Jquery - Quick Reference

Weight

jquery-1.4.4.js
jquery.validate.js
additional-methods.js
jquery-ui-1.8.7.custom.js
jquery-ui-1.8.7.custom.css

Dev (None minified, none compressed): 338KB
Testing (Most Minified, none compressed): 180KB
Production (Most minified, all compressed): 53 KB

See > Jquery Component Weight.xlsx

Script References

Development

```
  <script type="text/javascript"
src="script/jquery/jquery-
1.4.4.js"></script>
  <script
type="text/javascript"src="script/jquery/plu
gins/jquery-
validate/jquery.validate.js"></script>
```

Production

```
  <script type="text/javascript"
src="script/jquery/jquery-
1.4.4.min.js"></script>
  <script
type="text/javascript"src="script/jquery/plu
gins/jquery-
validate/jquery.validate.min.js"></script>
```

Basics

Launch code on document ready.

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

(Jquery Project > Tutorials:How jQuery Works)

Events

Add an event handler orthodox.

```
$(document).ready(function(){
   $("a").click(function(event){
     alert("Thanks for visiting!");
   });
 });
```

Add an event handler for large functions.

```
function targetOnClick () {
  // Do Stuff.
}

$(document).ready(function(){
   $("#target").click(targetOnClick);
 });
```

(Jquery Project > Tutorials:How jQuery Works)

Prevent default behavior

```
$(document).ready(function(){
   $("a").click(function(event){
     alert("As you can see, the link no
longer took you to jquery.com");
     event.preventDefault();
   });
 });
```

(Jquery Project > Tutorials:How jQuery Works)

Core

jquery() or $()

```
    * jQuery( selector, [ context ] )
          o jQuery( selector, [ context ] )
          o jQuery( element )
          o jQuery( elementArray )
          o jQuery( jQuery object )
          o jQuery()
    * jQuery( html, [ ownerDocument ] )
          o jQuery( html, [ ownerDocument ] )
          o jQuery( html, props )
    * jQuery( callback )
          o jQuery( callback )
```

http://api.jquery.com/jQuery/

Jquery(selector, [context]). Selector accepts a CSS, XPath, or Custom string to match a set of elements. Optionally limit the selection to **Context.**

```
// Example 1
$("#special").click(function(event){
  $('div', this).addClass("john");
}); // a.click.

// Example 2
$("input:text",
document.forms["frmMain"]).addClass("john");

// Example 3. Ajax
$("div", xml.responseXML);

//Example 4. Another Jquery object as
context.
$("p ~ div",
$("#special")).addClass("john");
```

http://api.jquery.com/jQuery/

jQuery(html, [ownerDocument]). Create DOM elements on the fly from a raw HTML String,

```
$("<p id='nice'>The cow is
<strong>hefty</strong>").appendTo("body");
```

http://api.jquery.com/jQuery/#jQuery2

jQuery(callback). Behaves like $(document).ready(), which you should use instead, for readability.

Reference DOM object

To retrurn a reference to a dom object, rather than a jquery object, use $(selector)[0]

```
$('.fieldUserHelp').parent().children(':inpu
t').focus(function() {
        $(this).css('display', 'block');
        alert($(this)[0].id);
      }); // hover
```

innerHTML

Use .html() as a replacement for .innerHTML =

```
// Native javascript method
document.getElementById("message").innerHTML
= "Hello world, Javascript";

// jquery method
$("#message").html("Hello world, jquery.");
```

Note that the javascript method doesn't use a hash, #, to reference the ID.

Class

Add and remove classes

```
("#special p").addClass("john");
("#special p").removeClass("john");
```

Special Effects

Effects

.animate()	Perform a custom animation of a set of CSS properties.
.clearQueue()	Remove from the queue all items that have not yet been run.
.delay()	Set a timer to delay execution of subsequent items in the queue.
.dequeue()	Execute the next function on the queue for the matched elements.
.fadeIn()	Display the matched elements by fading them to opaque.
.fadeOut()	Hide the matched elements by fading them to transparent.
.fadeTo()	Adjust the opacity of the matched elements.
.fadeToggle()	Display or hide the matched elements by animating their opacity.
jQuery.fx.interval	The rate (in milliseconds) at which animations fire.
jQuery.fx.off	Globally disable all animations.
.hide()	Hide the matched elements.
.queue()	Show the queue of functions to be executed on the matched elements.
.show()	Display the matched elements.
.slideDown()	Display the matched elements with a sliding motion.
.slideToggle()	Display or hide the matched elements with a sliding motion.
.slideUp()	Hide the matched elements with a sliding motion.
.stop()	Stop the currently-running animation on the matched elements.
.toggle()	Display or hide the matched elements.

