

**Jquery**  
(une, parmi d'autres, bibliothèque JavaScript,  
cross-browser)



Les « méthodes » de la bibliothèque jQuery s'utilise au travers de l'objet jQuery créé par l'inclusion du fichier jquery.js (jquery-1.7.1.min.js)

L'objet jQuery s'utilise selon deux modes :

> en tant qu'objet : il possède des propriétés et des méthodes qui s'utiliseront sous la forme

*jQuery.property*  
*jQuery.method()*

>en tant que fonction, utilisé sous la forme

*jQuery(selector, context)*

Le plus souvent, l'objet jQuery est utilisé via un raccourci :

La variable \$

```
var $ = jQuery;
```

```
$.property
```

```
$.method()
```

```
$(selector,context).method1( )
```

```
....
```

*Selecteur magique \$('anything'):*

*\$(selector [,context])* permet d'accéder aux éléments HTML de la page (de *Document*).

L'objet (javascript) retourné (ensemble de noeuds de la page) possède des propriétés et fonctions pouvant être utilisés pour « travailler » ces différents éléments (i.e DHTML).

```
<script src=jquery.js></script>
```

```
....
```

```
<p> Paragraphe1 </p>
```

```
<p> paragraphe deux </p>
```

```
....
```

```
<script>
```

```
$("#p").css({ background: "black", color : "white"});
```

```
</script>
```

# Selecteur name, class, id, ...

- `X[name="____"]` (où X est une « balise », *name* un attribut)
- `X[id="____"][...]`
- `X.className` ou `X[class="className"]`
- `X#ident` ou `X[id="ident"]`
- *Etc, il existe de (très) nombreuses variantes permettant d'affiner la selection (de petits airs de Xpath) :*

`$(document), $('body'), $(h1:first), $(h1:eq(3)), $  
(h1:contains(« test »))'`

`$('#truc'), $('#truc').parent(), $('#truc').find('p'), $('#truc').next()`

**Le Paramètre *context* permet d'exprimer un noeud contexte de selection (sous-entendu différent du document lui-même)**

Exemple :

```
<script src=jquery.js></script>
```

```
....
```

```
<div id=d1>
```

```
<p> Paragraphe1 </p>
```

```
<p> paragraphe deux </p>
```

```
</div>
```

```
<div id=d2>
```

```
<p> Paragraphe autre </p>
```

```
<p> Para chute </p>
```

```
....
```

```
<script>
```

```
$("p",document.getElementById("d2")).css({background:"black", color:"white"});
```

```
</script>
```

**– testjquery2**

Consécutivement à la selection, on peut enchaîner également sur une action (fonction associée à un traitement) particulière

```
$ ("li:odd") .prepend ( '<span>Changed</span>' ) .css ( {background: "red" } )
```

The diagram illustrates the transformation of an HTML list through two steps:

**Initial HTML:**

```
<ul>  
  <li>  
    First item  
  </li>  
  <li>  
    Second item  
  </li>  
  <li>  
    Third item  
  </li>  
</ul>
```

**Step 1 (after .prepend):**

```
<ul>  
  <li>  
    <span>Changed</span>  
    First item  
  </li>  
  <li>  
    Second item  
  </li>  
  <li>  
    <span>Changed</span>  
    Third item  
  </li>  
</ul>
```

**Step 2 (after .css):**

```
<ul>  
  <li  
    style="background:red;">  
    <span>Changed</span>  
    First item  
  </li>  
  <li>  
    Second item  
  </li>  
  <li  
    style="background:red;">  
    <span>Changed</span>  
    Third item  
  </li>  
</ul>
```

### Gestion de Contenus Texte:

- .html() :: (); renvoie le contenu ; ('blablabla') remplace le contenu par
- .append() :: ajoute du contenu html à la fin du contenu existant
- .prepend() :: ajoute du contenu html en tête du contenu existant
- .remove() :: supprime la balise sélectionnée
- .empty() :: vide la balise sélectionnée

## Gérer le retour (collection de noeuds - nodelist) d'une sélection

- La méthode *each(callback)* : cette méthode parcourt les différents éléments et applique sur chacun d'entre-eux la fonction *callback(index)* . Le paramètre *index* indique le rang de l'élément courant dans la liste. La variable *this* fait référence, dans un appel de la fonction *callback*, à l'élément DOM concerné.
- Exemple :

```
<script src=jquery.js></script>
```

```
<div>
```

```
  <p>P1</p>
```

```
  <p>P2</p>
```

```
  <p id="p3">p3</p>
```

```
  ...
```

```
</div>
```

```
<script>
```

```
$("#p").each(function (index)
```

```
{
```

```
  if (this.id!="p3") $(this).css({color: "red"});
```

```
});
```

```
</script>
```



# Dynamique événementielle : bind(eventName, callback)

```
<script src="jquery.js"> </script>
```

```
...
```

```
<p id="p1">P1 </p>
```

```
<p id="p2">P2 </p>
```

```
...
```

```
<script>
```

```
var txt="message1";
```

```
$("p#p1").bind("click", function (event)  
    { alert(txt); });
```

```
txt = "message 2";
```

```
$("p#p2").bind("click", function (event)  
    { alert(txt); });
```

```
</script>
```

Que se passe-t-il au « click » sur les paragraphes de cette page ?

