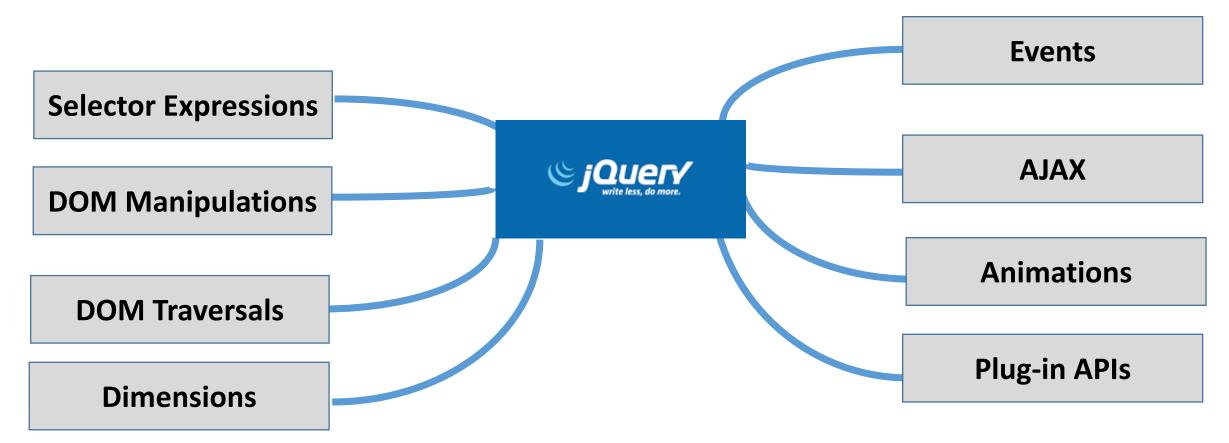


Karthik Raman

- jQuery is a fast, small, and feature-rich JavaScript library
- Lot more easier and simpler than vanilla javascript with an easy-to-use API
- Cross Browser Support





- ❖ Version 1.0: First stable release. Support for IE 6-8
- Version 2.0: Dropped IE 6-8 support for performance improvements and reduction in file size
- Version 3.0: Advance Javascript features jQuery.Deferred objects compatibility with Promises/A+ \$.then and \$.when

jQuery Migrate Plugin





• jQuery UI is a collection of GUI widgets, animated visual effects, and themes implemented with jQuery, Cascading Style Sheets, and HTML



❖ jQuery Mobile is a user interface framework, built on jQuery Core and used for developing responsive websites or applications that are accessible on mobile, tablet, and desktop devices. It uses features of both jQuery and jQueryUI to provide API features for mobile web applications



❖ Local Installation – You can download jQuery library on your local machine and include it in your HTML code.

```
<script src= "js/jquery-1.12.4.min.js"></script>
```

CDN Based Version – You can include jQuery library into your HTML code directly from Content Delivery Network (CDN)

Google and Microsoft provides content deliver for the latest version

```
<script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js">
</script>
```

jq1

#### Window.onload()



```
<script type="text/javascript">
  window.onload = function ()
    if (document.addEventListener) // For all modern browsers
      document.getElementById('btn').addEventListener('click', clickHandler, false);
                // For Internet Explorer < 9
    else
      document.getElementById('btn').attachEvent('onclick', clickHandler);
    function clickHandler()
      alert('Clayfin');
</script>
<input type="button" value="Click" id="btn" />
```



```
<script src= "js/jquery-1.12.4.min.js"></script>
```

```
jq2
```

```
<script type="text/javascript">
    $(document).ready(function () {
        $('#btn').click(function () {
            alert('Clayfin');
        });
    });
</script>
<input type="button" value="Click" id="btn" />
```

- \$ is a shortcut for jQuery.
- ready() function ensures that the DOM is fully loaded.
- \$(document).ready( handler ) or \$(handler )



```
<script src= "js/jquery-1.12.4.min.js"></script>
```

```
jq3
```

```
<script type="text/javascript">
    $('#btn').click(function () {
        alert('Clayfin');
    });
</script>
<input type="button" value="Click" id="btn" />
```

- This code will not work
- Script Code executes before the button element



```
<script src= "js/jquery-1.12.4.min.js"></script>
```

```
jq4
```

```
<script type="text/javascript">
    $(document).ready(function () {
        $('#btn').click(function () {
            alert('Clayfin');
        });
    });
</script>
<input type="button" value="Click" id="btn" />
```

- \$(document).ready jQuery event is fired as soon as the DOM is loaded and ready to be manipulated by script
- scripts can safely access elements in the html DOM





```
<script src= "js/jquery-1.12.4.min.js"></script>
```

```
jq5
```

```
$(window).load(function () {
    alert("Window Load");
});

$(document).ready(function () {
    alert("DOM Ready");
});
```

- \$(document).ready is fired before \$(window).load
- \$(window).load event fires when the DOM loaded and all the content on the page (images, css etc) is fully loaded
- \$(document).ready even is fired DOM is loaded but before all the images, css etc.. are fully loaded.



