



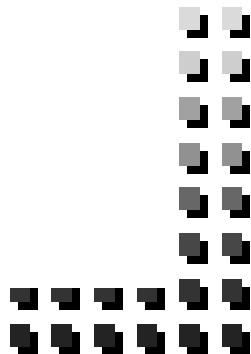
# Programmation sur Internet

## Cours 3 - La bibliothèque JQuery

---

Jean-Michel Ilié

IUT Paris 5 - département informatique



# Intérêts des bibliothèques javascript

---

## Bibliothèque avancée

- ☞ Référencement des objets HTML-DOM (collection de références supportée)
  - Accès aux styles, aux attributs et au contenu
  - Gestion des événements
  - Animation (apparition/disparition, cliquer/glisser...)
  - Widget - Objets graphiques évolués (menu, accordéon...)
  - Encapsulation Ajax (communication serveur interactive)

## Principe de séparation de code

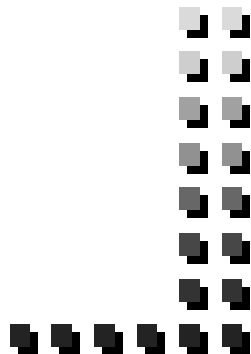
- ☞ Libère la page HTML du JAVASCRIPT

## Très bonne évolutivité

- ☞ Approche Orientée OBJET (héritage de présentation, de comportement)

## Large compatibilité

- ☞ Firefox , Internet Explorer, Safari , Opera, ...





# Principes d'utilisation de jquery

## Inclusion du script correspondant à instancier un objet JQUERY pour la page HTML

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
```

*( on peut aussi copier le fichier pour en avoir une version locale au site)*

*(Si un autre fichier de script fait référence à l'objet JQUERY, alors il doit être inclus à la suite !)*

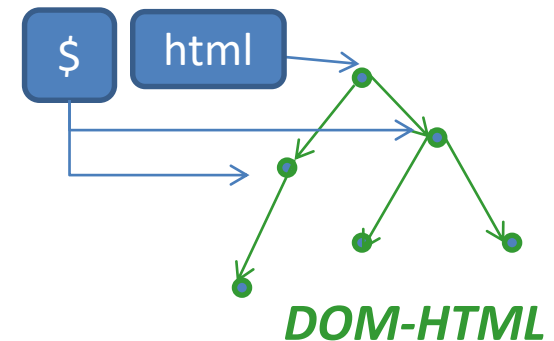
## Attribut global référençant l'objet JQuery créé :

\$ (vu comme un attribut de nom \$ dans l'objet window)

ou JQuery

L'objet JQuery ou \$ possède de nombreuses méthodes. La plupart s'applique à un tableau interne qu'il faudra initialiser et avec de références sur des objets HTML de la page.

*window*



# Paramétrage d'un objet JQUERY

*jQuery(...paramètre...)*

*ou*

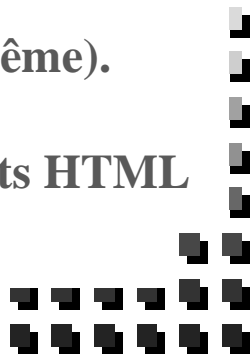
*\$(...paramètre...)*

L'instanciation d'un objet JQuery est paramétré pour initialiser a en interne :

- un tableau de références sur des objets HTML à gérer
- un tableau de références à des fonctions JS à lancer après le chargement des objets de la page HTML en mémoire.

Le paramètre peut prendre diverses formes:

- ✓ - une référence à une **fonction** (i.e. son nom ou la déclaration de la fct elle-même).
- ✓ - une **référence directe** à un objet HTML ou
  - une expression type **CSS spécifiant un ensemble de références** sur des objets HTML



# Exemple de code JQUERY

Le code JQUERY reste du Javascript mais on exploite les fonctionnalités de l'objet JQUERY

```
var r = $("p");  
r.html("un texte html");  
r.css('color','green');
```

sélection interne de tous les paragraphes.  
"r" référence l'objet JQuery ainsi instancié.

html et css sont 2 méthodes de l'objet JQuery : ici, modification de contenu et de style pour tous les paragraphes.

Equivalent JS pur :  
souvent moins  
intuitif  
Et moins compact !

