



# JQUERY



# 1

**Select elements**  
using CSS-style  
selectors



```
jQuery( 'li.hot' );
```



FUNCTION

```
jQuery( 'li.hot' );
```



CSS-STYLE SELECTOR



# SHORTHAND FOR JQUERY FUNCTION

```
$( 'li.hot' );
```



When you select an element or set of elements, it creates a **jQuery object**.

That object contains *references* to the elements.

Like any object, the jQuery object has properties and methods. They allow you to work with those elements.



# 1

**Select elements**  
using CSS-style  
selectors

# 2

**Do something**  
using methods of  
the jQuery object



# HIDE ALL MATCHING LIST ITEMS

```
$( 'li.hot' ).hide();
```



METHOD OF THE JQUERY OBJECT





# With jQuery, you can:

- Select or find elements

- Update their content / size / visibility

- Simplify event handling



# STORING SELECTIONS IN VARIABLES



When a variable holds a jQuery object, its name often begins with a \$ symbol.



```
var $titles = $( '.title' );
```



```
var $titles = $( '.title' );
```



**VARIABLE NAME**



**CSS-STYLE SELECTOR**



# CSS-STYLE SELECTORS

This variable holds a jQuery object containing the element whose `id` attribute has a value of `author`:

```
var $author = $( '#author' );
```



# GETTING & SETTING INFORMATION



# GETTING HTML CONTENT

The `.html( )` method gets the content of the selection (including markup).

It only retrieves content from the *first element* in the matched set.





# GETTING HTML CONTENT

This example retrieves the content of the first list item.

```
$ ( 'li' ) .html ( ) ;
```



# GETTING TEXT CONTENT

The `.text ( )` method gets the *text content only* of the selection (excluding markup).



# GETTING TEXT CONTENT

This example retrieves the text content of the first list item.

```
$ ( 'li' ) . text ( ) ;
```



# SETTING CONTENT

New content is added inside the parentheses after the method name.

It updates *all* of the elements in the matched set (not just the first). This is known as **implicit iteration**.



# SETTING HTML CONTENT

This example will replace the content of each list item with the word **Updated** in `<b>` tags.

```
$( 'li' ).html( ' <b>Updated</b>' );
```



# SETTING TEXT CONTENT

This example will replace the text content of each list item with the word **Updated**.

```
$( 'li' ).text( 'Updated' );
```



# CHAINING

It is possible to call multiple methods on the same selection.

```
$ ( 'li' ) .hide ( ) .fadeIn ( 500 ) ;
```



# CHAINING

It is possible to call multiple methods on the same selection.

```
$( 'li' ).hide( ).fadeIn( 500 ) ;
```

FIRST METHOD

SECOND METHOD





CHECKING A PAGE IS READY  
TO WORK WITH



jQuery's `.ready( )` method checks that the page is ready for your code to work with.



Checks to see if the page  
has loaded before...

Creates a jQuery object  
containing the entire page

...running the code inside  
the parentheses

**JQUERY OBJECT**

**JQUERY METHOD**

**CODE**

```
$(document).ready(function() {  
    // code goes here  
});
```



This is a shortcut for writing the  
.`ready()` method.

```
$(document).ready(function() {  
    // code goes here  
});
```

```
$(function() {  
    // code goes here  
});
```



# MORE ABOUT UPDATING ELEMENTS



# UPDATE ELEMENTS

<code>.html()</code>	<code>.text()</code>
<code>.replaceWith()</code>	<code>.remove()</code>

```
$('li#one').remove();
```



# INSERT ELEMENTS

`.before()`



`.after()`



`<li> item </li>`



`.prepend()`



`.append()`



# GET & SET ATTRIBUTES

<code>.attr()</code>	<code>.removeAttr()</code>
<code>.addClass()</code>	<code>.removeClass()</code>

```
$('a#top').addClass('button');
```





# UPDATE CSS

```
.css({  
  'font-family': 'Arial',  
  'color': '#ffffff',  
  'font-size': '+=2'  
});
```



# EACH ELEMENT IN A SELECTION

```
$( 'li' ).each(function() {  
    var ids = this.id;  
    $(this).append( ' ' + ids );  
});
```



# EVENTS



# EVENT METHODS

The `.on( )` method handles events:

```
$('li').on('click', function() {  
    // code to happen on click  
});
```



# EVENT METHODS

The type of event is the first argument:

```
$('li').on('click', function() {  
    // code to happen on click  
});
```



# EVENT METHODS

When the event happens, an anonymous function is called:

```
$('li').on('click', function() {  
    // code to happen on click  
});
```



# EVENTS

focus, blur, change, input,  
keydown, keyup, keypress,  
click, dblclick, mouseup,  
mousedown, mouseover,  
mousemove, mouseout, hover,  
submit, select, ready, load,  
unload, error, resize, scroll



# EVENT OBJECT

Passed into event handling function:

```
$( 'li' ).on( 'click', function(e) {  
    var eventType = e.type;  
} );
```





