

Université Montpellier-II
IUT de Béziers
Département SRC



Programmation Web avancée jQuery et Ajax

Responsable : Chouki TIBERMACHINE
Bureau : Direction des études
Tél. : 04 67 11 18 21
Mél. : Chouki.Tibermachine@iutbeziers.fr
Web : <http://www.lirmm.fr/~tibermacin/ens/pwa/>

2011-2012

Cours 3

Sélecteurs et fonctions de sélection

Plan du cours

- Sélecteurs jQuery
- Fonctions de sélection

Plan du cours

- Sélecteurs jQuery
- Fonctions de sélection

Sélecteurs de base

- Sélectionner des balises avec leur nom : mettre le nom de la balise "p" : permet de sélectionner toutes les balises paragraphes
- Sélectionner toutes les balises indépendamment de leur nom : * (on appelle ça un joker : *wildcard*)
- Ce sont les deux sélecteurs de base. Ensuite, on applique des filtres (#id, :xxx, ...)
- Le joker est le sélecteur par défaut si rien n'est précisé avant les filtres
- Les filtres sont appliqués de gauche à droite et chacun réduit davantage le nombre d'éléments sélectionnés

Filtres des sélecteurs

- `#id` : sélectionne l'élément ayant un id dont la valeur est égale à ce qui est précisé après le symbole `#`
Ce sélecteur choisit une seule balise
- `.class` : sélectionne les éléments ayant comme classe CSS `class` ou bien ayant plusieurs classes CSS dont `class`
- `[attr]` : sélectionne tout élément HTML ayant un attribut qui a comme nom `attr`, indépendamment de la valeur de cet attribut
- `[attr=val]` : sélectionne tout élément HTML ayant un attribut qui a comme nom `attr` et qui a la valeur `val`

Filtres des sélecteurs -suite-

- `[attr!=val]` : sélectionne tout élément HTML n'ayant pas un attribut qui a comme nom `attr`, ou qui a un tel attribut mais sa valeur n'est pas égale à `val`
- `[attr^=val]` : sélectionne tout élément HTML ayant un attribut `attr` dont la valeur commence par `val`
- `[attr$=val]` : sélectionne tout élément HTML ayant un attribut `attr` dont la valeur se termine par `val`
- `[attr*=val]` : sélectionne tout élément HTML ayant un attribut `attr` dont la valeur contient `val`

Filtres des sélecteurs -suite-

- `:animated` : sélectionne tout élément HTML qui est en train d'être animé par jQuery
- `:button` : sélectionne les éléments `<button>` et `<input type="button">`
- `:checkbox` : sélectionne les éléments `<input type="checkbox">`
- `:checked` : sélectionne tous les éléments input qui sont cochés

Filtres des sélecteurs -suite-

- `:contains(text)` : sélectionne tous les éléments HTML qui ont un contenu texte correspondant à ce qui est placé en paramètre. Pas besoin ici d'utiliser des guillemets pour délimiter la valeur du paramètre
- `:disabled` : sélectionne les éléments désactivés
- `:empty` : sélectionne les éléments qui sont vides (pas d'éléments fils ou de contenu texte)
- `:enabled` : sélectionne tous les éléments qui ne sont pas désactivés

Filtres des sélecteurs -suite-

- `:eq(n)` : sélectionne l'élément HTML qui se trouve dans le document à la position précisée en paramètre (le premier élément a la position 0)
- `:even` : sélectionne les éléments qui se trouvent à une position paire
- `:file` : sélectionne les éléments `<input type="file">`
- `:first` : sélectionne le premier élément dans le document. C'est l'équivalent de `:eq(0)`

Filtres des sélecteurs -suite-

- `:first-child` : sélectionne l'élément qui est le premier enfant de son élément parent
- `:gt(n)` : sélectionne les éléments qui se trouvent à une position supérieure à `n`
- `:has(sel)` : sélectionne les éléments qui ont un descendant qui match avec le sélecteur `sel`
- `:header` : sélectionne les éléments `<h1>`, `<h2>`, ... `<h6>`
- `:hidden` : sélectionne les éléments cachés

