



jQuery

MIMIAGE 2008/2009

Adil El Ghali



Introduction

- Framework Javascript très populaire
 - Minimalisme et élégance
 - Largement utilisé: BBC, Digg, Intel, MSNBC, Technorati, ...
- Permet de prendre de la hauteur par rapport aux différences des navigateurs



Plan

- Premiers pas
- Concepts de base
 - *Sélecteurs, Événements & DOM*
 - *AJAX*
 - *Effets*
 - *Plugins*
- Ressources



Premiers pas

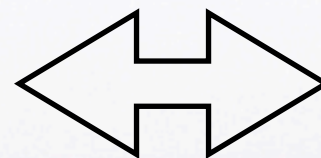
- http://docs.jquery.com/Downloading_jQuery

- inclure jquery.js dans votre HTML:

```
<script type="text/javascript" src="jquery.js"></script>
```

- associer une fonction au chargement de la page:

```
$(document).ready(function(){  
    // Votre code  
});
```



```
window.onload = function(){  
    // Votre code  
}
```




Premiers pas

- Ajouter/enlever des styles:

```
$("#h1").addClass("test");  
$("#h1").removeClass("test");
```

```
<style type="text/css">  
  h1.test { font-color: #ff0000; }  
</style>
```

- Effets:

```
$("#a").click(function(event){  
  event.preventDefault();  
  $(this).hide("slow");  
});
```

fait disparaître les liens après un click



Concepts de base (I)

Références aux éléments

- Tout élément du document peut être désigné par :

`$("expression").filtre()`

- Combinaison de chemin XPath et de CSS
- Utilisation de fonctions de filtrage jQuery



Concepts de base (2)

Chaînage

- Tout appel jQuery renvoie l'objet vous permettant d'appliquer une chaîne de méthodes :

```
$("exp1").function1().function2("exp2")
```



Concepts de base (3)

Callback

- C'est une fonction qui est passée en paramètre à une autre fonction et qui est exécutée après la terminaison de sa fonction mère :

`$.get("toto.html", callBack)`

- `callBack` est exécutée à la fin de `get`



Sélecteurs

- Ensemble d'expressions permettant de désigner les éléments du document:
- Plusieurs types:
 - Basic, Hiérarchie, Attribut, Filtres
 - <http://docs.jquery.com/Selectors>
- Tous les sélecteurs retournent un tableau d'éléments



Sélecteurs: Basic

- `#id` : retourne l'élément ayant l'identifiant `#id`.
- `element` : tous les éléments ayant le nom `element`.
- `.class` : tous les éléments d'une classe.
- `*` : tous les éléments du documents.
- `selector1, selector2, selectorN` : tous les éléments correspondant à un des sélecteurs.

```
$("#div,span,p.myClass").css("border","3px solid red");
```

div

p class="myClass"

p class="notMyClass"

span



Sélecteurs: Hiérarchie

- **ancestor descendant** : tous les éléments descendants correspondants à “descendant” de l’élément “ancestor”.
- **parent > child** : tous les fils correspondants à “child” des éléments spécifiés par “parent”.
- **prev + next** : tous les éléments correspondants à “next” après les éléments spécifiés par “prev”.
- **prev ~ siblings** : tous les frères des éléments spécifiés par "prev" correspondants au sélecteur "siblings".



Sélecteurs: Filtres

- **:first/:last** le premier/dernier élément de la sélection.
- **:not(sel)** les éléments ne correspondant pas à “sel”.
- **:even/:odd** les éléments pair/impair
- **:eq(N)** l'élément à la position “N”.
- **:gt(N)/:lt(N)** les éléments ayant un index sup/inf à “N”.
- **:header** tous les éléments h1, h2, ...
- **:animated** tous les éléments animés.



Sélecteurs: Filtres

- Contenu:
 - `:contains(text)` les éléments contenant “text”.
 - `:empty` les éléments vide (pas de fils).
 - `:has(sel)` les éléments dont un des fils correspond à “sel”.
 - `:parent` tous les parents.
- Visibilité:
 - `:hidden` tous les éléments cachés.
 - `:visible` tous les éléments visibles.



Sélecteurs: Attributs

- **[attr]** les éléments ayant un attribut “attr”.
- **[attr=val]** les éléments ayant un attribut “attr” de valeur “val”.
- **[attributeX=val]** $X=!/^\$/*$ pour resp. le attribut différent/commençant/finissant/contenant la valeur “val”
- **[attrFilter1][attrFilter2][attrFilterN]** tous les éléments correspondant à un des friltres.