```
var T = document.getElementsByTagName("p");  
for (var r in T) {  
    r.innerHTML = "un texte html";  
    r.style.color= "green";  
}
```

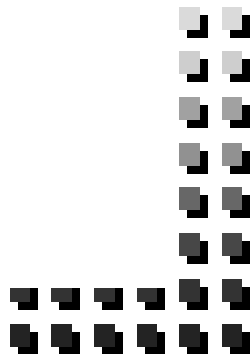
# Autre écriture : invocation chaînée des méthodes

---

Toutes les méthodes d'un objet JQuery retourne une référence correspondant à l'objet JQuery lui-même, ce qui permet d'enchaîner les méthodes de cet objet.

Exemple :

```
$("p").html("un texte html").css('color','green');
```



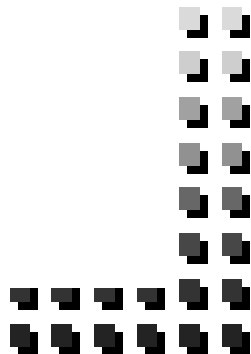
# Instanciación múltiple de l'objet JQuery

Il suffit d'utiliser l'objet \$ ! (pas de new)

```
var r1, r2;  
r1 = $("p");  
r2 = $("div");
```

Les 2 instanciaciones de l'objet \$  
sont en mémoire référencées par  
r1 et r2.

*Note : en Javascript, une allocation mémoire subsiste  
tant qu'elle est référencée.*





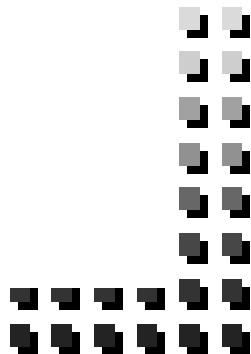
# Exemple de code lancé après chargement des objets-balises en mémoire.

---

```
$( function(){  
    var r = $("p");  
    r.html("un texte html");  
    r.css('color','green');  
});
```

le code de cette fonction  
ne sera exécuté qu'après  
le chargement des objets HTML  
en mémoire,  
*ce qui est heureux  
puisque'on fait référence  
aux objets-balises paragraphes !*

*Note : L'objet \$ peut être instancié plusieurs fois.*



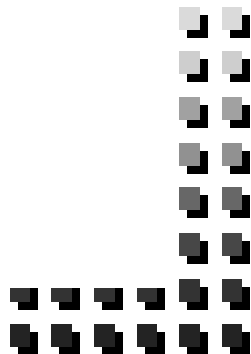
# Alternative pour lancer du code après chargement des objets-balises en mémoire.

---

Exploiter la fonction ready  
de l'objet \$  
L'objet document HTML doit être référencé  
Dans le tableau interne

