

Js, JQuery And Ajax

To use jquery functions in javascript we should precede the function name by JQuery or \$, the last one is more used.

Syntax :

Basic syntax is: \$(selector).action()

A \$ sign to define/access jQuery

A (selector) to "query (or find)" HTML elements

A jQuery action() to be performed on the element(s)

Selectors :

\$("#html_tag") to select all tags of type « html tag ».

\$("#id") to select html tag which has a specific id

\$(".id") to select html tags which belongs to a specific class

\$("#html_tag :not(#id)) : to select all tags of type « html tag » except the one which has the id «#id»

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

\$('[width="40"]') sélectionne tous les éléments qui ont un attribut width égal à 40.

Run after page loading ends :

to prevent any jQuery code from running before the document is finished loading (is ready) :

```
$(document).ready(function(){  
//jQuery methods go here...  
});
```

Calling a function the following way does the same thing :

```
$(function(){  
    // jQuery methods go here...  
});
```

Events :

\$("#p").click(function(){//define it here});

\$("#p").dblclick(function(){});

\$("#p1").mouseenter(function(){});

\$("#p1").mouseleave(function(){});

When any mouse button (left,middle or right) is down : \$("#p1").mousedown(function(){});

\$("#p1").mouseup(function(){});

Hover : the combination between **mousedown** and **mouseup**, it takes two functions :

```
$("#p1").hover(function(){});
```

Keypress(function(event){}) ; // event.which==value to check which key has been pressed (13 for enter)

```
$("#input").focus(function(){});
```

The **on()** method attaches one or more event handlers for the selected elements :

```
$("#selector").on{  
    mouseenter: function(){},  
    mouseleave: function(){},  
    click: function(){}  
};
```

Predefined functions : \$(selector).function(){define it here} ;

```
css("name","value") ;
```

```
hide() ;
```

```
$("#checkbox_id").prop("checked") ; //check if a checkbox is checked
```

```
event.preventDefault() ; // to prevent a default action, for example line return on enter.
```

selector.**html**() returns the current content of a selector.

selector.**html**(String) set the content of a selector to String.

selector.**scrollTop**() ; returns the current position of the scroll (0 is the top)

selector.**scrollTop**(value) ; set the position of the scroll

selector.**prop**("ScrollHeight") ; returns the total height of the scroll.

The **attr()** method sets or returns attributes and values of the selected elements :

```
*attr(attribute[,value])
```

```
show(speed,easing,callback) (hide)
```

animate({params},speed,callback) params receives css properties and values (prop: val), **all property names must be camel-cased** when used with the animate() method (didn't work with color & backgroundColor)

```
.children(":selected")
```

```
$('#masterdiv div').html("");
```

Insert a tag before/after another :

```
$(existent_tag).before("<tag>new tag content...</tag>")
```

(same for after)

Check if a value is a number :

isNumeric(value) returns boolean

Get the previous/next element :

\$(element).prev and same for next

Get the closest parent of a given type :

\$(element).closest(parent_to_search)

PRACTICAL EXAMPLE :

```
$(...).show("slow").delay(4000).hide("slow");
```

AJAX :

```
$(selector).load(URL[ selector],data,callback);
```

The required URL parameter specifies the URL you wish to load.

The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

The optional callback parameter is the name of a function to be executed after the load() method is completed.

Examples :

```
$("#div1").load("demo_test.txt"); // load the demo_test.txt content to #div1
```

```
$("#div1").load("demo_test.txt #p1"); // loads the p1 paragraph from demo_test.txt to #div1
```

Callback :

The optional callback parameter specifies a callback function to run when the load() method is completed. The callback function can have different parameters:

responseTxt - contains the resulting content if the call succeeds

statusTxt - contains the status of the call

xhr - contains the XMLHttpRequest object

```
$("#button").click(function(){
    $("#div1").load("demo_test.txt", function(responseTxt, statusTxt, xhr){
        if(statusTxt == "success")
            alert("External content loaded successfully!");
        if(statusTxt == "error")
            alert("Error: " + xhr.status + ": " + xhr.statusText);
    });
});
```

```
});
```

```
});
```

.ajax({name : value, name : value ...}) method :

url : link of the processing page,

type : 'post' or 'get' ,

dataType : 'html',

data : 'data1='+data+'&data2='+data2 ... ,

success : function(html){} // html contains the returned html code

error : function({})