*eventName* : load, resize, mousedown, mouseup, click, dblclick,  
mouseover, mouseout, scroll, keydown, keypress, focus,  
change, select, ...

Propriété de *event* : type, timestamp, target, keyCode, screenX,  
screenY ....

--testjquery3

ou passer par des fonctions:

## Exemple:

```
$( "span.none" ).click(  
    // les span de class none  
  
    function() {  
        // les suivants de type checkbox  
  
        $(this).siblings( ":checkbox" ).removeAttr( "checked" );  
  
    }  
)  
  
-----  
  
$( "span.all" ).click(  
    // les span de class all  
  
    function() {  
        // les suivants de type checkbox  
  
        $(this).siblings( ":checkbox" ).attr( "checked", "checked" );  
  
    }  
)
```

# Requêtes Ajax:

JQuery possède toute une panoplie de fonctions permettant de simplifier les requêtes Ajax

La plus simple :

```
$ ( '#maDiv' ) .load ( 'page.html' ) ;
```

Ou encore :

**`$.get( . . . )` avec les Paramètres:**

url (String): URI de la page à charger

params (Map): (optionnel) paires de clé/valeur qui seront envoyées au serveur.

callback (Fonction): (optionnel) fonction qui sera exécutée quand les données sera chargées.

type (String): (optionnel) format des données renvoyées à la fonction de callback

```
$.get("test.php",  
    { name: "John", time: "2pm" },  
    function(data){  
        alert("Data Loaded: " + data);  
    }  
);
```

## Requêtes Ajax avec \$.ajax(options)

Les options permettent de configurer la requête Ajax :

options.url :

options.data : données transmises aux serveurs (paramètres)

options.type : GET (par défaut) ou POST

options.datatype : "text", "xml", "html", "json"

options.complete :

fonction de la forme complete(xhr, textStatus)

Xhr correspond à la requête XMLHttpRequest créée

options.error :

Fonction de la forme error(xhr, textStatus)

....

```
<html>
<head>
<script src="jquery.js"></script>
</head>
<body>
</body>
<script type="text/javascript">
    $.ajax({
    url : "hello.txt" ,
    complete: function(xhr,textStatus){
        alert(xhr.responseText);
        alert(textStatus);
    }
    });
</script>
</html>
```

hello.text :

*Hello World*

# Ajax / XML

```
$.ajax({ type: "GET", url: "courses.xml",
dataType: "xml",
complete : function(data, status)
{ var products = data.responseXML;
  var html = "";
  $(products).find('product').each(function()
  { var id = $(this).attr('id');
    var name = $(this).find('name').text();
    var price =$(this).find('price').text();
    html += "<li>#" + id + " - <strong>" + name + "</strong> : "
           + price + "</li>"; });
  $("#cousesList").html(html); })};
```

## Jquery et Json

```
$('#requestDataBtn').click(function(){ $.getJSON(
    'getPeople.php',
    { minimumAge: 21 },
    function(data, textStatus, jqXHR){
        // Success handler. // We expect something from the server like this:
        // { people: [ { name: 'Robbie', age: 24 }, { name: 'Sarah', age: 29 } ] }
        // Display the data.
        var newText = $('#results').html ;
        for(var i = 0; i < data.people.length; i++)
            { newText += 'Name: ' + data.people[i].name + ', Age: ' + data.people[i].age; }
        $('#results').html(newText);
    });
});
```

## Produire des effets visuels (les effets s'appliquent à des résultats de sélection) :

Quelques exemples :

Jquery possède des méthodes permettant :

> `slideUp()`, `slideDown()`, `slideToggle()` : effets (roulant) sur la hauteur des éléments

> `fadeIn()`, `fadeOut()`, `fadeTo()` : effets sur l'opacité (fondu)

> `show()`, `hide()`, `toggle()`: apparition/disparition ('slow', 'fast', durée)

Chacune de ses méthodes possède les paramètres (optionnels) :

>durée : indique la durée de l'effet en millisecondes (0 effet immédiat)

**>callback : fonction appelée sur chaque élément de la liste lorsque l'effet est terminé. Dans cette fonction *this* représente l'élément sur lequel l'effet vient de s'effectuer. Cela permet donc d'enrichir l'effet.**



```
<html>
<head>
<title>Enter your title here</title>
<script src="jquery.js"></script>
</head>
<body>
<p style="display:none">paragraphe 1</p>
<p style="display:none">paragraphe 2</p>
</body>
<script type="text/javascript">
var i = 1;
    $("p").show(1000,
        function(){
            this.style.color = "green";
        });

</script>
</html>
```

Qu'est que cela fait ?