```
$( document).ready (
    function (){
        ... code à lancer ...
    }
)
```

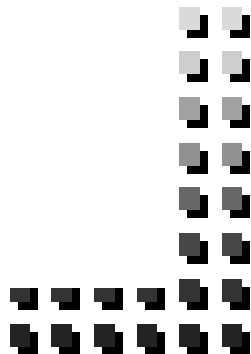
le code de cette fonction  
ne sera exécuté qu'après  
le chargement des objets HTML  
en mémoire,



# JQUERY plus en détails

---

- Techniques de sélection des objets HTML
- Panorama des méthodes de l'objet JQuery



Copyright © 2010 Pearson Education, Inc. All rights reserved. Printed in the United States of America. This publication is protected by copyright. Any unauthorized reproduction or distribution, in any form or by any means, without written permission from Pearson Education, Inc., is prohibited. All rights reserved.

Le paramètre de l'objet  
jQuery() accepte les  
expressions de chemins  
d'éléments DOM  
et les noms symboliques.

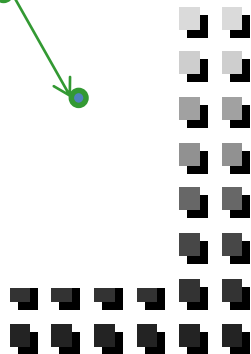
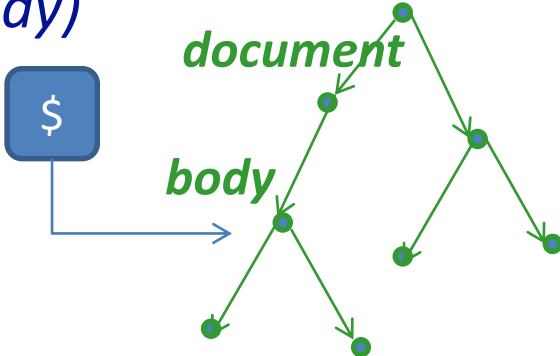
 **d1. ... .dn**

## Exple

***\$ (document.body)***

# DOM-HTML

***window***



# Référencement d'objets-html via un sélecteur type CSS

Un expression CSS permet d'initialiser le tableau interne d'un objet \$

👉 **sélecteur global**

`$("*")`

*Dans son tableau interne :  
l'objet \$ référence toutes les balises*

👉 **sélecteur avec nom de balises**

`$("p")`

*toutes les balises-paragraphes*

👉 **sélecteur avec classe de style**

`$("a.menu")`

*toutes les balises a de classe menu*

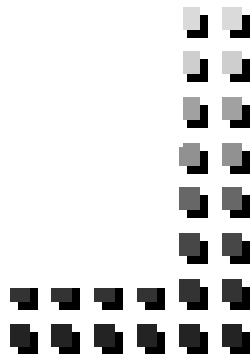
👉 **sélecteur avec identifiant de balises**

`$("#bloc")` *toutes les balises div avec id=bloc*

👉 **sélecteur avec un état**

`$("a:active.menu")`

*note : une suite de sélecteurs est possible avec le séparateur ","*



# Restriction des sélections de références avec un correcteur de l'expression CSS

👉 **:first :last**

`$("p:first")` soit le 1<sup>ER</sup> paragraphe

👉 **:eq(i), :lt(i), :gt(i)**

*L'élément d'indice =i  
ou ceux < ou > à i (i>=0)*

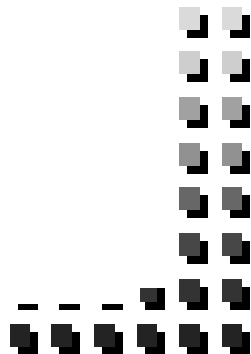
`:eq(0)`  
équivalent à:  
`:first`

👉 **:not(selecteur)**  
**:has(selecteur)**

`$("input:not(checked) ")`  
*Nombreux sélecteurs complexes possibles*

👉 **:hidden :visible :animated**  
**:enabled :disabled :selected**

*Restriction aux éléments masqués,  
visibles ou en cours d'animation  
grisé ou non, sélectionné*



# Restriction des sélections de références par le contenu

Exploiter l'imbrication des balises  
et leur contenu textuel

Exple

`$("div:contains('text'))"`

 **:contains('t')**

*1<sup>ER</sup> élément de la sélection qui  
contient la chaîne 't'*

 **:header**

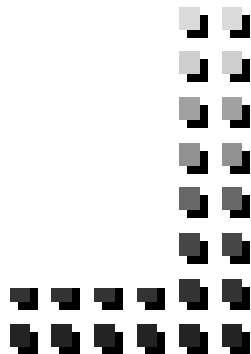
*Restriction aux éléments d'entête (balise <h.>)*

 **:empty**

*Les éléments vides  
(DOM sans fils (→ pas de contenu))*

 **:parent**

*Les éléments non vide  
(Les éléments DOM qui sont parents  
ont des fils et donc du contenu)*



# Restriction des sélections par les attributs

Exploiter les attributs des  
balises et leur valeur.

Exple

```
$("input[id]['name'$='pdg']")
```

👉 **`['a'] ['a'='v'] ['a'!='v']`**

Seuls sont sélectionnés,  
*les éléments ayant un attribut "a", ou encore,  
dont l'attribut "a" possède ou non la valeur "v"*

👉 **`['a'^='v'] ['a'$='v'] ['a'*='v']`**

*Pour obtenir les éléments ayant un attribut "a"  
commençant, finissant ou contenant "v"*

👉 **`[...] [...] [...]`**

*Enchaînement des conditions sur les attributs*





# Méthodes d'accès au contenu des objets HTML sélectionnés

Les fonctions de l'objet \$ s'appliquent sur les objets-balises qui sont référencés dans le tableau Interne de l'objet \$.

*Exple :*

*var c = \$("div#bloc").html();*

## Equivalent Javascript :

👉 **html()**

*innerHTML* (accès aux contenus visibles des blocs html de type div ...)

👉 **val()**

*attribut value* (ens. des valeurs visibles d'attributs pour les éléments sélectionnés: p, input text, radio btn ou checkbox, attrib value de

option)

👉 **text()**

*attribut text*

(ens. des contenus visibles des balises p, option select)

👉 **attr('a')**

*valeur de l'attribut "a" du 1<sup>ER</sup> élément de la sélection*

*Undefined si attribut 'a' non défini*

*(pour alt, src, width, style, ...).*

# Méthodes modifiant le contenu des objets HTML sélectionnés

*Exple :*

