

3. jQuery Syntax

With jQuery you select (query) HTML elements and perform "actions" on them.

3.1. jQuery Syntax

The jQuery syntax is tailor made for **selecting** HTML elements and performing some **action** on the element(s).

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".
```

Are you familiar with CSS selectors?



jQuery uses CSS syntax to select elements. You will learn more about the selector syntax in the next chapter of this tutorial.

3.2. The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

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

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

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

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

Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.

3.3. jQuery and Other JavaScript Frameworks

What if you wish to use other frameworks on your pages, while still using jQuery?

As you already know, jQuery uses the \$ sign as a shortcut for jQuery.

What if other JavaScript frameworks also use the \$ sign as a shortcut?

Some other popular JavaScript frameworks are: MooTools, Backbone, Sammy, Cappuccino, Knockout, JavaScript MVC, Google Web Toolkit, Google Closure, Ember, Batman, and Ext JS.

Some of the other frameworks also use the \$ character as a shortcut (just like jQuery), and then you suddenly have two different frameworks using the same shortcut, which might result in that your scripts stop working.

The jQuery team have already thought about this, and implemented the noConflict() method.

a) The jQuery noConflict() Method

The noConflict() method releases the hold on the \$ shortcut identifier, so that other scripts can use it.

You can of course still use jQuery, simply by writing the full name instead of the shortcut:

Example

```
$.noConflict();  
jQuery(document).ready(function() {  
    jQuery("button").click(function() {  
        jQuery("p").text("jQuery is still  
working!");  
    });  
});
```

You can also create your own shortcut very easily. The `noConflict()` method returns a reference to jQuery, that you can save in a variable, for later use. Here is an example:

Example

```
var jq = $.noConflict();  
jq(document).ready(function() {  
    jq("button").click(function() {  
        jq("p").text("jQuery is still  
working!");  
    });  
});
```

If you have a block of jQuery code which uses the `$` shortcut and you do not want to change it all, you can pass the `$` sign in as a parameter to the `ready` method. This allows you to access jQuery using `$`, inside this function - outside of it, you will have to use "jQuery":

Example

```
$.noConflict();  
jQuery(document).ready(function($) {  
    $("button").click(function() {  
        $("p").text("jQuery is still  
working!");  
    });  
});
```

3.4. jQuery Misc Reference

For a complete overview of all jQuery Misc methods, please go to our [jQuery Misc Reference](#).