Effect Code Example

```
$("a").click(function(event){
  event.preventDefault();
  $("#target").fadeToggle();
});
```

Animate

Basic animation

```
$("a").click(function(event){
  event.preventDefault();

  var resizeInterval = 20;

  // Can also be a number, in milliseconds,
eg 3000.
  var duration = 'fast';
  $("#target").animate(
    {
      height: '+=' + resizeInterval,
      paddingTop: '+=' + resizeInterval
    },
    duration,
    function() {
      // Animation complete.
      $("#message").html("Hello world,
jquery.");
    }
  ); // animate
}); // a.click.
```

.animate(properties, [duration], [easing], [callback])

properties A map of CSS properties that the animation will move toward.

duration A string or number determining how long the animation will run.

easing A string indicating which easing function to use for the transition.

callback A function to call once the animation is complete.

http://api.jquery.com/animate/

Menu item animation

See C:\Data\Sda\Code\EcmaScript\Libraries\Packaged\jqueryDemo\bubbleMenu.html

Selectors

Selectors are modelled after CSS, XPath, with some custom selectors.

Basic

All Selector ("*")	Selects all elements.
Class Selector (".class")	Selects all elements with the given class.
Element Selector ("element")	Selects all elements with the given tag name.
ID Selector ("#id")	Selects a single element with the given id attribute.
Multiple Selector ("selector1, selector2, selectorN")	Selects the combined results of all the specified selectors.

```
$(".row, #special, p")
```

Hierarchy

Child Selector ("parent > child")	Selects all direct child elements specified by "child" of elements specified by "parent".
Descendant Selector ("ancestor descendant")	Selects all elements that are descendants of a given ancestor.
Next Adjacent Selector ("prev + next")	Selects all next elements matching "next" that are immediately preceded by a sibling "prev".
Next Siblings Selector ("prev ~ siblings")	Selects all sibling elements that follow after the "prev" element, have the same parent , and match the filtering "siblings" selector.

```
$("legend + .row")
```

Basic Filter

:animated Selector	Select all elements that are in the progress of an animation at the time the selector is run.
:eq() Selector	Select the element at index n within the matched set.
:even Selector	Selects even elements, zero-indexed. See also odd.
:first Selector	Selects the first matched element.
:gt() Selector	Select all elements at an index greater than index within the matched set.
:header Selector	Selects all elements that are headers, like h1, h2, h3 and so on.
:last Selector	Selects the last matched element.
:lt() Selector	Select all elements at an index less than index within the matched set.
:not() Selector	Selects all elements that do not match the given selector.
:odd Selector	Selects odd elements, zero-indexed. See also even.

```
$(":header, .row:even")
```

Attribute

Attribute Contains Prefix Selector [name|="value"]	Selects elements that have the specified attribute with a value either equal to a given string or starting with that string followed by a hyphen (-).
Attribute Contains Selector [name*="value"]	Selects elements that have the specified attribute with a value containing the a given substring.
Attribute Contains Word Selector [name~="value"]	Selects elements that have the specified attribute with a value containing a given word, delimited by spaces.
Attribute Ends With Selector [name$="value"]	Selects elements that have the specified attribute with a value ending exactly with a given string. The comparison is case sensitive.
Attribute Equals Selector [name="value"]	Selects elements that have the specified attribute with a value exactly equal to a certain value.
Attribute Not Equal Selector [name!="value"]	Select elements that either don't have the specified attribute, or do have the specified attribute but not with a certain value.
Attribute Starts With Selector [name^="value"]	Selects elements that have the specified attribute with a value beginning exactly with a given string.
Has Attribute Selector [name]	Selects elements that have the specified attribute, with any value.
Multiple Attribute Selector [name="value"][name2="value2"]	Matches elements that match all of the specified attribute filters.

```
$("[type*='text']")
```

Child Filter

:first-child Selector	Selects all elements that are the first child of their parent.
:last-child Selector	Selects all elements that are the last child of their parent.
:nth-child() Selector	Selects all elements that are the nth-child of their parent.
:only-child Selector	Selects all elements that are the only child of their parent.

```
$("p:first-child")
```

Content Filter

:contains() Selector	Select all elements that contain the specified text. Case Sensitive.
:empty Selector	Select all elements that have no children (including text nodes).
:has() Selector	Selects elements which contain at least one element that matches the specified selector.
:parent Selector	Select all elements that are the parent of another element, including text nodes.

```
$("p:contains('laoreet')")
```