Sélecteurs: Forms

- **:input** tous les éléments de saisie: textarea, select et button.
- **:text** tous les champs de texte.
- **:password** les champs de mots de passe.
- **:submit** les éléments d'envois.
- **:button** les boutons.
- **:hidden** les éléments cachés.



Événements

- Toutes les méthodes associées aux événements retournent un objet jQuery
- `ready(fn)` définit la fonction à exécuter quand le document est chargé (à la fin de la construction de l'arbre DOM).
- `live(type, fn)` associe un gestionnaire à un type d'événements, pour toutes les instances survenant après l'appel de `live`.
- `die(type, fn)` l'inverse `live`, supprime le gestionnaire pour les événements survenant après l'appel de `die`.
- **Types d'événements:** `blur`, `focus`, `load`, `resize`, `scroll`, `unload`, `error`, `beforeunload`, `click`, `dblclick`, `mousedown`, `mouseup`, `mousemove`, `mouseover`, `mouseout`, `mouseenter`, `mouseleave`, `change`, `select`, `submit`, `keydown`, `keypress`, `keyup`



Evénements

- `bind(type, data, fn)` associe un gestionnaire à un type d'événement sur les éléments correspondant à un sélecteur , des données peuvent être passées au gestionnaire.
- `trigger(event, data)` lève un événement `event`.
- `triggerHandler(event, data)` lève un événement sur le premier élément du tableau jQuery (ne permet pas le chaînage) SANS exécuter les actions par défaut du navigateur.
- `unbind(type, fn)` inverse de bind, supprime l'association entre un événement et son gestionnaire.



Événements

- Gestion des événements par défaut:
- `event(fn)` associe un gestionnaire à l'événement `event` sur les éléments correspondant à un sélecteur.
 - `event` : mousedown, mouseenter, mouseleave, mousemove, mouseout, mouseover, mouseup, resize, scroll, load, unload
- `event()` lève un événement `event` pour les éléments correspondant à un sélecteur.
 - `event` : blur, change, click, dblclick, error, focus, keydown, keypress, keyup, select, submit() mousedown, mouseenter, mouseleave, mousemove, mouseout, mouseover, mouseup, resize, scroll, load, unload



Evénements

<http://docs.jquery.com/Events>



DOM

- Fonctions de test (renvoient un booléen):
- `hasClass(class)` vérifie si la sélection courante contient un élément ayant la classe “`class`”
- `is(expr)` vérifie si un élément de la sélection courante correspond au sélecteur défini par l’expression “`expr`”.



DOM

- **Filtrage**

- `eq(N)` réduit la sélection courante à l'élément à la position N.
- `filter(expr/fn)` supprime de la sélection les éléments ne correspondant pas à `expr` ou filtrés par la fonction `fn`.
- `not(expr)` supprime de la sélection les éléments correspondant à `expr`.
- `slice(start, end)` réduit la sélection aux éléments entre les positions `start` et `end`. Si `end` est omis de `start` à la fin.



DOM

- `add(expr)` ajoute les éléments correspondant à `expr` à la sélection.
- `children(expr)` remplace la sélection par l'ensemble des fils directs de ses éléments .
- `contents()` renvoie toutes les feuilles de la sélection.
- `find(expr)` trouve tous les éléments correspondants à `expr` dans le sous-arbre de la sélection.
- `next/prev(expr)` renvoie l'ensemble contenant le frère suivant/précédent de chaque élément de la sélection.
- `nextAll/prevAll(expr)` renvoie tous les frères suivants/précédents de l'élément courant.
- `parent(expr)` l'ensemble des parents directs des éléments de la sélection.
- `parents(expr)` l'ensemble des parents des éléments de la sélection.
- `siblings(expr)` l'ensemble contenant tous les frères des éléments de la sélection correspondant à `expr` .



AJAX

- `jQuery.ajax(options)` fonction de bas niveau d'appel AJAX, spécialisé par `$.get` et `$.post`, permet des manipulations avancées sur l'objet XHR.
- `jQuery.get(url, data, callback, type)` charge une page distante d'url avec un HTTP GET. `data` est un ensemble {(cle, valeur)}
- `jQuerygetJSON(url, data, callback)` charge des données JSON d'url avec un HTTP GET.
- `jQuery.getScript(url, callback)` charge et exécute un script local (même domaine que la page courante) avec un HTTP GET.
- `jQuery.post(url, data, callback, type)` charge une page distante avec un HTTP POST.
- `load(url, data, callback)` charge un fichier HTML et l'injecte dans le DOM.