```
$("p").text("le mot<b> juste</b>.");
```

## Equivalent Javascript

👉 **html("v")**

*innerHTML* (pour contenus visibles HTML des éléments sélectionnés : blocs div, paragraphe p, ...)

👉 **text("v")**

*attribut text* (contenus visibles textuels des éléments sélectionnés : option de select. Aussi p, ... MAIS les caractères spéciaux HTML ne sont pas interprétés)

👉 **val("v")**

*attribut value* (pour attributs "value" (visibles) des éléments sélectionnés :  
input diverses , checkbox , attribut value de option.

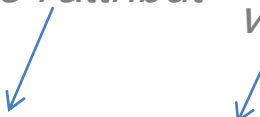


# Méthodes gérant les attributs des objets HTML sélectionnés

*Exple :*

*nom de l'attribut*      *valeur*

`$("button:gt(1)").attr("width",20px);`



☞ **attr("a","v")**

Positionne l'attribut "a" et sa valeur pour tous les éléments de la sélection.

☞ **attr("{a1:v1, ...an:vn}")**

Généralisation du précédent

☞ **removeAttr("a")**

Supprime l'attribut "a" des éléments sélectionnés.



# Méthodes de gestion des styles CSS

Les méthodes d'un objet  
JQUERY  
gèrent les styles css  
des éléments sélectionnés

## Exple

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

☞ ***css("pr")***

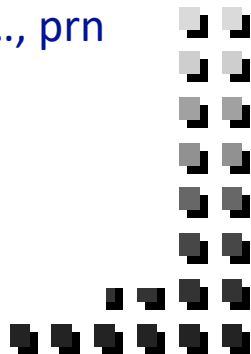
Rend la valeur de la propriété de style de nom "pr"  
(concerne le 1<sup>ER</sup> élément de la sélection seulement)

☞ ***css("pr:v")***

La propriété de style "pr" des éléments de la  
sélection auront la valeur "v"

☞ ***css("pr1:v1, ...prn:vn")***

Généralisation aux propriétés de style pr1, ..., prn



# Méthodes de gestion des classes de style CSS

Les méthodes d'un objet  
JQUERY  
gèrent les classes de  
styles css  
des éléments sélectionnés

## Exple

```
$("#onglets li").addClass("actif");
```

👉 **addClass("c")**

Crée ou redéfinit l'attribut class , qui aura la valeur "c"

👉 **removeClass("c")**

Élimine la spécification "c" pour l'attribut class

👉 **toggleClass("c")**

Les appels successifs de cette fonction alternent, add et remove concernant la classe de valeur "c"

👉 **hasClass("c")**

teste si l'attribut classe est spécifié avec la valeur "c"

# Méthodes filtrant la sélection

L'objet JQUERY  
possède des méthodes  
gérant les sélections

## Exple

```
var t=$("div, span, p.maClasse");  
t.eq(1).css("border","3px solid red");
```

👉 **eq(i)**

ne retient de la sélection que l'élément d'index i (i>=0)

👉 **filter("e")**

ne retient de la sélection que ceux correspondant à l'expression "e" (expression CSS+)

👉 **not("e")**

ne retient de la sélection que ceux ne correspondant pas à e

# Méthodes de référencement d'objets-balises exploitant la hiérarchie HTML

---

## Exple