Filtres des sélecteurs -suite-

- `:image` : sélectionne les éléments `<input type="image">`
(ne marche pas avec les balises ``)
- `:input` : sélectionne les éléments d'entrée par l'utilisateur :
`<input>`, `<textarea>`, `<select>` et `<button>`
- `:last` : sélectionne le dernier élément dans le document
`p:last` correspond au dernier paragraphe
- `:last-child` : sélectionne le dernier élément fils d'un élément
- `:lt(n)` : sélectionne les éléments qui ont une position inférieure à `n`

Filtres des sélecteurs -suite-

- `:not(sel)` : sélectionne les éléments qui ne correspondent pas au sélecteur `sel`
- `:nth(n)` : équivalent à `eq(n)`
- `:nth-child(n)` : sélectionne l'élément qui est le nième fils de son élément parent. `n` peut être un nombre, even ou odd
- `:odd` : sélectionne les éléments qui ont une position impaire
- `:only-child` : sélectionne les éléments qui sont les seuls fils de leur élément parent

Filtres des sélecteurs -suite-

- `:parent` : sélectionne les éléments qui sont parents.
C'est l'opposé de `:empty`
- `:password` : sélectionne les éléments `<input type="password">`
- `:radio` : sélectionne les éléments `<input type="radio">`
- `:reset` : sélectionne les éléments `<input type="reset">`
et `<button type="reset">`
- `:selected` : sélectionne les `<option>` qui ont été sélectionnés.
Pour les cases à cocher et les boutons radio, utiliser `:checked`

Filtres des sélecteurs -suite-

- `:submit` : sélectionne les éléments `<input type="submit">` et `<button type="submit">`
- `:text` : sélectionne les éléments `<input type="text">`
- `:visible` : sélectionne les éléments qui sont visibles.
C'est l'opposé de `:hidden`
- Les filtres sont utilisés de façon plus efficace lorsqu'ils sont précédés par le nom de la balise visé. Exemple : `input:radio` est plus efficace que `:radio`
- Une exception pour les filtres par id : `#id` mieux que `monForm#id`

Exemple de filtres des sélecteurs

- Que sélectionne le filtre suivant ?

```
p:nth-child(3n+1):text(JavaScript):not(has(a))
```


Exemple de filtres des sélecteurs

- Que sélectionne le filtre suivant ?

```
p:nth-child(3n+1):text(JavaScript):not(has(a))
```

Tous les paragraphes :

- qui sont le premier fils ou 3èmes suivants de leur parent,
- qui contiennent le mot JavaScript,
- et qui ne contiennent pas de balise <a>

Combinaison de sélecteurs

- Les sélecteurs peuvent être combinés en utilisant des opérateurs comme ceux de CSS 3
- $A \ B$: sélectionne tous les éléments qui matchent le sélecteur B et qui sont descendants d'éléments qui matchent le sélecteur A
- $A > B$: sélectionne tous les éléments qui matchent le sélecteur B et qui sont enfants directs d'éléments qui matchent le sélecteur A

Combinaison de sélecteurs -suite-

- $A + B$: sélectionne tous les éléments qui matchent le sélecteur B et qui suivent directement des éléments qui matchent le sélecteur A (les nœuds de type texte et les commentaires sont ignorés)
- $A \sim B$: sélectionne tous les éléments qui matchent le sélecteur B et qui sont des nœuds frères des éléments qui matchent le sélecteur A
- Exemples :
 - `"blockquote i"` // Sélectionne les `<i>` à l'intérieur de `<blockquote>`
 - `"ol > li"` // `` fils directs de ``
 - `"#output + *"` // Les nœuds suivant l'élément `#output`
 - `"div.note > h1 + p"` // `<p>` après `<h1>` à l'intérieur de `div.note`

Groupes de sélecteurs

- Ce sont des sélecteurs simples ou des combinaisons de sélecteurs séparés par des virgules

- Exemples :

