

Jquery : bibliothèque JavaScript

Faites davantage avec moins de code!

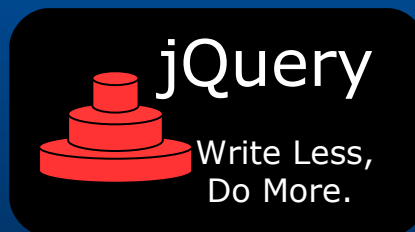
GTI780 / MTI780 - Sujets spéciaux en TI

Le Web 2.0 : concepts et outils

École de technologie supérieure

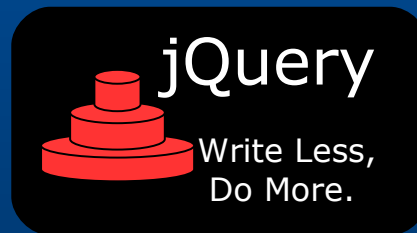
par

Claude Coulombe



jQuery - *Write less, do more.*

jQuery est une bibliothèque JavaScript qui retient l'attention en raison de sa syntaxe astucieuse, de ses performances, de sa compacité et de son approche modulaire à base de plugiciels.



<http://www.jQuery.com>

jQuery – Concepts de base

- Enveloppe logicielle (Wrapper) jQuery ou \$
- Emploi intelligent de sélecteurs basés sur CSS 3 pour sélectionner des objets DOM dans une page web
- Retour par les fonctions jQuery de tableaux (Array) d'objets
- Puissantes fonctions de manipulation d'objets du DOM
- Opération sur l'ensemble des objets d'un tableau d'objets sans utiliser de boucle explicite
- Chaînage des opérations
- http://docs.jquery.com/Main_Page

jQuery – Sélecteurs d'objets DOM

- Sélection par identifiant (id)
`jQuery("#monId")`
- Sélection par classe (CSS class)
`jQuery(".maClasse")`
- Sélection par balise (tag)
`jQuery("div")`
- Documentation
<http://docs.jquery.com/Selectors>

jQuery – Filtrés d'objets DOM

- Emploi de filtres pour réduire les tableaux d'objets (opérateur ":")

```
jQuery("div:eq(1)")
```

```
jQuery("span:not(#monId)")
```

```
jQuery("p:first").hasClass("myClass")
```

jQuery – Manipulation d'objets DOM

- Création d'un fragment de HTML et ajout à la fin du document

```
jQuery("<p>Qu'est-ce que jQuery?</p>").appendTo("body")
```

- Changement de CSS

```
jQuery("#monId").toggleClass("surbrillance")
```

```
jQuery("tr").toggleClass("couleurLigne")
```

- Changement de texte

```
jQuery(".classeTexte").text("texte de remplacement")
```

- Documentation

```
http://docs.jquery.com/Manipulation
```

jQuery – Chaînage des opérations

- Chaque opération jQuery retourne une référence à un tableau d'objets auquel on peut appliquer une autre opération, voir un enchaînement d'opérations.
- Par exemple

```
jQuery("#monId").addClass("maClasseCSS");
```

```
jQuery("#monId").show();
```

Devient

```
jQuery("#monId").addClass("maClasseCSS").show();
```

jQuery – Boucle implicite

- Certaines instructions de jQuery réalisent des boucles implicites sur des ensembles de données.

- Par exemple :

```
jQuery.each([0,1,2,3,4], function() {  
    document.write(this + 1);  
});
```

Écrit :

12345

jQuery – Initialisation au chargement

- Au moment du chargement d'une page jQuery fournit le contrôleur *ready*. Le contrôleur *ready* initie l'exécution du code lorsque le document est chargé mais sans attendre le chargement des images et l'exécution du contrôleur *onload*.

```
jQuery(document).ready( function() {  
    // code à exécuter au moment du chargement  
});
```

Note : Quand les contrôleurs *ready* et *onload* sont utilisés dans la même page, les 2 contrôleurs vont s'exécuter, le contrôleur *ready* s'exécutant avant le contrôleur *onload*.

