne le dernier élément de liste `` du document.

Exercice : Mettre un arrière-plan de couleur à la dernière cellule d'un tableau.

3.3 LES ELEMENTS PAIRS

:even

Sélectionne les éléments pairs selon un index commençant à 0.

`$("tr:even")` : sélectionne les lignes d'index 0, 2, 4

Exercice : Dans un tableau, doter une ligne sur deux d'un arrière-plan.

3.4 LES ELEMENTS IMPAIRS

:odd

Sélectionne les éléments impairs selon un index commençant à 0.

`$("tr:odd")` : sélectionne les cellules d'index 1, 3, 5.

Exercice : Appliquer le même effet au tableau en dotant une ligne sur deux d'un arrière-plan mais avec d'autres lignes qu'à l'exemple précédent.

3.5 UN ELEMENT DETERMINE

:eq(index)

Sélectionne l'élément déterminé par la valeur de l'index.

`$("td:eq(2)")` : sélectionne la troisième cellule d'un tableau.

Exercice : Entourer d'une bordure l'élément de liste qui apparaît en seconde position à l'écran.

3.6 LES ELEMENTS SUIVANTS

:gt(index)

Sélectionne les éléments avec une valeur d'index supérieure (greater than) à la valeur fournie en paramètre.

`$("a:gt(1)")` : sélectionne tous les liens de la page en commençant au-delà de l'index 1.

Exercice : Entourer d'une bordure les éléments à partir de l'item en quatrième position (inclus) jusqu'à la fin de la liste.

3.7 LES ELEMENTS PRECEDENTS

:lt(index)

Sélectionne les éléments avec une valeur d'index inférieure (less than) à la valeur fournie en paramètre.

`$("p:lt(3)")` : sélectionne tous les paragraphes situés avant le quatrième

Exercice : Entourer d'une bordure les quatre premiers éléments de la liste.

3.8 LES BALISES DE TITRE

:header

Retourne tous les éléments qui sont des balises de titre (`<h1>`, `<h2>`, etc.).

`$(":header")` : sélectionne toutes les balises de titre de la page.

Exercice : Entourer d'une bordure toutes les balises de titre de la page.

3.9 EXCLUSION D'UN ELEMENT

:not(sélecteur)

Exclut de la sélection tous les éléments qui répondent au critère spécifié .

`$("div:not(.box)")` : sélectionne toutes les divisions sauf celle de classe box.

Exercice : Entourer d'une bordure tous les éléments d'une liste à l'exclusion de premier et dernier item.

4. LES FILTRES ENFANTS

4.1 LE PREMIER ENFANT

:first-child

Sélectionne tous les éléments qui sont le premier enfant de leur parent.

`$("ul li:first-child")` : sélectionne les premiers items des listes ul.

`$("ul :first-child")` : sélectionne les premiers éléments dans les listes ul.

Exercice : Entourer d'une bordure les premiers liens dans les paragraphes.

4.2 LE DERNIER ENFANT

:last-child

Sélectionne tous les éléments qui sont le dernier enfant de leur parent.

`$("ul li:last-child")` : sélectionne les derniers items des listes ul.

`$("ul :last-child")` : sélectionne les derniers éléments dans les listes ul.

Exercice : Entourer d'une bordure les derniers éléments dans les paragraphes.

4.3 LE NIÈME ENFANT

:nth-child(index)

Sélectionne les éléments qui sont le *nième* enfant de leur parent. La position est fournie par le paramètre `index`. Au contraire de nombreux index en jQuery, l'index ne débute pas ici à 0 mais à 1, car dérivé du CSS.

`$("ul li:nth-child(2)")` : sélectionne le second élément `` de la liste ``.

Exercice : Entourer d'une bordure l'élément de liste qui apparaît en troisième position à l'écran.

4.4 LES ENFANTS PAIRS ET IMPAIRS

:nth-child(even ou odd)

Sélectionne les éléments qui sont les *nièmes* enfants pairs (even) ou impairs (odd) de leur parent. Comme `:nth-child`, l'index débute à 1.

`$("table tr:nth-child(even)")` : sélectionne les lignes paires du tableau.

Exercice : Mettre un arrière-plan de couleur aux colonnes impaires du tableau.

4.5 LES ENFANTS UNIQUES

:only-child

Sélectionne tous les éléments qui sont enfant unique de leur parent. Si le parent a plusieurs enfants, aucune sélection n'est effectuée.

`$("div button:only-child")` : retrouve les boutons (balise `<button>`) qui n'ont pas de frère(s) dans toutes les divisions rencontrées.

