

Introduction à jQuery

Tutoriaux jQuery - Décembre 2008

Clément Delmas (clementdelmas.fr)

Prérequis techniques conseillés

- Avant de lire ce tutorial, il est conseillé de connaître :
 - le code HTML et le DOM
 - le CSS et les sélecteurs
 - le Javascript et les fonctions basiques (variables, fonctions, boucles, etc.)

Présentation des librairies Javascript

- Ensemble de fonctionnalités disponibles et prêtes à l'usage
- Compatibilité avec l'ensemble des navigateurs
- Flexibilité et simplicité du code
- Intégration de plugins

Quelques librairies Javascript

- jQuery (<http://jquery.com/>)
- Mootools (<http://www.mootools.net/>)
- Prototype (<http://prototypejs.org/>)
- Script.aculo.us (<http://script.aculo.us/>)
- Dojo Toolkit (<http://dojotoolkit.org/>)
- Yahoo! User Interface Library (<http://developer.yahoo.com/yui/>)

Pourquoi choisir jQuery ?

- Basée sur des sélecteurs CSS
- Simple à comprendre et à apprendre
- Ne modifie pas l'espace de travail et s'interface avec d'autres librairies
- Javascript non-intrusif
- Multitude de plugins disponibles (<http://plugins.jquery.com/>)

La fonction jQuery()

- `jQuery('monSelecteurCSS')`

va sélectionner les objets html qui correspondent à `monSelecteurCSS` et les retourne sous forme de collection directement exploitable.
Sélecteurs CSS : `#monID`, `.maClasse`, `:monType`, `monElementHtml`

- `jQuery()` peut-être remplacé par `$()`
`$('#monID')`, `$('.maClasse')`, `$('input:radio')`, `$('#monElementHtml')`

Exemples de sélecteurs

- `$('#header')` > sélectionne l'objet html ayant pour id "header"
- `$('.container')` > sélectionne tous les objets ayant la classe "container"
- `$(a)` > sélectionne l'ensemble des liens de la page
- `$('#maListe li')` > sélectionne l'ensemble des 'li' de la liste #maListe
- `$(input:reset)` > sélectionne tous les boutons de type "reset"

Liste des sélecteurs

SELECTORS

#id, tag, .class, *	E[@attr]	
elm1, elm2, elmN	E[@attr=val]	
ancestor descendant	E[@attr^=val] (begins)	
parent > child	E[@attr\$=val] (ends)	
parent/child	E[@attr*=val] (contains)	
prev + next	E[@attr=val][@attr=val] (both)	
prev ~ siblings		
	:nth-child(index)	
:first	:first-child	
:last	:last-child	
:not(selector)	:only-child	:input
:even		:text
:odd	:enabled	:password
:eq(index)	:disabled	:radio
:gt(index)	:checked	:checkbox
:lt(index)	:selected	:submit
		:image
:contains(text)	:hidden	:reset
:empty	:visible	:button
:has(selector)	:header	:file
:parent	:animated	:hidden

- liste de l'ensemble des sélecteurs possibles avec jQuery 1.2
- Extrait de jQuery CheatSheet (par Color Charge)

<http://colorcharge.com/jquery/>

Manipuler une collection d'objets

- `$(‘p’).addClass(‘important’)` > ajoute la classe important à l'ensemble des paragraphes
- `$(‘img#photo’).attr(‘src’,‘img/photo.jpg’)` > change l'attribut “src” de l'image ayant l'id photo par “img/photo.jpg”
- `$(‘a.plus’).html(“en savoir plus”)` > modifie le code html des liens ayant la classe plus par “en savoir plus”
- `$(‘#container’).css(“background-color”,“white”)` > change la couleur de fond du #container par la couleur blanche

Récupérer des valeurs

- Même si les sélecteurs jQuery retournent des collection d'objets, beaucoup de méthodes (fonctions) s'appliquent directement au premier objet retourné. Elles permettent de récupérer facilement des attributs d'éléments html.

```
var hauteurContainer = $('#container').height();  
var srcPhoto = $('img.photo').attr('src');  
var contenuLastP = $('p:last').html();
```

Liste des manipulateurs

CSS	ATTRIBUTES
<code>.css(name, value)</code> <code>.css(properties)</code>	<code>.attr(name)</code> <code>.attr(properties)</code> <code>.attr(key, value)</code> <code>.attr(key, function)</code> <code>.removeAttr(name)</code>
<code>.height(value)</code> <code>.width(value)</code>	
<code>.addClass(class)</code> <code>.removeClass(class)</code> <code>.toggleClass(class)</code> <code>.offset()</code>	
	HTML
	<code>.html()</code> <code>.html(value)</code> <code>.text()</code> , <code>.text(value)</code> <code>.val(value)</code>

- liste de l'ensemble des manipulateurs possibles avec jQuery 1.2
- Extrait de jQuery CheatSheet (par Color Charge)

<http://colorcharge.com/jquery/>

Ajouter des effets graphiques

- jQuery intègre une série complète d'effets graphiques qui permettent d'animer une page html très facilement

`$('#p.secret').hide()` > masque le paragraphe ayant la classe secret

`$('#info').fadeIn()` > affiche le div ayant l'id info en fondu

`$('.menu').slideUp("slow")` > masque le menu en glissant lentement

`$('.foo').toggle('slide')` > affiche ou masque le .foo en glissant, en fonction de son état (on l'affiche s'il est masqué, et vice et versa)

Animations avancées

- jQuery possède aussi une fonction **animate**, qui permet de faire des animations plus complexes.

```
$("#block").animate({  
  // Change la taille du bloc et son opacité  
  width: "70%", opacity: 0.4,  
  // Ajoute une marge gauche et change la taille de la police  
  marginLeft: "13px", fontSize: "3em" },  
  // La durée de l'animation sera de 3 secondes  
  { duration: 3000 }  
);
```