# selector.css ( PropertyName, PropertyValue );

```
css( name )

Return a style property on the first matched element

css( name, value )

Set a single style property to a value on all matched elements

css( properties )

Set a key/value object as style properties to all matched elements
```





**Selector Expressions** 







• jQuery Selector makes use of expressions to find out matching elements from a DOM based on the given criteria

Selector	Description
Tag Name	Represents a tag name available in the DOM.  For example \$('p') selects all paragraphs  in the document
Tag ID	Represents a tag available with the given ID in the DOM.  For example \$('#myid') selects the single element in the document that has an ID of myid
Tag Class	Represents a tag available with the given class in the DOM.  For example \$('.myclass') selects all elements in the document that have a class of myclass

jq6



```
Syntax: $('TagName')
 $('a') // Selects all anchor elements
 $('div p') // Select all paragraph elements that are descendants of div
 $('div, p, span, a') // Selects all div, paragraph, span and anchor elements
 <script type="text/javascript">
   $(document).ready(function () {
     $('#btn').click(function () {
        console.log($('span').length);
       $('span').css({ 'color': 'red', 'font-size': '150%',
                                       'background-Color': 'Aqua'});
      }); </script>
 <span>Clayfin</span>
  <span> Vayana </span>
 <input type="button" value="Click" id="btn" />
```

Syntax: \$('#id') - find an HTML element by the given ID jQuery #id selector returns only the first element, if you have 2 or more elements with the same ID

```
<script type="text/javascript">
  $(document).ready(function () {
    $('#btn').click(function () {
     $('#myspan').css({ 'color': 'red', 'font-size': '150%',
                                    'background-Color': 'Aqua'});
    }); });
</script>
<span id="myspan">Clayfin </span>
<span id="myspan">Vayana </span>
<input type="button" value="Click" id="btn" />
```



```
Syntax: $('.class')
$('.myclass') // Selects all elements with class myclass
<script type="text/javascript">
  $(document).ready(function () {
    $('#btn').click(function () {
      $('.myclass').css({ 'color': 'red', 'font-size': '150%',
                                     'background-Color': 'Aqua'});
    }); });
</script>
<span class="myclass">Clayfin </span>
<span class="myclass">Vayana </span>
<input type="button" value="Click" id="btn" />
```

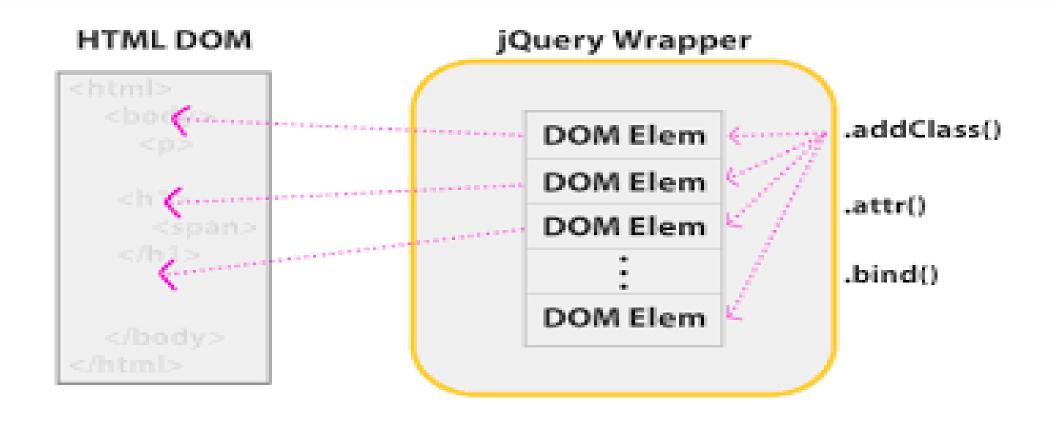
jq8

jQuery TagName selector uses document.getElementsByTagName() jQuery id selector uses document.getElementById() jQuery class selectors uses document.getElementsByClassName()