jQuery – Ajax

- jQuery fournit plusieurs fonctions Ajax pour différents besoins dont `load(...)` et `ajax(...)`

La fonction `load(...)`

```
jQuery('#nombre').load('/ServeurAjaxSimple/reponse');
```

Ou la fonction `ajax(...)`

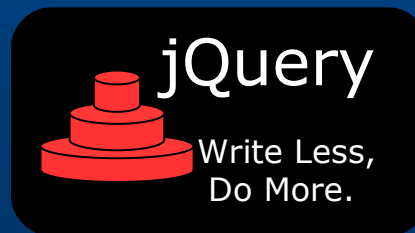
```
jQuery.ajax( { type: "POST",  
              url: "/ServeurAjaxSimple/reponse",  
              data: "param1=valeur1&param2=valeur2",  
              success: function(rep){ /* callback */  
                  alert( "Donnée traitées: " + rep ); }  
            });
```

jQuery - Avantages



jQuery - Avantages

- Assure la compatibilité entre les différents navigateurs
- Syntaxe efficace, peu verbeuse, instruction puissante*
- Bibliothèque compacte (version de base 15 Ko)
- Emploi intelligent de sélecteurs basés sur CSS 3 pour sélectionner des objets DOM dans une page web
- Opération sur l'ensemble des objets d'un tableau d'objets sans utiliser de boucle explicite
- Chaînage des opérations
- Structure modulaire et extensible par plugiciels (Plugins)



jQuery - inconvénients

- Les experts jQuery et JS sont rares...
- JS trop permissif et difficile à déboguer
- JS n'a pas été conçu pour de gros logiciels
- Pas de bon support par les IDEs
- Même modularisé, le code peut devenir très gros

Ressources - Livres

jQuery in Action

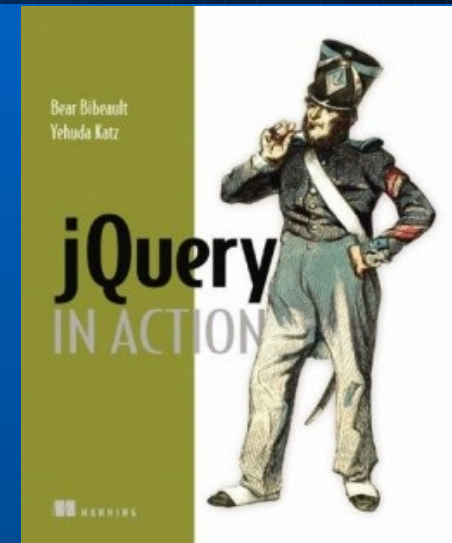
par Bear Bibeault, Yehuda Katz

376 pages

Manning Publications

(7 février, 2008)

www.manning.com/bibeault/



Learning jQuery: Better Interaction Design and Web Development with Simple JavaScript Techniques

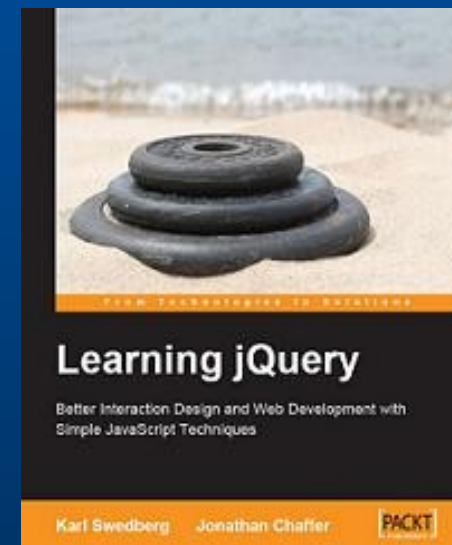
par Karl Swedberg, Jonathan Chaffer

380 pages

Packt Publishing

(29 juin, 2007)

www.packtpub.com/jquery/book#indetail



Questions

