

1. Essentiel de JQuery

1.1. Syntaxe générale et comportement

\$(selecteurXy) est un équivalent compact de **jQuery**(selecteurXy) .

Et, en particulier, on a les équivalences suivantes au niveau du point de démarrage :

```
jQuery(document).ready( function() {  
    // ici, le DOM de la page est entièrement défini et peut être manipulé via jQuery  
});
```

```
$(document).ready( function() {  
    // ici, le DOM de la page est entièrement défini et peut être manipulé via jQuery  
});
```

```
$( function() {  
    // ici, le DOM de la page est entièrement défini et peut être manipulé via jQuery  
});
```

La syntaxe générale d'une expression jQuery est **\$('selecteurXy').actionZz(parametres);**

La syntaxe des sélecteurs est la même que pour les styles CSS.

Les actions sont quelquefois en mode "lecture" , d'autres fois en mode "modification" et peuvent être enchaînées : **\$(selecteurXy).actionA('param1a' , 'param2a').actionB('paramB') ;**

1.2. Principaux sélecteurs (idem CSS)

\$('*');	tous les éléments (inclus <html>, <head> et <body>).
\$('element');	tous les éléments étant de type correspondant.
\$('#element');	l'élément ayant l' id donné (unique).
\$('.element');	tous les éléments ayant la classe donnée.
\$('element1, element2');	les éléments 1 et 2 et ... (selon # , . ou autre)
\$('parent enfant');	tous les enfants directs ou indirects de l'élément parent.
\$('parent > enfant');	tous les enfants directs de l'élément parent.
\$('frere + element');	tous les éléments précédés directement d'un frère.
\$('frere ~ element');	les éléments précédés directement ou indirectement d'un frère.
\$('element[attr]');	tous les éléments possédant l'attribut donné.
\$('element[attr="val"]');	tous les éléments possédant l'attribut donné et dont la valeur est

<code>\$(element[attr!="val"]);</code>	égale à celle spécifiée. tous les éléments possédant l'attribut donné et dont la valeur est différente de celle spécifiée.
<code>\$(element[attr*="val"]);</code>	tous les éléments possédant l'attribut donné et dont la valeur contient , entre autres, la chaîne spécifiée.
<code>\$(element[attr^="val"]);</code>	tous les éléments possédant l'attribut donné et dont la valeur commence par la chaîne spécifiée.

`$('a')[0]` retourne le premier lien hypertexte de la page.

`$(p.redClass')[3]` retourne la quatrième balise `<p>` de classe `redClass`.

`$('[src]')` sélectionne tous les éléments qui possèdent un attribut `src` ;

`$('[width="100"]')` sélectionne tous les éléments qui ont un attribut `width` égal à 100.

Quelques filtres :

<code>\$(':even'); \$(':odd');</code>	index est pair ou impair
<code>\$(':empty');</code>	Sans enfant
<code>\$(':first'); \$(':last');</code>	premier , dernier
<code>\$(':first-child'); \$(':last-child');</code>	premier , dernier enfant
<code>\$(':focus');</code>	tous les éléments ayant actuellement le focus.
<code>\$(':hidden'); \$(':visible');</code>	caché , visible

Exemples :

`$('tr:odd')` sélectionne les lignes impaires d'un tableau.

`$('td:first')` sélectionne la première cellule

Sélection selon type des éléments d'un formulaire :

`$(':input'); $(':button'); $(':checkbox'); $(':checked'); $(':file'); $(':password');`
`$(':radio'); $(':submit'); $(':text');`

Conversions (js/dom , jquery):

```
var variableJQ = $(variableJS);  
var variableJS = $(selecteurJQ).get();
```

1.3. Principales actions

Méthodes de parcours:

find()	Permet de trouver un enfant particulier.
children()	Trouve l'enfant direct de l'élément ciblé.
parent()	Trouve le premier parent de l'élément ciblé.
parents()	Trouve tous les ancêtres de l'élément ciblé.
siblings()	Trouve tous les éléments "frères" (de même niveau)
each()	Boucle sur chaque élément.

```
$('sel').each( function(index) {
    //Une ou plusieurs instructions JavaScript
    //NB: index vaudra automatiquement 0,1,2,..., n-1
});
```

Récupération de positions et de tailles:

width(), height()	largeur, hauteur
innerWidth(), innerHeight()	Largeur, hauteur en prenant en compte les marges intérieures mais pas les bordures.
outerWidth(), outerHeight()	Largeur, hauteur en prenant en compte ses marges intérieures, extérieures, et ses bordures.
offset()	Récupère les coordonnées absolues de l'élément ciblé.
position()	Récupère les coordonnées relatives de l'élément ciblé.
scrollLeft(),scrollTop()	Positions horizontale et verticale de la barre de défilement par rapport à la page.