Visibility Filter

:hidden Selector	Selects all elements that are hidden.
:visible Selector	Selects all elements that are visible.

```
$("p:visible")
```

Form

:button Selector	Selects all button elements and elements of type button.
:checkbox Selector	Selects all elements of type checkbox.
:checked Selector	Matches all elements that are checked.
:disabled Selector	Selects all elements that are disabled.
:enabled Selector	Selects all elements that are enabled.
:file Selector	Selects all elements of type file.
:image Selector	Selects all elements of type image.
:input Selector	Selects all form controls.
:password Selector	Selects all elements of type password.
:radio Selector	Selects all elements of type radio.
:reset Selector	Selects all elements of type reset.
:selected Selector	Selects all elements that are selected.
:submit Selector	Selects all elements of type submit.
:text Selector	Selects all elements of type text.

```
$(":input")
```

Event Binding

You must reference the radio buttons within the fieldset to avoid the event being triggered twice.

```
   $('#fsetPaymentType
:radio').click(fsetPaymentTypeOnClick);
```

Test

Determine whether radiobutton, in a group, is checked

```
$('#chkSponsorshipOfSupporter')[0].checked

#chkSponsorshipOfSupporter is the id of the
checkbox.

// decprcated:
if($('#ReceiptNo:checked').val()) { .. . }
```

#ReceiptNo is the id of the radiobuttion.

Determine whether a particular radio button is checked

```
$('#PaymentMethodCheque')[0].checked
```

Determine whether a particular radio button is enabled

```
$('#PaymentMethodCheque')[0].disabled
```

Determine whether a group of controls are enabled

```
$('#fsetPaymentMethod
:input:enabled').length > 0
```

Disable and Hide

Disable/enable a control

```
targetFieldset.attr('disabled', 'disabled');
targetFieldset.removeAttr('disabled');
```

Hide/Show a control

```
targetFieldset.hide();
targetFieldset.show();
```

Set Values

Set value of an input

```
$('#FullName').val('Mary');
```

Plugins

Validation

Essential Links

Home Page:
http://bassistance.de/jquery-plugins/jquery-plugin-validation/

Documentation:
http://docs.jquery.com/Plugins/Validation

Library

Development

```
<script type="text/javascript"
src="script/jquery/plugins/jquery-
validate/jquery.validate.js"></script>
```

Production

```
<script type="text/javascript"
src="script/jquery/plugins/jquery-
validate/jquery.validate.min.js"></script>
```

HTML Preparations

Controls. Add a label element with a for attribute. The control element must have a name attribute.

```
<label for="txtFirstName">FirstName</label>
<input id="txtFirstName" name="txtFirstName"
/>
```

CSS Preparations

By default when an error message is triggered it will be inserted as a label with class="error" after the control. Style it appropriately.

```
// HTML with generated error
<label for="txtFirstName">First Name</label>
<input type="text" id="txtFirstName"
name="txtFirstName" class="required" />
<label for="txtFirstName" generated="true"
class="error">This field is
required.</label>

// Example CSS.
label.error
{
  float: none;
  color: red;
  margin-left: 0.5em;
}

label.error:after
{
  content: "";
}
```

Basic

Call

```
$(document).ready(function(){
  $("#frmMain").validate();
}); // ready
```

Reset

Html

```
<input type="reset" id="cmdReset"
value="reset" />
```

Javascript

```
$(document).ready(function(){
  var validator = $("#frmMain").validate();

  $(":reset").click(function(event) {
    validator.resetForm();
  }); // reset.click
}); // ready
```

Validation Rules and Methods

Validation methods and validation rules.

* method: A validation method implements the logic to validate an element, like an email method that checks for the right format of an text input's value. A set of standard methods is available, and it is easy to write your own.

* rule: A validation rule associates an element with a validation method, like "validate input with name "primary-mail" with methods "required" and "email".

http://docs.jquery.com/Plugins/validation#Options_for_the_validate.28.29_method

One validation rule, for an element, may specify many validation methods.

Validation Methods, Built-in

required( )	Makes the element always required.
required(dependency-expression)	Makes the element required, depending on the result of the given expression.
required(dependency-callback)	Makes the element required, depending on the result of the given callback.
remote(options)	Requests a resource to check the element for validity.
minlength(length)	Makes the element require a given minimum length.
maxlength(length)	Makes the element require a given maxmimum length.
rangelength(range)	Makes the element require a given value range.
min(value)	Makes the element require a given minimum.
max(value)	Makes the element require a given maximum.
range(range)	Makes the element require a given value range.
email( )	Makes the element require a valid email
url( )	Makes the element require a valid url
date( )	Makes the element require a date.
dateISO( )	Makes the element require a ISO date.
dateDE( )	Makes the element require a german date.
number( )	Makes the element require a decimal number.
numberDE( )	Makes the element require a decimal number with german format.
digits( )	Makes the element require digits only.
creditcard( )	Makes the element require a creditcard number.
accept(extension)	Makes the element require a certain file extension.
equalTo(other)	Requires the element to be the same as another one.