```
$("div").children("li").css("border","3px solid red");  
$("div").parent("div");
```

☞ **children()**

sélectionne les objet HTML **enfant**

☞ **children("e")**

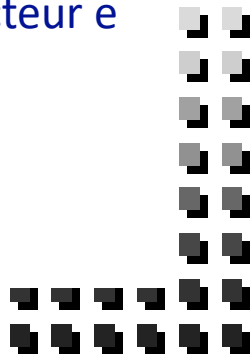
sélectionne ceux qui correspondent au sélecteur e

☞ **parent()**

sélectionne l'objet HTML **parent**

☞ **parent("e")**

idem avec vérification qu'il correspond au sélecteur e



# Exemple d'accès aux balises via la hiérarchie

```
function charge() {  
    var inputs= $("div#main .content").children();  
  
    var id0= inputs.eq(0).attr("value");  
    var id1= inputs.eq(1).attr("value");  
  
    var charge_url = "index.php? Id_0=" + id0 + "&id_1=" + id1;  
    window.location.replace(charge_url);  
}
```

```
<div id="main"><ul>  
    <li class="content">  
        <input type="hidden" name="id1" value="11" />  
        <input type="hidden" name="id2" value="1" />  
        Activité  
    </li></ul>  
</div>
```



# Méthodes permettant un traitement itératif sur les objets HTML sélectionnés

L'objet JQUERY possède un itérateur pour les éléments sélectionnés

## Exple

```
$("#div").each(function () {  
    if (this.style.color != "black") {  
        this.style.color = "black";  
    } else {  
        this.style.color = "red";  
    }  
});
```

👉 **each(f)**

La fonction *f* (ou référencée par *f*) est appliquée à chaque élément de la sélection

👉 **valeur this**

this = la référence de l'élément courant dans l'itération

👉 **length ou size()**

information sur le nombre d'éléments de la sélection

# Méthodes définissant les réactions aux événements pour les objets HTML sélectionnés

## Méthodes permettant d'associer une méthode de réaction événementielle aux objets HTML sélectionnés

- ☞ **click(f), dblclick(f)**
- ☞ **toggle(f1, f..., fn)**
- ☞ **mouseout(f), mouseover(f),**
- ☞ **mouseenter(f), mouseleave(f),**
- ☞ **mousedown(f), mouseup(f),**
- ☞ **keypress(), keypress(f)**
  
- ☞ **select(f), blur(f),**
- ☞ **change(f) , focus(f)**
- ☞ **submit(f)**
- ☞ **error(f)**
  
- ☞ **load(f), unload(f)**
- ☞ **resize(f), scroll(f)**

### Exple

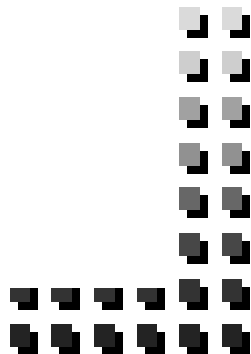
```
$("a").click( function() {  
    alert("Coucou!");  
    return(false);  
});
```

avec :

```
<a href="#" > Cliquez fait coucou</a>
```

*Ecriture équivalente en HTML !*

```
<a href="#"  
    onclick="alert('Coucou');"  
>  
Cliquez fait Coucou </a>
```



# Autres exemples de gestion événementielle

```
$("#p").mouseover ( //sur un survol du paragraphe
    function () { $(this).css("color","red"); }
);
```

```
$("#p").hover( //fonctionnement alterné du survol (mouseout,moseover) → 2 fonctions
    function () {
        $(this).css({'background-color' : 'yellow', 'font-weight' : 'bolder'});
    },
    function () {
        var cssObj = { 'background-color' : '#ddd', 'font-weight' : '', 'color' : 'rgb(0,40,244)' }
        $(this).css(cssObj);
    }
);
```

```
$("#div").click(function () {
    var color = $(this).css("background-color");
    $("#result").html("Ce bloc est de couleur : "
        + "<span style='color:" + color + "';>" + color + "</span>.");
});
```

# Méthodes d'activation des événements pour les objets HTML sélectionnés

---

Réalise l'événement sur les objets HTML sélectionnés.

Ne pas mettre de paramètres dans les parenthèses.

☞ **click() dblclick()**

☞ **mouseover() mouseout()**

☞ **select()**

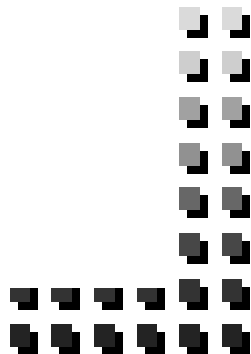
☞ **blur()**

☞ **change()**

☞ **focus()**

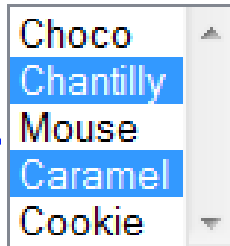
☞ **submit()**

☞ **error()**



# Exemple de définition et d'activation des événements

👉 **change()**