Manipulation d'attributs:

attr()	Récupère ou modifie l'attribut d'un élément.
prop()	Gère une propriété d'un élément (ex: disabled)
removeAttr()	Supprime l'attribut d'un élément.
addClass() removeClass()	ajoute ou supprime une classe à un élément.
hasClass()	Vérifie si un élément a ou pas telle classe css.

Exemples:

```
$('#itemXy').attr('src','logo.gif'); $('#btnHide').prop('disabled',true);
$('#itemXy').attr({ src: 'logo.gif', alt: 'société Xy', width: '250px'});
$('a').attr('target', function() {
    if(this.host == location.host) return '_self'
    else return '_blank'
});
```

Manipulations HTML :

html()	Récupère ou modifie le contenu HTML de l'élément ciblé.
text()	Récupère ou modifie le contenu textuel de l'élément ciblé.
val()	Récupère ou modifie la valeur d'un élément de formulaire.
append() <i>ou</i>	Ajoute du contenu HTML ou textuel à la fin de l'élément ciblé(à l'intérieur , comme enfant , sous élément)
appendTo()	
prepend() <i>ou</i>	Ajoute du contenu HTML ou textuel au début de l'élément ciblé (à l'intérieur , comme enfant , sous élément)
prependTo()	
before()	Ajoute du contenu HTML ou textuel avant l'élément ciblé.
after()	Ajoute du contenu HTML ou textuel après l'élément ciblé.
empty()	Vide un élément.
remove()	Supprime un élément.
wrap()	Enveloppe un élément.

1.4. Principaux gestionnaires d'événement

La source d'un événement est **evt.target** et l'on peut commencer des instructions/actions utiles avec **evt.target.id** ou bien **\$(evt.target)** .

Bien que l'on puisse directement gérer des événements javascript via des méthodes jQuery reprenant spécifiquement les noms des événements, il est également possible d'utiliser la fonction générique **.on('type_evenement', function() { ...})** qui est "universelle", "plus paramétrable" et "utilisable sur de nouveaux événements personnalisés" .

Exemple simple:

```
$('#btnEuroToFranc').on('click',function(){
    $('#txtFranc').html( 6.55957 * Number( $('#txtEuro').val() ) );
});
```

Principaux événements "souris":

Événements (evt.type)	circonstances	paramètres importants
click	click gauche	
dblclick	double click	
mousedown	Appui sur un bouton (gauche, droit, ...) de la souris	evt.which valant 1 pour bouton gauche 2 pour bouton central 3 pour bouton droit
mouseover ou mouseenter	début de survol de l'élément	
mouseout ou mouseleave	fin de survol de l'élément	
mousemove	Déplacement du pointeur de souris au dessus de l'élément	
mouseup	Relachement d'un bouton (gauche, droit, ...) de la souris	idem mousedown
scroll	scroll via molette de la souris	

Exemple:

```
$('#target').on('mousedown', function(evt){
    $('#message').html('Événement : ' + evt.type + '. Bouton pressé : ' + evt.which );
});
```

Position de la souris (véhiculée par l'événement source) :

'x='+ **evt.pageX**+' y='+ **evt.pageY** // position absolue par rapport à la page

'x='+ **evt.clientX**+' y='+ **evt.clientY** // position absolue par rapport à la zone cliente (partie visible)

// de la page dans le navigateur selon position des ascenseurs)

'x='+ (evt.pageX - this.offsetLeft) +' y='+ (evt.pageY - this.offsetTop) // % élément courant

Principaux événements "clavier":

Événements (evt.type)	circonstances	paramètres importants
keydown	Appui sur une touche	evt.which valant le code touche (sans différence min et MAJ et avec code pour F1, F2 , ...)
keyup	Relachement d'une touche	idem keydown
keypress	(Appui + relachement) effectué	evt.which valant le code ASCII du caractère (ex : 65 pour 'A' 97 pour 'a')

NB : var c = **String.fromCharCode**(e.which); permet de convertir le code ASCII en caractère correspondant ; ce qui est pratique pour coder 'keypress' .