**jQuery selector functions** returns a jQuery object that wraps the DOM object and provides jQuery methods jQuery methods like css(), html() click() bind(), attr() etc on the object returned by jQuery

To get the underlying DOM object from a jQuery object write \$('#id')[0]









#### Syntax:

```
$('[attribute]')
$('[attribute="value"]')
```

```
$('[title]') // Select all elements that has title attribute
$('[title="mytitle"]') // Selects all elements that has title attribute value mytitle
$('div[title]') // Selects all div elements that has title attribute
$('div[title="mytitle"]') // Selects all div elements that has title attribute value mytitle
```





#### Syntax:

```
Attribute Value Equals Selector $('[name="value"]')
Attribute Value Not Equal Selector $('[name!="value"]')
Attribute Value Starts With Selector $('[name^="value"]')
Attribute Value Ends With Selector $('[name*="value"]')
Attribute Value Contains Selector $('[name*="value"]')
Attribute Value Contains Word Selector $('[name^="value"]')
Attribute Value Contains Prefix Selector $('[name]="value"]')
```







#### Syntax:

Get Attribute Value selector.attr(name)

Set Attribute Value selector.attr(name,value)

Remove Attribute selector.removeAttr(name)



hasClass( class ) - Returns true if the specified class is present on at least one of the set of matched elements

addClass( class ) - Adds the specified class(es) to each element in the set of matched elements

removeClass( class ) - Removes the specified class(es) from the set of matched elements toggleClass( class ) - Adds the specified class(es) if it is not present, removes the specified class if it is present





html() - Get the html contents (innerHTML) of the first matched element. html( htmlcontent) - Set the html contents of every matched element. text() - Get the combined text contents of all matched elements. text( string ) - Set the text contents of all matched elements. val() - Get the input value of the first matched element. val(value) - Set the value attribute of every matched element if it is called on <input> on <select> with the passed <option> value then option would be selected on check box or radio box then all the matching check box and radio box would be checked







# \$(selector).each ()

### jQuery each function is used to iterate over the items in a collection

Each - function with no arguments

Each - function with index as argument

Each – function with index and Element as argument

Exit Each – function with return false

Implicit Iterations





### jQuery each functions

\$(selector).each () is used to iterate over the items in a collection

\$.each() is used to iterate over javascript arrays any objects







# (selector).map() is used to iterate over the items in a collection

# \$.map() is used to iterate over javascript arrays any objects

\$.each	\$.map
each method is an immutable iterator	map method can be used as an iterator.
Returns the original array	Returns a new array
The order of callback arguments - index, element	The order of callback arguments - element, index
\$.each(elements, function () { index, element })	\$.map(elements, function () { element, index })
Return false to terminate the iteration	Does not have a way to terminate the iteration





**jQuery method chaining** allows us to run multiple jQuery commands, one after the other, on the same element(s)

```
$("li").css("font-size", "150%")
.css({'border': '3px solid red','backgroundColor': blue'})
.slideUp(2000)
.slideDown(2000);
```

Method chaining will not work if a method in the chain does not return an object

Example .text() returns the string that contains the text and NOT the object





# \$(selector).filter(criteria)

filter out all the elements that do not match the selected criteria and those matches will be returned

criteria: It specifies a selector expression, a jQuery object or one or more elements to be returned from a group of selected elements

### \$(selector).filter(function(index))

**function(index)**: It specifies a function to run for each element in the set. If the function returns true, the element is kept. Otherwise, it is removed **index**: index position of the element in the set





\$('input') selects elements with an input tag

\$('input[type="text"]') selects elements with an input tag with type text

\$(':input') selects elements with an input tag , Select, Buttons, Textarea

\$(':input[type="text"]') selects elements with an input tag with type text





\$('input[type="radio"]:checked) selects checked elements of the Radio Button

\$('input[type="checkbox"]) selects elements of the Checkboxes

\$('input[type=" checkbox"]:checked) selects checked elements of the Checkboxes

\$('input[name="grpName"]:checked) selects checked elements of the Radio Button or CheckBox with has the Group Name as grpName





\$(: enabled) selects all elements that are enabled
\$('input:enabled') selects all input elements that are enabled
\$('input[type="text"]:enabled) selects all input elements with type text that are enabled