```
<body>
<select
    multiple="multiple">
<option>Choco</option>
<option selected="selected">
    Chantilly
</option>
<option>Mouse</option>
<option selected="selected">
    Caramel
</option>
<option>Cookie</option>
</select>
<div></div> </body>
```

```
$(document).ready(
    function() {
```

```
    $("select").change (
        function () {
```

```
        var str = "";
```

```
        $("select option:selected").each(
            function () {
```

```
            str += $(this).text() + " ";
```

```
        });
```

```
        $("div").text(str);
```

```
    }
```

```
})
```

```
.change(); //force l'exécution de la réaction
           //demandée sur change
```

```
});
```

*Affiche la sélection multiple  
à chaque changement.*

# Informations sur les objets sélectionnés

---

```
if ($(...).length) {  
}
```

👉 **length ou size()**

information sur le nombre d'éléments de la sélection  
*très utile pour savoir si un objet est défini.*

👉 **context**

Référence du noeud DOM-HTML,  
point de départ de la sélection JQUERY  
(*par défaut : objet document*)

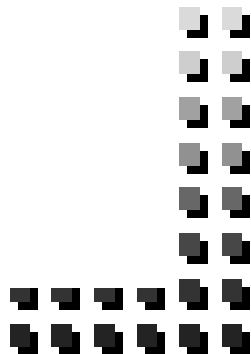
👉 **selector**

Le sélecteur utilisé pour la sélection JQuery

👉 **get**

👉 **get(i)**

obtenir un tableau correspondant aux  
références des objets HTML sélectionnés  
ou plus finement une référence sur l'objet  
d'index i (i>=0)



# Méthodes d'animation appliquées aux objets HTML sélectionnés

Exple

```
$("button").click(function () {  
    $("p").show("slow");  
});
```

👉 **show()** **show(s, f)**

👉 **hide()** **hide(s, f)**

👉 **toggle()** **toggle(b)**

*rend visible/invisible avec la vitesse 's' : slow,fast,normal ;  
et termine en exécutant la fonction 'f' optionnelle.*

*toggle : alterne la visibilité (caché-> apparaît et vice versa),  
optionnellement toggle peut dépendre d'un booléen "b"*

👉 **slideDown(s, f)**

👉 **slideUp(s, f)**

*rend visible en jouant sur la taille des éléments*

👉 **fadeIn(s, f)**

👉 **fadeOut(s, f)**

👉 **fadeTo(s, o, f)**

*rend visible en jouant sur l'opacité  
(o : valeur borne d'opacité à ne pas dépasser)*



# Méthodes d'animation personnalisables

👉 **animate**(*params, duration, easing, f*)

👉 **animate**(*params, options*)

👉 **stop**()

*Les valeurs peuvent  
être spécifiées  
en em ... ou %.*

*L'opérateur =  
peut être remplacé par += et -=*

*Anime toutes les propriétés  
spécifiées,  
suivant une durée donnée*

Exple

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



# Extensions de JQUERY

## PLUGINGs (extension JQuery)

Interface <http://interface.eyecon.ro/?page=demos>

ThickBox <http://jqueryui.com/demos/>

UI Datepicker <http://marcgrabanski.com/code/ui-datepicker/>

...

*Exples Interface*

*Menus*

*Accordéons*

*Retaillable*

*Cliquer/glisser...*

*Autocomplétion*

...

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

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

```
<link rel="stylesheet" href="../../../interface.css" type="text/css" media="screen" />
```



# Quelques liens utiles

---

<http://jqueryui.com/demos>

<http://interface.eyecon.ro/demos/?page=demos>

[http://docs.jquery.com/Tutorials#Tutoriaux\\_en\\_Fran.C3.A7ais](http://docs.jquery.com/Tutorials#Tutoriaux_en_Fran.C3.A7ais)

