

< **BASE** >

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)

// *Examples:*

\$(this).hide() - hides the current element.

\$("p").hide() - hides all <p> elements.

\$(".test").hide() - hides all elements with class="test".

\$("#test").hide() - hides the element with id="test".

**Déclarer une variable:**

```
var numero_chance = 7;
```

```
var nom = "Codecademy";
```

```
var $p = $('p');
```

**Syntaxe de base jQuery:**

```
$(document).ready(function() {  
    // Quand le document est prêt : Instructions(s)  
});
```

**Syntaxe action:**

```
$(document).ready(function() {  
    $(element).action(paramètre);  
    // + Appliquer sur l'élément l'action choisie, spécifiée par son paramètre.  
});
```

**Syntaxe évènement:**

```
$(document).ready(function() {  
    $(element).evenement(function() {  
        $(element).action();  
        // Lorsque l'évènement apparaît sur l'élément choisi, effectuer l'action choisie.  
    });  
});
```

< **SELECTION** >**Sélection d'éléments :**

\$("\*") Selects all elements

\$(this) Selects the current HTML element

\$("p.intro") Selects all <p> elements with class="intro"

\$("p:first") Selects the first <p> element

\$("ul li:first") Selects the first <li> element of the first <ul>

\$("ul li:first-child") Selects the first <li> element of every <ul>

\$("[href]") Selects all elements with an href attribute

\$("a[target='\_blank']") Selects all <a> elements with a target attribute value equal to "\_blank"

\$("a[target!='\_blank']") Selects all <a> elements with a target attribute value NOT equal to "\_blank"

\$(":button") Selects all <button> elements and <input> elements of type="button"

\$("tr:even") Selects all even <tr> elements

\$("tr:odd") Selects all odd <tr> elements

< **EVENT** >

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload
mousedown			
mouseup			
hover			

**The on() Method:**

```
$(document).ready(function() {  
    $("p").on({  
        mouseenter: function() {  
            $(this).css("background-color", "lightgray");  
        },  
        mouseleave: function() {  
            $(this).css("background-color", "lightblue");  
        },  
        click: function() {  
            $(this).css("background-color", "yellow");  
        }  
    });  
});
```

// *Permet de lier un et plus évènements de gestion à un seul élément.*

## < EFFECTS >

```
$(document).ready(function() {  
    $("button").click(function() {  
        $('div').toggle('slow');  
    });  
});
```

### Hide / Show :

```
.hide(); //Masquer / Param : slow, fast, milliseconde...  
.show(); //Montrer / Param : slow, fast, milliseconde...  
.toggle(); //Hide and show 2 en 1
```

### Fade :

```
.fadeOut('fast', 0.25); //Faire disparaître  
.fadeIn('fast', 0.25); //Faire apparaître  
.fadeTo('fast', 0.25); //Griser
```

### Slide :

```
.slideUp('slow'); // Faire translater  
.slideDown('slow'); // Faire translater  
.slideToggle('slow'); // Faire translater
```

### Animate (1 ou plusieurs propriétés) :

```
$("div").animate({  
    left: '250px',  
    opacity: '0.5',  
    height: '150px',  
    width: '150px'  
});
```

### Animate (Relative Values) :

```
width: '+=150px'
```

### Animate (Valeurs prédéfini) :

```
.animate({  
    height: 'toggle'  
});
```

### Animate (Queues de fonctionnalités) :

```
$("button").click(function() {  
    var div = $("div");  
    div.animate({left: '100px'}, "slow");  
    div.animate({fontSize: '3em'}, "slow");  
});
```

### Stop:

```
.stop(); // Stop l'animation à laquelle l'action est rattachée
```

### Callback:

```
// Typical syntax: $(selector).hide(speed,callback);  
$("p").hide("slow", function() {  
    alert("The paragraph is now hidden");  
}); // Mettre comme callback, une alerte
```

### Chaines d'instructions:

```
$("#p1").css("color", "red")  
.slideUp(2000)
```

```
.slideDown(2000);
```

### Autres effets:

```
.effect('explode'); // explosion de l'élément  
.effect('bounce',{times:3}, 500); // seconer de haut en bas  
.effect('shake'); // seconer de gauche à droite  
.effect('slide'); // Rentre en glissant  
.effect('drop'); // Sort en glissant  
.accordion({collapsible: true, active: false}); // Effet  
accordéon  
.draggable(); // Active le drag and drop  
.resizable(); // Active le redimensionnement  
.selectable(); // Rend la sélection possible  
.sortable(); // Active déplacement d'éléments d'une liste
```

## < WITH HTML >

### Get Content and Attributes

Three simple, but useful, jQuery methods for DOM manipulation are:

- ✓ text() - Sets or returns the text content of selected elements
- ✓ html() - Sets or returns the content of selected elements (including HTML markup)
- ✓ val() - Sets or returns the value of form fields

```
alert( $("#test").text() );  
// Affichera le contenu de #test sous forme de texte  
alert( $("#test").html() );  
// Affichera le contenu de #test sous forme de html (avec les  
balises)  
alert( $("#test").val() );  
// Affichera la valeur contenu dans #test  
alert( $("#w3s").attr("href") );  
// Affichera l'attribue (l'@) contenu dans #w3s
```