Exercice : Entourer d'une bordure le lien qui est enfant unique d'une balise de paragraphe.

5. LES FILTRES DE CONTENU

5.1 UN TEXTE DONNE

:contains(texte)

Sélectionne les éléments qui contiennent les caractères fournis en argument.

`$("div:contains('aab')")` : sélectionne les div qui contiennent le texte "aab".

Exercice : Mettre un arrière-plan aux paragraphes qui contiennent le fragment de texte "En".

5.2 UN CONTENU VIDE

:empty

Sélectionne tous les éléments qui n'ont pas d'enfants (inclus les nœuds de texte).

`$("div:empty")` : sélectionne les divisions vides.

Exercice : Mettre un arrière-plan aux cellules vides d'un tableau.

5.3 LA QUALITE DE PARENT

:parent

Sélectionne les éléments qui sont parent, c'est-à-dire qui ont des éléments enfants, inclus les nœuds de texte.

`$("div:parent")` : sélectionne les divisions qui ont des éléments enfants. Ce sélecteur est en quelque sorte l'inverse du précédent.

Exercice : Mettre un arrière-plan aux cellules non vides d'un tableau.

5.4 UN SELECTEUR DETERMINE

:has(sélecteur)

Sélectionne les éléments qui contiennent le sélecteur transmis en argument.

`$("div:has(p)")` : sélectionne les divisions qui contiennent des paragraphes

Exercice : Mettre un arrière-plan au paragraphe qui a lien de classe "special".

6. LES FILTRES DE VISIBILITE

6.1 ELEMENT VISIBLE

:visible

Sélectionne les éléments qui sont visibles.

`$("p:visible")` : sélectionne les paragraphes visibles.

Exercice : Doter les divisions visibles d'un arrière-plan de couleur.

6.2 ELEMENT CACHE

:hidden

Sélectionne les éléments qui sont cachés.

`$("p:hidden")` : sélectionne les paragraphes cachés.

Exercice : Afficher, au clic du bouton, les divisions cachées.

7. LES FILTRES D'ATTRIBUT

7.1 L'ATTRIBUT

[attribut]

Sélectionne les éléments dotés d'un attribut spécifié.

`$("div[id]")` : sélectionne les éléments qui ont un attribut `id`.

Exercice : Mettre un arrière-plan aux éléments avec l'attribut `class`.

7.2 L'ATTRIBUT AVEC UNE CERTAINE VALEUR

[attribut=valeur]

Sélectionne les éléments dotés d'un attribut avec une valeur déterminée.

`$("input[name='newsletter']")` : sélectionne l'élément de formulaire `<input>` avec un attribut `name="newsletter"`.

Exercice : Mettre un arrière-plan aux éléments de liste dotés de classe "gras".

7.3 L'ATTRIBUT QUI N'A PAS UNE CERTAINE VALEUR

[attribut!=valeur]

Sélectionne les éléments qui n'ont pas l'attribut spécifié et ceux qui ont l'attribut spécifié mais pas avec la valeur indiquée..

`$("input[name!=newsletter]")` : sélectionne les éléments de formulaire `<input>` qui n'ont pas d'attribut `name` et ceux qui ont un attribut `name` avec une autre valeur que `newsletter`.

Exercice : Mettre un arrière-plan au liens sans `title="lien interne"`.

7.4 L'ATTRIBUT DONT LA VALEUR COMMENCE PAR

[attribut^=value]

Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur commence par les caractères indiqués.

`$("input[name^='news']")` : sélectionne les éléments `<input>` avec l'attribut `name` et dont la valeur de celui-ci commence par les caractères "news".

Exercice : Mettre un arrière-plan aux liens où l'attribut `title` débute par "X".

7.5 L'ATTRIBUT DONT LA VALEUR FINIT PAR

[attribut\$=value]

Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur se termine par les caractères indiqués.

Exercice : Mettre un arrière-plan aux liens dont l'attribut termine par "ml".

7.6 L'ATTRIBUT DONT LA VALEUR CONTIENT

[attribut*=value]

Sélectionne les éléments qui possèdent l'attribut spécifié et dont la valeur contient les caractères indiqués.

Exercice : Mettre un arrière-plan aux liens dont l'attribut `title` contient "tm".

7.7 LES FILTRES MULTIPLES D'ATTRIBUT

[filtre d'attribut 1][filtre d'attribut 2][filtre d'attribut 3]...

Sélectionne les éléments qui répondent à tous les filtres d'attribut spécifiés.

Exercice : Retenir les liens dont l'attribut `title` commence par "lien", se termine par "interne" et comporte "chapitre 1".