`"h1, h2, h3"` // Toutes les balises : `<h1>`, `<h2>` et `<h3>`

`"#p1, #p2, #p3"` // Éléments avec les id `p1`, `p2` et `p3`

`"div.note, p.note"` // `<div>` ou `<p>` avec `class="note"`

`"body>p, div.note>p"` // `<p>` enfants de `<body>` ou `div.note`

- Impossible de placer entre parenthèses un groupe de sélecteurs et le combiner avec d'autres sélecteurs

`(h1, h2, h3) + p` // INCORRECT

`h1 + p, h2 + p, h3 + p` // écriture équivalente correcte

Plan du cours

- Sélecteurs jQuery
- Fonctions de sélection

Fonctions de sélection

- En plus des sélecteurs qui peuvent être placés sous forme de chaînes de caractères en tant que paramètres de la fonction `$(...)`, jQuery introduit des fonctions de sélection qui peuvent être appelées sur des objets jQuery
- Ces fonctions ont l'avantage de pouvoir modifier l'ensemble des éléments sélectionnés (la fonction `add(...)`, par exemple)
- La fonction `first()` retourne un **objet jQuery** enveloppant le premier élément (`last()` retourne un objet jQuery représentant le dernier elt)
- La fonction `eq(n)` retourne un objet jQuery enveloppant le nième élément

Fonctions de sélection -suite-

- La fonction `eq(n)` peut recevoir en paramètre un index négatif : pour la sélection, le parcours commence alors depuis la fin
- La sélection par le biais de ces fonctions est différent de l'accès par la notation des tableaux : cette dernière retourne un objet JavaScript et non un objet jQuery

- Exemples :

```
var paras = $("p");  
paras.first(); // retourne le premier <p>  
paras.last(); // retourne le dernier <p>  
paras.eq(1) // retourne le deuxième <p>  
paras.eq(-2) // le 2nd <p> à partir de la fin  
paras[1] // le 2nd <p> (objet JS)
```

Fonction slice()

- La fonction slice() retourne un sous-ensemble des éléments à partir d'un index donné et jusqu'à un index de fin (ou la fin de l'ensemble)
- L'élément qui correspond à l'indice de fin n'est pas inclus dans le sous-ensemble retenu

- Exemples :

```
$("p").slice(2,5); // 3ème, 4ème et 5ème <p>  
$("div").slice(-3); // les 3 derniers <div>
```


Fonction filter()

- Trois utilisations possibles de la fonction filter() :
 - On lui passe un sélecteur sous la forme d'une chaîne : elle retourne un objet jQuery contenant les éléments qui matchent le sélecteur (passé en paramètre)
 - On lui passe un objet jQuery : elle retourne un objet jQuery contenant l'intersection des deux objets jQuery (le receveur et le paramètre)
 - On lui passe une fonction qui retourne un booléen : elle appelle la fonction sur l'ensemble des objets sélectionnés. Elle retient les éléments pour lesquels la fonction retourne true. Cette fonction admet en paramètre un entier (position de l'élément)

Exemples avec la fonction filter()

- Exemples des utilisations possibles de la fonction filter() :
 - Utilisation 1 : param = sélecteur sous forme d'une chaîne
`$("div").filter(".note"); // équiv. $("div.note")`
 - Utilisation 2 : param = objet jQuery
`$("div").filter($(".note")); // $("div.note")`
 - Utilisation 3 : param = fonction callback
`$("div").filter(function(i) { // $("div:even")
 return i%2 == 0;
})`

Fonction not()

- La fonction not() a le comportement inverse de la fonction filter()
Elle peut être interprétée comme une fonction qui exclut les éléments qui matchent le paramètre
- Si un sélecteur sous forme de chaîne est passée en paramètre, la fonction not() retourne un objet jQuery contenant les éléments qui ne matchent pas le sélecteur
- Si une fonction callback est précisée comme paramètre, les éléments retenus sont ceux pour lesquels la fonction retourne false
- Exemple : tous les div sauf les deux précisés en paramètre
`$ ("div") .not ("#header, #footer") ;`