All these return a Boolean.

http://docs.jquery.com/Plugins/validation#Validator

Add Validation Rules And Methods

Several techniques to add validation rules (and validation methods):

Rules option of the validate() method. Rule key in key/value pair refers to the element name (not id).

```
// Basic
$("#frmMain").validate({
  rules: {
    // simple rule, converted to
{required:true}
    txtFirstName: "required",

    // Compound rule, with many methods
    txtBirthDate: "required dateISO",

    // compound rule, with many methods
    txtEmail: {
      required: true,
      email: true
    }
  } // rules
}); // validate

// Specifies a contact element as required
and as email address, the latter depending
on a checkbox being checked for contacting
via email.
$(".selector").validate({
   rules: {
     contact: {
       required: true,
       email: {
         depends: function(element) {
           return
$("#contactform_email:checked")
         }
       }
     }
   }
})
```

http://docs.jquery.com/Plugins/Validation/validate#options

Rule method of jquery selected element.

```
$("#frmMain").validate();

// Must be called after .validate()
$("#txtQuantity").rules("add", {
  required: true,
  digits: true,
  range: [10, 20]
}); // #quantity.rules
```

Via classes of an element (for validation methods without arguments, eg 'required', dateISO).

```
<input type="text" id="txtQuantity"
name="txtQuantity" class="required digits"
/>
```

Via attributes of an element (but produces invalid HTML, and is available only for validation methods with one argument).

```
<input type="text" id="txtQuantity"
name="txtQuantity" minlength="4" />
```

Custom Method Messages (Element independent)

```
// overwrite default messages for methods
$.extend($.validator.messages, {
  required: "Fields is mothefuck'n'
required.",
  digits: "Must be digits, fucker."
});
```

Custom Rule and Method Messages (For particular Elements)

Custom rule and method messages (for particular elements) override custom method messages (element independent).

Override the default rule message with a custom message. Do this via validate() option or .rules("add", …) of jquery selected element.

```
$("#frmMain").validate({
  rules: {
    // simple rule, converted to
{required:true}
    txtFirstName: "required",

    // compound rule, with many methods
    txtEmail: {
      required: true,
      email: true
    },
    txtQuantity: {
      required: true,
      digits: true,
      range: [10,20]
    }
  }, // rules
  messages: {
    // Custom message for all methods in a
validation rule
    txtEmail: "For fuck's sake get it
right",

     // Custom messages for each method in a
validation rule
    txtQuantity: {
      required: "You need to enter a
Quantity",
      digits: "Funk on with digits only",

      // Call back
      range: jQuery.format("Between {0} and
{1} only, motherfucker.")    }
  }
}); // validate
```

http://docs.jquery.com/Plugins/Validation/validate#options

Message Callbacks. Each message value (in the key/value pair) can be a String or a Callback. The callback is called in the scope of the validator and with the rule's parameters as the first and the element as the second argument, it must return a String to display as the message.

```
messages: {
  txtQuantity: {
    // Call back
    range: jQuery.format("Between {0} and
{1} only, motherfucker.")
  }
}
```

http://docs.jquery.com/Plugins/Validation/validate#options

Additional Validation Methods

Add reference to additional-methods.js

```
<script type="text/javascript"
src="script/jquery/plugins/jquery-
validate/additional-methods.js"></script>
```

maxWords	
minWords	
rangeWords	
letterswithbasicpunc	
alphanumeric	
lettersonly	
nowhitespace	
ziprange	
vinUS	
dateITA	Return true, if the value is a valid date, also making this formal check dd/mm/yyyy. @example jQuery.validator.methods.date (01/01/1900) @result true @example jQuery.validator.methods.date (01/13/1990) @result false @example jQuery.validator.methods.date (01.01.1900) @result false @example <input name=pippo class={dateITA:true} /> @desc Declares an optional input element whose value must be a valid date. @name jQuery.validator.methods.date ITA @type Boolean @cat Plugins/Validate/Methods
dateNL	
time	
phone	matches US phone number format where the area code may not start with 1 and the prefix may not start with 1 allows '-' or ' ' as a separator and allows parens around area code some people may want to put a '1' in front of their number 1(212)-999-2345 or 212 999 2344 or 212-999-0983 but not 111-123-5434 and not 212 123 4567
strippedminlength	TODO check if value starts with <, otherwise don't try stripping anything
email2	same as email, but TLD is optional
url2	same as url, but TLD is optional
creditcardtypes	Modified version of Castle.Components.Validator. CreditCardValidator Redistributed under the the Apache License 2.0 at http://www.apache.org/licenses/LICENSE-2.0 Valid Types: mastercard, visa, amex, dinersclub, enroute, discover, jcb, unknown, all (overrides all other settings)