Principaux événements applicables sur "contrôle d'un formulaire ou ...":

Événements (evt.type)	circonstances
focus	Réception du focus
blur	Perte du focus
focusin	Réception du focus par l'élément ou un de ses enfants
focusout	Perte du focus par l'élément ou un de ses enfants
resize	Redimensionnement
change	Changement d'état ou de sélection

```
function my_generic_event_handler(evt){
    var rapport_evt = "evt.type=" + evt.type + " and evt.which=" + evt.which + " on #" + evt.target.id;
    var new_option = "<option>" + msg + "</option>" ;
    $("#listboxEvents").append(new_option) ;
}
```

```
$('#select1').on('click change focus blur',my_generic_event_handler);
```

```
$('#radioCelibataire,#radioEnCouple').on('click change focus blur',my_generic_event_handler);
```

Les événements **"load"** et **"unload"** sont d'autre part déclenchés lors du téléchargement des pages et des images .

Eventuelle désactivation de certains gestionnaires d'événements :

`$('#txt1').off('click dblclick mousedown mouseup mouseover mouseout scroll');`

1.5. Principaux effets (transition , animation)

animate()	Anime une ou plusieurs propriété(s) CSS, à l'aide d'arguments tels que la durée ou l'accélération de l'animation.
hide()	Fait disparaître un élément (et lui donne la propriété <i>display:none</i>).
show()	Fait réapparaître un élément.
fadeOut()	Fait disparaître un élément (qui aura la propriété <i>display:none</i>) avec un effet de fondu.
fadeIn()	Fait réapparaître un élément avec un effet de fondu.
slideUp()	Fait disparaître un élément avec un effet de glissement.
slideDown()	Fait réapparaître un élément avec un effet de glissement.
stop()	Arrête l'animation en cours.

1.6. Ajax via jquery

\$.ajax()	Exécute une requête AJAX de type GET ou POST ou PUT ou DELETE ou ...
\$.post()	Exécute une requête AJAX de type POST (raccourci de \$.ajax()).
\$.get()	Exécute une requête AJAX de type GET (raccourci de \$.ajax()).
load()	Charge du contenu HTML de manière asynchrone.

2. Plugins JQuery

Utilisation de plugins prédéfinis

`<script src="....js"></script>` et lire la documentation du plugin .

Pack "jquery-ui"

<http://jqueryui.com> (demo , download) est l'URL de référence pour récupérer le pack de plugins "jquery-ui" .

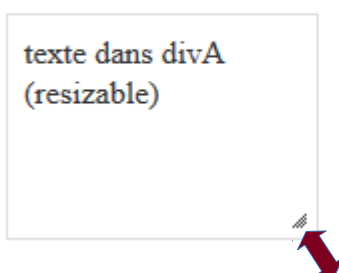
Interactions (automatiques) prises en charge par jquery-ui :

Draggable , **Droppable** , **Resizable** , **Selectable** , **Sortable**

Exemple :

```
<html>
<head> <meta charset="UTF-8"> <title>draggable , resizable with jquery-ui</title>
  <link rel="stylesheet" type="text/css" href="css/jquery-ui.css" />
  <script src="lib/jquery-2.2.1.js"></script>
  <script src="lib/jquery-ui.min.js"></script>
  <style>
    #divA { width: 150px; height: 150px; padding: 0.5em; }
  </style>
  <script>
    $(function() {
      $('#spanXx').css('border-width', '3px').css('border-style','solid');
      $('#spanXx').draggable();
      $('#divA').resizable();
    });
  </script>
</head>
<body>
  <span id="spanXx"> texte dans div or spanXx (draggable)</span>
  <br/><br/>
  <div id="divA" class="ui-widget-content"> texte dans divA (resizable)</div>
</body>
```

→ texte dans div or spanXx (draggable)



Principaux Widgets de jquery-ui :

[Accordion](#) , [Autocomplete](#), [Button](#), [Datepicker](#) , [Dialog](#) , [Menu](#)
[Progressbar](#) , [Selectmenu](#) , [Slider](#) , [Spinner](#) , [Tabs](#) , [Tooltip](#)

Exemples :

```
<html>
<head> <meta charset="UTF-8"> <title>jquery-ui widget</title>
  <link rel="stylesheet" type="text/css" href="css/jquery-ui.css" />
  <script src="lib/jquery-2.2.1.js"></script>
  <script src="lib/jquery-ui.min.js"></script>
  <style> #myDialog { display: none;}</style>
</script>
$(function() {
$.datepicker.regional['fr'] = {
monthNames:['Janvier','Février','Mars','Avril','Mai','Juin','Juillet',
'Août','Septembre','Octobre','Novembre','Décembre'],
dayNames: ['Dimanche','Lundi','Mardi','Mercredi','Jeudi','Vendredi','Samedi'],
dayNamesMin: ['Di','Lu','Ma','Me','Je','Ve','Sa'],
dateFormat: 'dd/mm/yy', firstDay: 1, isRTL: false,
showMonthAfterYear: false, yearSuffix: '';
$.datepicker.setDefaults($.datepicker.regional['fr']);
$("#datepicker1").datepicker();
$('#btnOpenDlg').on('click', function(){
  $("#myDialog").dialog( {modal: true,
    buttons: { "Oui": function() {
      $('body').css('background', $('#sBackColor').val());
      $( this ).dialog( "close" );},
      "Non": function() { $( this ).dialog( "close" );}
    }
  });
} //end of evt function
); //end of on_click
$('#serieOnglets').tabs(); //look et comportement sous forme d'onglets
});
</script>
</head>
<body> Date: <input type="text" id="datepicker1" /> <br/>
```

```
<input type='button' id='btnOpenDlg' value='open myDialog' /> <br/>
```

```
<div id='myDialog' title='Simple Dialog'>
```