\$(: disabled) selects all elements that are disabled
\$('input:disabled') selects all input elements that are disabled
\$('input[type="text"]:disabled) selects all input elements with type text that are
disabled





### jQuery's Pseudo-class Selectors

a colon-prefixed keyword added to a selector that specifies a special state of the selected element

```
jQuery's own pseudo-class selectors include :button, :checkbox, :eq(), :even, :file, :first, :gt(), :has(), :header, :hidden, :image, :input, :last, :lt(), :odd, :parent, :password, :radio, :reset, :selected, :submit, :text, and :visible
```



### Sizzle A pure-JavaScript CSS selector engine

jQuery uses Sizzle selector engine internally



#### **Selector Features**

- \* CSS 3 Selector support
- \* Full Unicode support
- \* Escaped selector support #id\:value
- \* Contains text :contains(text)
- \* Complex :not :not(a#id)
- \* Multiple :not :not(div,p)
- \* Not attribute value [name!=value]
- \* Has selector :has(div)
- \* Position selectors :first, :last, :even, :odd, :gt, :lt, :eq
- \* Easy Form selectors: input,:text,:checkbox,:file,:password,:submit,etc
- \* Header selector :header





**DOM Manipulations** 





html() - Get the html contents (innerHTML) of the first matched element. html(value) - Set the html contents of every matched element. text() - Get the combined text contents of all matched elements. text(value) - Set the text contents of all matched elements. val() - Get the input value of the first matched element. val(value) - Set the value attribute of every matched element if it is called on <input> on <select> with the passed <option> value then option would be selected on check box or radio box then all the matching check box and radio box would be checked





\$(selector). append(content) Append content to the inside of every matched element \$(selector). prepend(content) Prepend content to the inside of every matched element \$(selector). before(content) Insert content before each of the matched elements \$(selector). after(content) Insert content after each of the matched elements



```
<div>
   <label>This is div.</label>
</div>
       This is paragaph.
>
```

```
$('div').prepend('This is prepended paragraph');
                               $('div').before('This is new paragraph');
             This is new paragraph
             <div>
                This is prepended paragraph
                 <label>This is div.</label>
                This is appended paragraph
             </div>
             This is new paragraph
             This is paragaph.
                                $('div').after('This is new paragraph');
$('div').append('This is appended paragraph');
```

jq24



\$(selector). appendTo( selector ) Append all of the matched elements to another, specified, set of elements

\$(selector). prependTo( selector ) Prepend all of the matched elements to another, specified, set of elements

\$(selector). insertAfter( selector ) Insert all of the matched elements after another, specified, set of elements

\$(selector). insertBefore( selector ) Insert all of the matched elements before another, specified, set of elements





replaceWith( content ) Replaces all matched elements with the specified HTML or DOM elements

replaceAll( selector ) Replaces the elements matched by the specified selector with the matched elements

remove() Removes all matched elements from the DOM

detach() Same as remove() but retains the jQuery metadata and events

empty() Remove all child nodes from the set of matched elements







# wrap( html ) Wrap each matched element with the specified HTML content

```
<div class="containerDiv">
                                                                      <div id="div1">
                                                                           DIV 1
<div id="div1">
                                                                      </div>
    DIV 1
                                                                  </div>
</div>
                                                                  <div class="containerDiv">
<div id="div2">
                 $('div').wrap('<dix class="containerDiv"></div>');
                                                                      <div id="div2">
    DIV 2
                                                                           DIV 2
</div>
                                                                      </div>
<div id="div3">
                                                                  </div>
    DIV 3
                                                                  <div class="containerDiv">
</div>
                                                                      <div id="div3">
                                                                           DIV 3
                                                                      </div>
                                                                  </div>
```