AJAX: Événements

- **ajaxComplete(callback)** définit une fonction à exécuter à la complétion de tous les appels AJAX.
- **ajaxError(callback)** définit une fonction à exécuter lors d'une erreur d'un appel AJAX.
- **ajaxSend(callback)** définit une fonction à exécuter avant l'envoi des requêtes AJAX.
- **ajaxStart(callback)** définit une fonction à exécuter au lancement d'une requête AJAX.
- **ajaxStop(callback)** définit une fonction à exécuter à la fin de toutes les requêtes AJAX.
- **ajaxSuccess(callback)** définit une fonction à exécuter en cas de terminaison avec succès d'un appel AJAX.



AJAX: Exemple

- Authentication

```
$.get("/authServlet", { user: "joe", passwd: "xxxx" },  
function(data){  
  if (data == true)  
    alert("User authenticated");  
  else  
    alert("Bad login or password");  
});
```



AJAX: Exemple

- Affichage des 6 premières photos sur Flickr ayant le tag “Paris”

```
$.getJSON("http://api.flickr.com/services/feeds/photos_public.gne?tags=Paris&tagmode=any&format=json&jsoncallback=?",  
function(data){  
  $.each(data.items, function(i,item){  
    $("<img/>").attr("src", item.media.m).appendTo("#images");  
    if ( i == 5 ) return false;  
  });  
});
```





```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <script src="http://code.jquery.com/jquery-latest.js"></script>
  <script>
    $(document).ready(function(){
      $.getJSON("http://api.flickr.com/services/feeds/photos_public.gne?tags=Paris&...
tagmode=any&format=json&jsoncallback=?",
        function(data){
          $.each(data.items, function(i,item){
            $("<img/>").attr("src", item.media.m).appendTo("#images");
            if ( i == 5 ) return false;
          });
        });
    });
  </script>
  <style>img{ height: 100px; float: left; }</style>
</head>
<body>
  <div id="images">
  </div>
</body>
</html>
```



Animation et effets

- Fonctions de base
- [`show\(speed, callback\)`](#) rend visible les éléments de la sélection s'ils sont cachés.
- [`hide\(speed, callback\)`](#) cache les éléments de la sélection s'ils sont visibles.
- [`toggle\(speed, callback\)`](#) bascule le mode d'affichage des éléments de la sélection.
- Il est possible de spécifier la vitesse d'apparition ("slow", "normal" ou "fast". Ou bien la durée en *ms*) et une fonction à exécuter à la fin de l'opération.



Animation et effets

- `animate(params, duration, easing, callback)` permet de réaliser des animations paramétrées. `params` est une ensemble `{(cle, valeur)}` qui encode les propriétés à appliquer aux éléments de la sélection.

```
// Using multiple unit types within one animation.
$("#go").click(function(){
  $("#block").animate({
    width: "70%",
    opacity: 0.4,
    marginLeft: "0.6in",
    fontSize: "3em",
    borderWidth: "10px"
  }, 1500 );
});
```

- `stop(clearQueue, gotoEnd)` permet d'arrêter toutes les animations sur la sélection.



Plugins

- Une des grandes forces de jQuery est son système de plugins
- Il y en a beaucoup
 - <http://docs.jquery.com/Plugins>
- C'est facile à écrire



Plugins

- **Ecrire un plugin:**

- Choisir un nom

- plugin `toto`  fichier javascript: `jquery.toto.js`

- Ajouter des fonctions

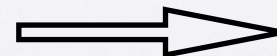
```
jQuery.fn.toto = function() {  
  // do something  
};
```



appelée avec `$(...).toto();`

- Définir les paramètres par défaut

```
jQuery.fn.toto = function(options) {  
  var settings = jQuery.extend({  
    value: 5, foo: "toto", bar: 655  
  }, options);  
};
```



```
$("...").toto();
```

```
$("...").toto({ value: 123, bar: 9 });
```

- Documenter le plugins



Ressources

- http://docs.jquery.com/Main_Page
- http://docs.jquery.com/Tutorials#Tutoriaux_en_Fran.C3.A7ais
- http://www.digital-web.com/articles/jquery_crash_course
- <http://www.learningjquery.com/>
- <http://video.google.com/videoplay?docid=-474821803269194441>