Cette boîte de dialogue peut être redimensionnée, déplacée et fermée. <hr/>

```
backColor : <select id='sBackColor'>
```

```
<option checked='true' >white</option>
```

```
<option value='#eeee00'>yellow</option>
```

```
</select> Voulez vous cette couleur de fond ?
```

```
</div>
```

```
<div id='serieOnglets'>
```

```
<ul>
```

```
<li><a href='#onglet-1'>Titre onglet 1</a></li>
```

```
<li><a href='#onglet-2'>Titre onglet 2</a></li>
```

```
<!-- <li><a href='....'>recup possible contenu via ajax</a></li> -->
```

```
</ul>
```

```
<div id='onglet-1'> <p>contenu onglet1</p> </div>
```

```
<div id='onglet-2'> <p>contenu de l' onglet 2</p> </div>
```

```
</div> </body> </html>
```

The screenshot displays three components of a web application:

- Date Picker:** A date input field showing '30/03/2016' with a dropdown arrow. Below it is a calendar for 'Mars 2016' with days of the week (Lu, Ma, Me, Je, Ve, Sa, Di) and dates 1 through 31. The date '30' is highlighted in yellow.
- Simple Dialog:** A modal window titled 'Simple Dialog' with a close button (X). The text inside reads: 'Cette boîte de dialogue peut être redimensionnée, déplacée et fermée.' Below this, there is a label 'backColor :' followed by a dropdown menu showing 'white' and the text 'Voulez vous cette couleur de fond ?'. At the bottom are two buttons: 'Oui' and 'Non'.
- Tabbed Interface:** Two tabs are visible: 'Titre onglet 1' and 'Titre onglet 2'. The second tab is active and highlighted in blue. Below the tabs, the text 'contenu de l' onglet 2' is displayed.

2.1. Programmation (facile) de plugins personnalisés

Squelette d'un plugin jquery :

```
(function($) {  
  $.fn.myFct = function(éventuels_paramètres)  
  {  
    this.each(function() {  
      // les instructions du plugin  
      $(this).wrap('<b><i></i></b>') ; //exemple simple  
    });  
    return this; //pour pouvoir enchaîner (si nécessaire)  
  };  
})(jQuery);
```

Utilisation (habituelle) :

`$('#idXy').myFct('param1', 'param2');`

Exemple :

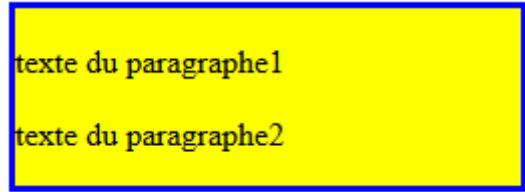
js/my-jquery-plugin.js

```
(function($) {  
  $.fn.withBorder = function(myBorderColorParam)  
  {  
    var borderColor = myBorderColorParam;  
    if(borderColor == undefined){  
      borderColor = 'black';  
    }  
  
    this.each( function() {  
      $(this).css('border-width', '3px')  
        .css('border-color', borderColor).css('border-style','solid');  
    });  
    return this;  
  };  
})(jQuery);
```

Utilisation :

```
<html>  
<head> <meta charset="UTF-8"> <title>test custom jquery plugin</title>  
  <script src="lib/jquery-2.2.1.js"></script>  
  <script src="js/my-jquery-plugin.js"></script>  
</script>  
  $(function() {
```

```
//$('#divA').withBorder().css('background','yellow');  
$('#divA').withBorder('blue').css('background','yellow');  
});  
</script>  
</head>  
<body>  
  <div id="divA">  
    <p id="p1"> texte du paragraphe1 </p>  
    <p id="p2"> texte du paragraphe2 </p>  
  </div>  
</body>  
</html>
```



texte du paragraphe1

texte du paragraphe2