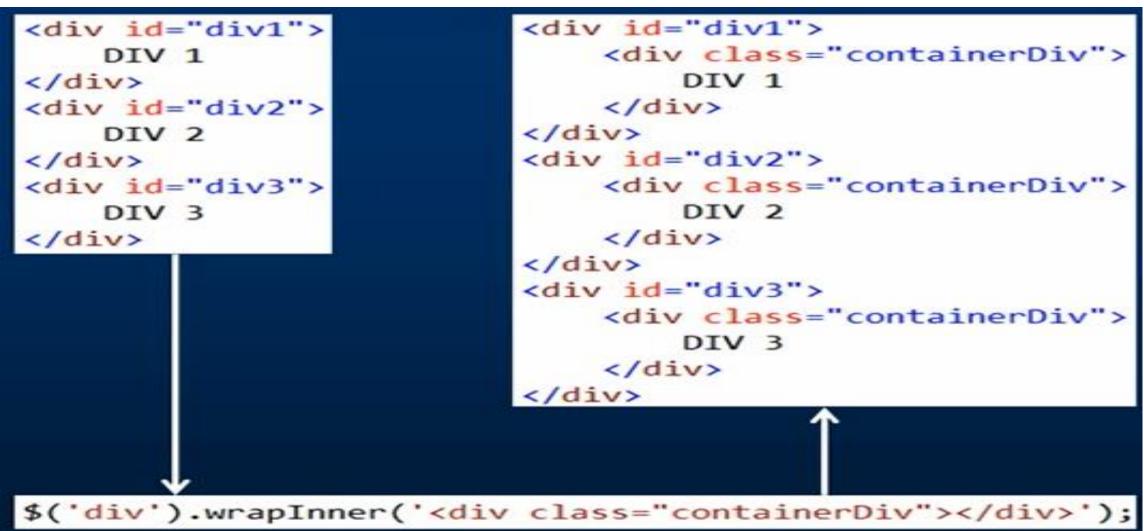


wrapAll( html ) Wrap all the elements in the matched set into a single

```
wrapper element
                    <div id="div1">
                        DIV 1
                    </div>
                    <div id="div2">
                        DIV 2
                    </div>
                    <div id="div3">
                        DIV 3
                    </div>
$('div').wrapAll('<div class="containerDiv"></div>');
              <div class="containerDiv">
                   <div id="div1">
                       DIV 1
                   </div><div id="div2">
                       DIV 2
                   </div><div id="div3">
                       DIV 3
                   </div>
              </div>
```



wrapInner( html ) Wrap the inner child contents of each matched element (including text nodes) with an HTML structure





unwrap() to remove the parent element of the selected elements





clone() Clone matched DOM Elements and select the clones
clone(bool) Clone matched DOM Elements, and all their event handlers,
and select the clones





eq(index) filters the set of matched elements to a single element

filter( selector ) Removes all elements from the set of matched elements that do not match the specified selector(s)

is( selector ) Checks the current selection against an expression and returns true, if at least one element of the selection fits the given selector

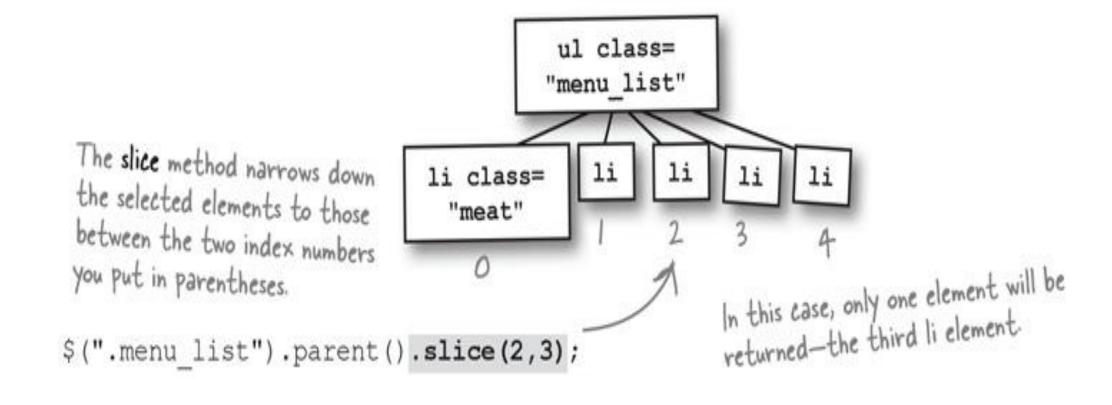
**not( selector )** Removes elements matching the specified selector from the set of matched elements