Les effets peuvent être enchaînés :

```
$("p").fadeOut(2000).fadeIn(1000).slideUp(1000);
```

Testjquery5 - testjquery1

Séquence « complexe »:

```
$('#truc').fadeOut('slow', function(){  
    $('#truc').load('actu.html', function(){  
        $('#truc').show('slow', function(){  
            Alert('la séquence est finie');  
        });  
    });  
});
```

# Jquery UI : un ensemble de composants graphiques

>un noyau (Core)

>des « **comportements** » (interactions)

draggable : pour glisser-déplacer un élément

droppable : pour « déposer » un élément

resizable : pour redimensionner un élément

selectable : pour sélectionner des éléments à la souris

sortable : pour trier des éléments

> des « **widgets** » : « accordéon » « calendrier » boîte de dialogue, barre de progression, curseur, onglets ...

> des **effets**: Clignotement, disparition, apparition, éclatement, transition...

> des **thèmes**

**Exemples :** [www.jquerysample.com](http://www.jquerysample.com)

[http://www.w3schools.com/JQuery/jquery\\_examples.asp](http://www.w3schools.com/JQuery/jquery_examples.asp)  
(possibilité d'éditer le code – visualiser les effets)

The screenshot shows a web browser window with the URL [www.w3schools.com/JQuery/jquery\\_examples.asp](http://www.w3schools.com/JQuery/jquery_examples.asp). The page features the w3schools.com logo and a navigation menu with links to HOME, HTML, CSS, JAVASCRIPT, JQUERY, XML, ASP.NET, PHP, SQL, and MORE... The main content area is titled "jQuery Examples" and includes a "Previous" link, a paragraph about developing jQuery selector skills, and a link to a "jQuery Selector Tester". Below this, the "jQuery Selectors" section is displayed, showing examples of the `hide()` method: `$("#p").hide()`, `$("#test").hide()`, `$(".test").hide()`, and `$(this).hide()`. The right sidebar contains sections for "SHARE THIS PAGE", "WEB HOSTING", "WEB BUILDING", and "STATISTICS".

Be/Lonely Looking Sky/L x M NAW 2013: TP 10 x jQuery Examples x

www.w3schools.com/JQuery/jquery\_examples.asp

7 solutions open s Database Oracle -... javascript JExcelApi Swing : JTable How to Use Table... jacques.bapst.ho... Autres favoris

Cette page est en anglais Voulez-vous la traduire ? Traduire Non Ne jamais traduire les pages rédigées en anglais Options

w3schools.com

Sélectionner une langue Google Custom Search

HOME HTML CSS JAVASCRIPT JQUERY XML ASP.NET PHP SQL MORE... REFERENCES | EXAMPLES | FORUM | ABOUT

Get Certified  
Study Web Technologies  
and get a diploma  
at w3schools.com

Microsoft Dynamics CRM  
Voyez loin !  
Réinventez-vous avec  
Microsoft Dynamics CRM  
Efficacité commerciale, social  
CRM, relation clients :  
Téléchargez  
nos livres blancs

SHARE THIS PAGE  
f t e y d i su  
f J'aime 49k

WEB HOSTING  
Best Web Hosting  
eUK Web Hosting  
UK Reseller Hosting  
Cloud Hosting  
Your Ad Here

WEB BUILDING  
Download XML Editor  
FREE Website BUILDER  
FREE Website Creator

STATISTICS  
Browser Statistics  
OS Statistics  
Display Statistics

jQuery Tutorial  
jQuery HOME  
jQuery Intro  
jQuery Install  
jQuery Syntax  
jQuery Selectors  
jQuery Events

jQuery Effects  
jQuery Hide/Show  
jQuery Fade  
jQuery Slide  
jQuery Animate  
jQuery stop()  
jQuery Callback  
jQuery Chaining

jQuery HTML  
jQuery Get  
jQuery Set

## jQuery Examples

« Previous Next Chapter »

Do you want to develop your jQuery selector skills?  
Please try our excellent [jQuery Selector Tester](#)

## jQuery Selectors

`$("#p").hide()`  
Demonstrates the jQuery hide() method, hiding all <p> elements.

`$("#test").hide()`  
Demonstrates the jQuery hide() method, hiding the element with id="test".

`$(".test").hide()`  
Demonstrates the jQuery hide() method, hiding all elements with class="test".

`$(this).hide()`  
Demonstrates the jQuery hide() method, hiding the current HTML element.

**Examples explained**

# Autre bibliothèque (testhighchart1 et 2) :