Custom Validation, Single Field Simple

Use Validator addMethod(name, method, [message]) to test for a simple condition.

```
var validator = $("#frmMain").validate({
  rules: {
    txtNotes : {
      required: true,
      hasJohnString: true,
    }
  }
}); // validate

$.validator.addMethod("hasJohnString",
function (value, element) {
  return this.optional(element) ||
/.*(john).*/.test(value);
}, "Your field must contain 'john'.");
```

Use Validator addMethod(name, method, [message]) to test for a complex condition (multiple parameters)

```
var validator = $("#frmMain").validate({
  rules: {
    txtQuantity: {
      required: true,
      digits: true,
      sumCheck: [4, 5]
    }
  }
}); // validate

$.validator.addMethod("sumCheck", function
(value, element, params) {
  return this.optional(element) || value ==
params[0] + params[1];
}, $.format("Your field must be the sum of
{0} and {1}."));
```

Custom Validation, Single field dependent on other fields.

Within Validator addMethod(name, method, [message]) reference other fields

```
var validator = $("#frmMain").validate({
  rules: {
    txtQuantity: {
      required: true,
      digits: true,
      sumFieldsCheck: true
    }
  }
}); // validate


$.validator.addMethod("sumFieldsCheck",
function (value, element, params) {
  return this.optional(element) ||
    value ==
parseInt($("#txtMagicNumber01").val())
           +
parseInt($("#txtMagicNumber02").val());
}, "Your field must be the sum of
txtMagicNumber01 and txtMagicNumber02.");
```

Highlight Invalid Fields

Define an error class in your stylesheet

```
/* An "error" class gets added by jQuery
validator plugin, both to the invalid field
and the generated error label */
.error
{
   border: 1px solid red;
}

/* You may have to override declarations in
the .error class */
label.error, label.info
{
  float: none;
  color: red;
  margin-left: 0.5em;
  border: none;
}
```

Group validation methods into one validation method (for use on several elements)

Use $.validator.addClassRules and add this definition to the class of the relevant fields.

```
var validator = $("#frmMain").validate();

$.validator.addClassRules({
  personName: {
    required: true,
    minlength: 2
  }
});

<input type="text" id="txtFirstName"
name="txtFirstName" class="personName" />
<input type="text" id="txtLastName"
name="txtLastName" class="personName" />
```

Once a class rule is defined you can only add the method via a HTML class. That is, you can't use the validate() rule option nor an element .rules() method.

http://docs.jquery.com/Plugins/Validation/Validator/addClassRules

Custom Validation, Multifield (aka Form Level Validation)

Input Mask Plugin?

Reference Example

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.1//EN"

"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dt
d">
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en">
<head>
  <meta http-equiv="content-type"
content="text/html; charset=utf-8" />
  <meta http-equiv="content-script-type"
content="text/javascript" />


  <title>Jquery Demo</title>

  <meta name="author" content="John Bentley"
/>
  <meta name="description" content="Created
by John Bentley to explore features of
JQuery." />

  <link type="text/css" rel="stylesheet"
href="style/softmake.css" />

  <script type="text/javascript"
src="script/jquery/jquery-
1.4.3.min.js"></script>
  <script
type="text/javascript"src="script/jquery/plu
gins/jquery-
validate/jquery.validate.min.js"></script>



  <script type="text/javascript">
  /* <![CDATA[ */
      $(document).ready(function(){
         $("a").click(function(event){
           alert("As you can see, the link
no longer goes through.");
           event.preventDefault();
         });
       });

  /* ]]> */
  </script>
</head>
<body>

  <div id="contentArea">
    <div class="buffer">
      <div class="header">
        <h1>Jquery Demo</h1>
      </div>
      <p><a
href="http://www.softmake.com.au">Softmake</
a> Lorem ipsum </p>


    </div><!-- buffer -->
  </div><!-- contentArea -->
</body>
</html>
```

Sources

Jquery Project > Tutorials:How jQuery Works. (n.d.). *Tutorials:How jQuery Works.* Retrieved Nov 11, 2010, from jquery: http://docs.jquery.com/How_jQuery_Works

Document Licence