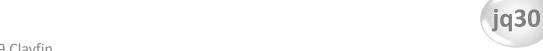






# slice(start, [end]) Selects a subset of the matched elements









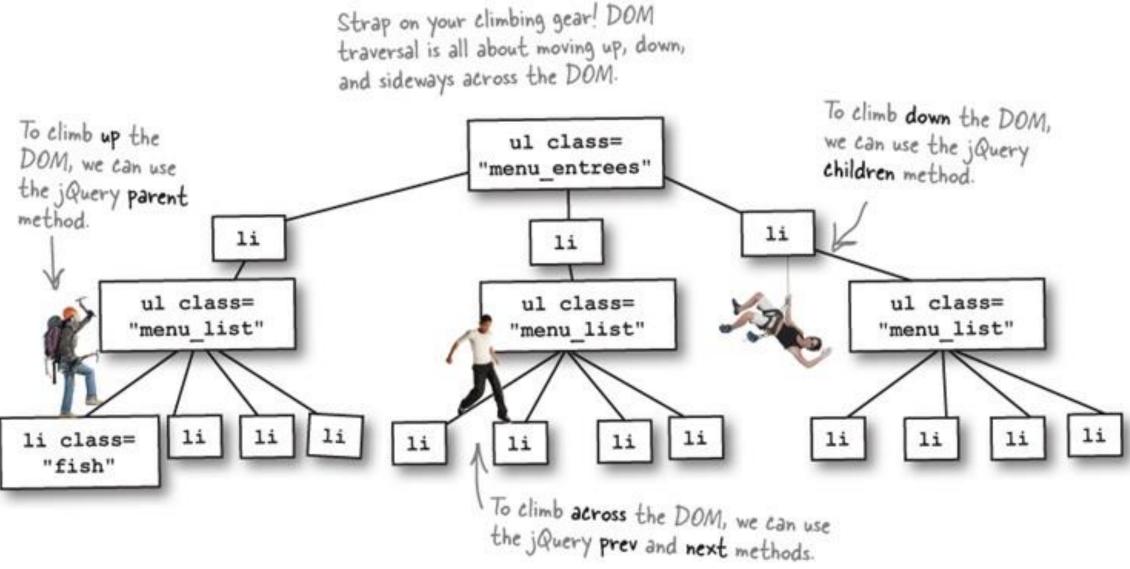
**DOM Traversals** 





prev( [selector] ) Get a set of elements containing the unique previous siblings of each of the matched set of elements next([selector]) Get a set of elements containing the unique next siblings of each of the given set of elements parent( [selector] ) Get the direct parent of an element If called on a set of elements, parent returns a set of their unique direct parent elements children([selector]) Get a set of elements containing all of the unique immediate children of each of the matched set of elements siblings([selector]) Get a set of elements containing all of the unique siblings of each of the matched set of elements





49



```
<div>
                  This is First paragraph.
$('#myDiv').prev()
              </div>
               <div id="myDiv">
                  >
                      This is second paragraph.
                  <div id="inrDiv">
                      This is third paragraph.
                                                     $('#myDiv').siblings()
$('#myDiv').children()
                  </div>
                  <div>
                      <l
$('#inrDiv').parent()
                          First
                          Second
                          Third
                      </div>
              <div>
                  This is fourth paragraph.
              </div>
                                                    $('#myDiv').next()
```

©2019 Clayfin

jq31

parents( [selector] ) Get a set of elements containing the unique ancestors
of the matched set of elements (except for the root element)
closest( selector ) Get a set of elements containing the closest parent
element that matches the specified selector, the starting element included

.parents()	.closest()
Begins with the parent element	Begins with the current element
Traverses up to complete DOM tree and found all selectors and returns all the set	Traverses up the DOM tree and if found the first selector stops there



find( selector ) Searches for descendant elements that match the specified selectors

element that matches the specified selector, the starting element included prevAll([selector]) Find all sibling elements in front of the current element nextAll([selector]]) Find all sibling elements after the current element





### jQuery .closest and .find are the best friends

.closest selector traverses up the DOM to find the parent that matches the conditions

.find selector traverses down the DOM that matches the conditions







**Events** 



# jQuery Event Helper Methods



click	keydown	
change	keypress	
focus	keyup	
blur	mousedown	
select	mouseenter	
submit	mousemove	
error	mouseleave	
load	mouseout	
unload	mouseover	
dblclick	mouseup	

bind( type, [data], fn ) Binds a handler to one or more events (like click) for each matched element Can also bind custom events unbind( [type], [fn] ) it removes bound events from each of the matched elements