Liste des effets

CORE UI EFFECTS

SHOW / HIDE

`.show()`
`.show(speed, callback)`
`.hide()`
`.hide(speed, callback)`
`.toggle()`

SLIDE (speed, callback)

`.slideDown(s, c)`
`.slideUp(s, c)`
`.slideToggle(s, c)`

ANIMATE

`.stop()`
`.queue()`,
`.queue(callback)`,
`.queue(queue)`
`.dequeue()`
`.animate(params, duration, easing, callback)`
`.animate(params, options)`

FADE

`.fadeIn(speed, callback)`
`.fadeOut(speed, callback)`
`.fadeTo(speed, opacity, callback)`

- liste de l'ensemble des effets qui permettent de modifier les éléments DOM (compatibles avec jQuery 1.2)
- Extrait de jQuery CheatSheet (par Color Charge)

<http://colorcharge.com/jquery/>

Traverser le DOM

- jQuery permet de naviguer facilement à travers les différents objets html du DOM

`$('#div.section').next()` > retourne l'élément suivant

`$('#div.section').prev()` > retourne l'élément précédent

`$('#div.section').parent()` > retourne l'élément parent

`$('##sidebar').children('ul')` > retourne l'ensemble des listes enfant

`$('#a.more').prev('a')` > retourne le lien précédent

Liste des méthodes de navigation

TRaversING

FILTER

.hasClass(class)
.filter(expr)
.filter(fn)
.is(expr)
.map(callback)
.not(expr)
.slice(start, end)

FIND (expr)

.add(e)
.children(e), .siblings(e)
.contents()
.find(e)
.next(e), .nextAll(expr)
.parent(e), .parents(e)
.prev(e), .prevAll(e)

ACCESS

.each(callback)
.size()
.length
.get()
.get(index)
.index(subject)

CHAIN

.andSelf()
.end()

- liste de l'ensemble des méthodes qui permettent de traverser le DOM (compatibles avec jQuery 1.2)
- Extrait de jQuery CheatSheet (par Color Charge)

<http://colorcharge.com/jquery/>

Création d'événements

- jQuery permet de créer des déclencheurs d'événements à des éléments de la page, sans se soucier des problèmes de compatibilité des navigateurs.

```
$('#a.changeColor').click(function(){  
    $(this).css('color','orange');  
});
```

Liste des événements

EVENTS

HANDLERS

.bind(type, data, fn)
.one(type, data, fn)
.trigger(type, data)
.triggerHandler(type, data)
.unbind(type, data)

MOUSE

.mousedown(fn)
.mousemove(fn)
.mouseout(fn)
.mouseover(fn)
.mouseup(fn)

WINDOW

.load(fn) .scroll(fn)
.resize(fn)

ERROR

.error()
.error(fn)

KEYBOARD

.keydown()
.keydown(fn)
.keypress()
.keypress(fn)
.keyup()
.keyup(fn)

PAGE

.ready(fn)

INTERACTION

.hover(fnIN, fnOUT)
.toggle(fnIN, fnOUT)
.blur() .blur(fn)
.change() .change(fn)
.click() .click(fn)
.dblclick() .dblclick(fn)
.focus() .focus(fn)
.select() .select(fn)
.submit() .submit(fn)
.unload() .unload(fn)
.unblur() .unblur(fn)

- liste de l'ensemble des événements qui compatibles avec jQuery 1.2
- Extrait de jQuery CheatSheet (par Color Charge)

<http://colorcharge.com/jquery/>

jQuery et le Javascript non-intrusif

- jQuery est une librairie qui permet de faire du Javascript non-intrusif

Cela signifie que le code JS n'est exécuté que lorsque la page est totalement chargée.

```
$(document).ready(function(){  
    alert("Le document est prêt !");  
});
```

Réactions en chaîne

- La plupart des fonctions jQuery peuvent être exécutées à la chaîne, c'est à dire, les unes après les autres, sans répéter l'élément auxquelles elles s'appliquent.

```
$('#div.aMasquer').removeClass('foo').children('p').hide();
```

Dans l'exemple ci dessus, on ajoute la classe "foo" au div ayant la classe aMasquer, et on masque tous ses paragraphes enfants.

Exemple de chaînes folles

- `$('#form#login')`
 - `// Masque l'ensemble des labels ayant la classe "optional"`
 - `.find('label.optional').hide().end()`
 - `// Ajoute une bordure rouge d'1px aux input de type password`
 - `.find('input:password').css('border', '1px solid red').end()`
 - `// Ajoute une fonction de validation de l'envoi au formulaire`
 - `.submit(function(){`
 - `return confirm('Êtes vous sûr de vouloir valider le formulaire ?');`
 - `});`

Exemple repris de : <http://www.ibm.com/developerworks/library/x-ajaxjquery.html>

Les callbacks

- Les callbacks sont des fonctions qui sont appelées une fois un événement déclenché ou une action faite. Elles sont généralement utilisés en Ajax, mais peuvent être intégrés à la fin d'animations graphiques.

```
$('#grow').animate(  
    { height: 500, width: 500 }, "slow",  
    function callback(){ alert('Croissance terminée !'); }  
);
```

Exemple repris de : <http://www.ibm.com/developerworks/library/x-ajaxjquery.html>

Copyrights

- [jQuery in 15 minutes](#) - Torchbox
- [Color Charge jQuery CheatSheet](#)
- [Simplify Ajax development with jQuery](#)

Plus d'informations...



- Sur le site officiel : <http://jquery.com/>

Vous trouverez aussi la liste des fonctions, des plugins disponibles, et plein d'autres ressources....

Merci