Fonction has()

- La fonction has() admet un paramètre représentant un sélecteur. Elle retourne un objet jQuery contenant les éléments ayant des éléments fils qui matchent le sélecteur
- Exemple : tous les paragraphes incluant des liens

```
$ ("p") .has ("a[href] ") ;
```

Fonction add()

- La fonction add() ajoute des éléments à un objet jQuery
- Cette fonction peut tout recevoir en paramètre, sauf des fonctions
- Nous pouvons donc créer de nouveaux éléments grâce à la fonction \$() et les passer en paramètre à add()
- La fonction add() enlève les éléments dupliqués et ordonne les éléments sélectionnés selon leur ordre dans le document
- Exemples : écritures équivalents pour sélectionner tous les div et p
`$("div,p") // Utilisation d'un groupe de sélecteurs`
`$("div").add("p") // Passer un sélecteur à add`
`$("div").add($("p")) // Passer un objet jQuery`

Fonctions de sélection sans effet de bord

- La fonction `find()` permet de rechercher des éléments fils dans une sélection d'éléments (enveloppée par l'objet jQuery sur lequel cette fonction est appelée)
- Cette fonction retourne un nouvel objet jQuery contenant les éléments fils qui matchent le sélecteur passé en paramètre
- Ne pas confondre avec la fonction `has()`, qui réduit le nombre d'éléments de la sélection (elle a donc un effet de bord)
- Exemple : retourner les `<p>` qui sont inclus dans des `<div>`
`$("div").find("p")` // Équivalent à `$("div p")`

Fonctions de sélection sans effet de bord -suite-

- La fonction `children()` retourne les éléments fils immédiats de chacun des éléments de la sélection (en les filtrant éventuellement avec un sélecteur)

- Exemple :

```
$("#header, #footer").children("span")
```

Retourner tous les `` qui sont des éléments fils des éléments ayant comme id header et footer

Équivalent à :

```
$("#header>span, #footer>span")
```

Fonctions de sélection sans effet de bord -suite-

- La fonction `contents()` a un comportement similaire à `children()`
- Elle retourne par contre tous les nœuds fils, y compris les nœuds texte (`children()` ne retourne que des éléments)
- Elle n'admet pas de sélecteur en paramètre
- Les fonctions `prev()` et `next()` retournent un élément frère précédant et succédant respectivement à un élément de la sélection. Ces fonctions peuvent filtrer en fonction d'un sélecteur précisé comme paramètre

Fonctions de sélection sans effet de bord -suite-

- jQuery fournit aussi les fonctions : `prevAll()`, `nextAll()` et `siblings()`
Elles retournent les éléments frères (avant, après et tous respect.)
- Les fonctions `prevUntil()` et `nextUntil()` retournent les éléments frères jusqu'à un élément qui match le sélecteur passé en paramètre de ces fonctions
- Si aucun paramètre n'est précisé, `prevUntil()` et `nextUntil()` retournent le même résultat que `prevAll()` et `nextAll()`

Fonctions de sélection sans effet de bord -suite-

- La fonction `parent()` retourne l'élément parent :
`$("li").parent()` // éléments ayant des `` comme fils
- La fonction `parents()` retourne les ancêtres (jusqu'à `<html>`) de chacun des éléments sélectionnés
- Ces deux fonctions peuvent recevoir un sélecteur comme paramètre :
`$("a[href]").parents("p")` // les `<p>` qui contiennent des liens
- Autres fonctions possibles : `parentsUntil()` et `closest()`

Références bibliographiques

- Bear Bibeault et Yehuda Katz. « **jQuery in Action** ». Second Edition. Manning Publications Co. Juin 2010.
- David Flanagan. « **jQuery – Pocket Reference** ». First Edition. Éditions O'Reilly. Décembre 2010.
- Anthony T. Holdener III. « **Ajax: The Definitive Guide – Interactive Applications for the Web** ». Éditions O'Reilly. Janvier 2008.
- Documentation en ligne (très pratique) : <http://api.jquery.com/>

Questions