### Set Content and Attributes

```
$("#btn1").click(function() {  
    $("#test1").text("Hello world!");  
});  
$("#btn2").click(function() {  
    $("#test2").html("<b>Hello world!</b>");  
});  
$("#btn3").click(function() {  
    $("#test3").val("Dolly Duck");  
});  
// Set le texte, puis l'html, puis la valeur  
$("#w3s").attr({  
    "href": "http://www.w3schools.com/jquery",  
    "title": "W3Schools jQuery Tutorial"  
});  
// Modifier la valeur d'un lien et son nom
```

### Add Elements

```
$("p").append("Some appended text.");  
// insère l'élément spécifié en dernier enfant de l'élément cible  
$("p").prepend("Some prepended text.");  
// Insère l'élément spécifié en premier enfant de l'élément cible  
EX: function appendText() {  
    var txt1 = "<p>Text.</p>";
```

*// Create element with HTML*

```
var txt2 = $("

</p>").text("Text.");  
// Create with jQuery  
var txt3 = document.createElement("p");  
// Create with DOM  
txt3.innerHTML = "Text.";  
$("#p").append(txt1, txt2, txt3);  
// Append the new elements  
} // Insère plusieurs éléments


```

#### Before / After:

```
$("#img").before("Some text before");  
.before('<p>rgrgrg</p>'); // Insérer avant  
$("#img").after("Some text after");  
.after('<p>rgrg</p>'); // Insérer après  
EX: function afterText() {  
    var txt1 = "<b>I </b>";  
// Create element with HTML  
    var txt2 = $("<i></i>").text("love ");  
// Create with jQuery  
    var txt3 = document.createElement("b");  
// Create with DOM  
    txt3.innerHTML = "jQuery!";  
    $("#img").after(txt1, txt2, txt3);  
// Insert new elements after <img>  
}
```

#### Remove:

```
$("#div1").remove(); // Effacer un élément  
$("#div1").empty(); // Effacer les éléments enfants d'un élément  
$("#p").remove(".test, .demo"); // Efface tous les éléments <p>  
qui ont pour class est et demo
```

#### Modifier les classes CSS:

```
.addClass('nomClass'); // Ajoute une classe  
EX: $("#button").click(function() {  
    $("#h1, h2, p").addClass("blue");  
    $("#div").addClass("important blue");  
});  
.removeClass('nomClass'); // Supprime une classe  
EX: $("#button").click(function() {  
    $("#h1, h2, p").removeClass("blue");  
});  
.toggleClass("nomClass") // Ajoute et supprime la classe
```

#### Méthode css():

```
css("propertyname", "value");  
.css("border-radius", "10px"); // Ajout d'un paramètre CSS  
.css({"background-color": "yellow", "font-size":  
"200%"}); // Ajout de plusieurs paramètres CSS
```

#### Dimensions:

[http://www.w3schools.com/jquery/jquery\\_dimensions.asp](http://www.w3schools.com/jquery/jquery_dimensions.asp)

## < AJAX jQuery >

```
$("#div1").load("demo_test.txt");  
// Charge le fichier demo_test.txt et l'affiche dans le #div1  
$("#div1").load("demo_test.txt #p1");  
// Charge le fichier demo_test.txt et l'affiche dans le #div1  
// Mais ne charge QUE le #p1
```

#### jQuery \$.get() Method

*The \$.get() method requests data from the server with an HTTP GET request.*

*Syntax:*

```
$.get(URL, callback);
```

*EX:*

```
$("#button").click(function() {  
    $.get("demo_test.asp", function(data, status) {  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

#### jQuery \$.post() Method

*The \$.post() method requests data from the server using an HTTP POST request.*

*Syntax:*

```
$.post(URL, data, callback);
```

*EX:*

```
$("#button").click(function() {  
    $.post("demo_test_post.asp",  
    {  
        name: "Donald Duck",  
        city: "Duckburg"  
    },  
    function(data, status) {  
        alert("Data: " + data + "\nStatus: " + status);  
    });  
});
```

*Bonus:*

#### Mouvement aux flèches :

```
$(document).keydown(function(key) {  
    switch(parseInt(key.which, 10)) {  
        case 37: // On a appuyé sur la touche flèche gauche  
            $('img').animate({left: "-=10px"}, 'fast');  
            break;  
        case 38: // On a appuyé sur la touche flèche vers le haut  
            $('img').animate({top: "-=10px"}, 'fast');  
            break;  
        case 39: // On a appuyé sur la touche flèche droite  
            $('img').animate({left: "+=10px"}, 'fast');  
            break;  
        case 40: // On a appuyé sur la touche flèche vers le bas  
            $('img').animate({top: "+=10px"}, 'fast');  
            break;  
    }  
});
```