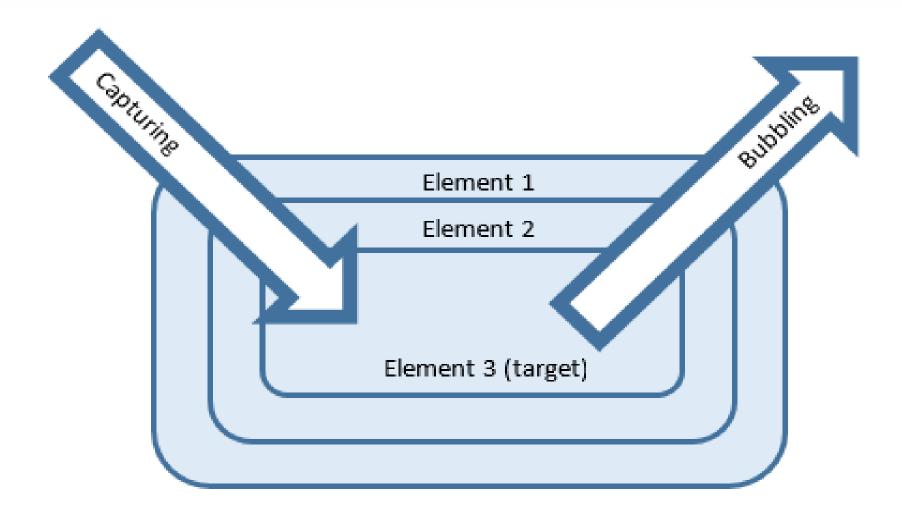
live and die - Attach and Detach a event handler for all elements which match the current selector, now and in the future

**delegate** and **undelegated** – Attach and Detach event handler for specified elements which are the children of selected element

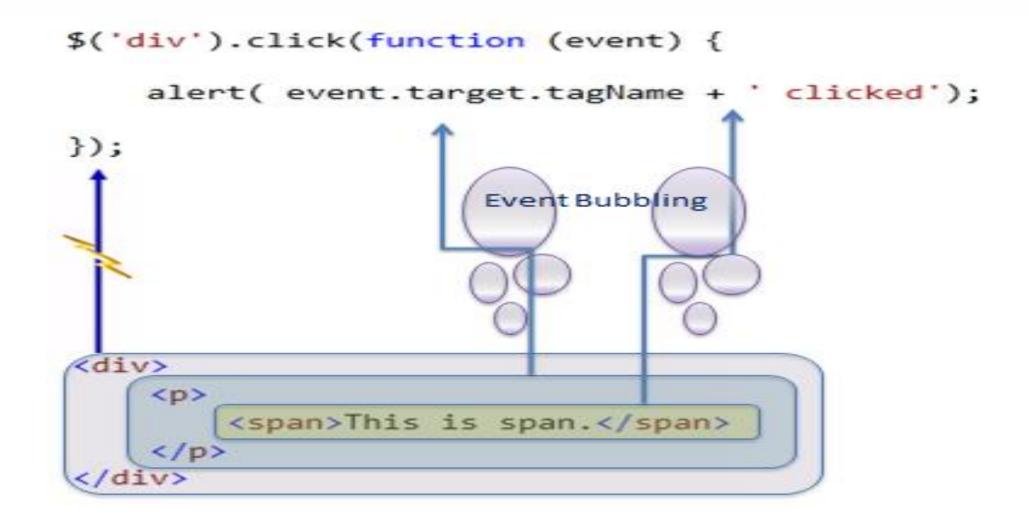
on( events [, selector ] [, data ], handler ) Binds a handler to an event (like click) for all current – and future – matched element off( events [, selector ] [, handler(eventObject) ] ) it removes a bound live event

one(type, [data], fn) Binds a handler to one or more events to be executed once for each matched element











preventDefault() Prevents the browser from executing the default action
stopPropagation() Stops the bubbling of an event to parent elements,
preventing any parent handlers from being notified of the event





trigger( event, [data] ) Trigger an event on every matched element triggerHandler( event, [data] ) Triggers all bound event handlers on an element

- The .triggerHandler("event") method will not call .event() on the element it is triggered on. This means .triggerHandler("submit") on a form will not call .submit() on the form.
- While .trigger() will operate on all elements matched by the jQuery object,
   .triggerHandler() only affects the first matched element.
- Events triggered with .triggerHandler() do not bubble up the DOM hierarchy; if they are not handled by the target element directly, they do nothing.
- Instead of returning the jQuery object (to allow chaining), .triggerHandler() returns
  whatever value was returned by the last handler it caused to be executed. If no
  handlers are triggered, it returns undefined





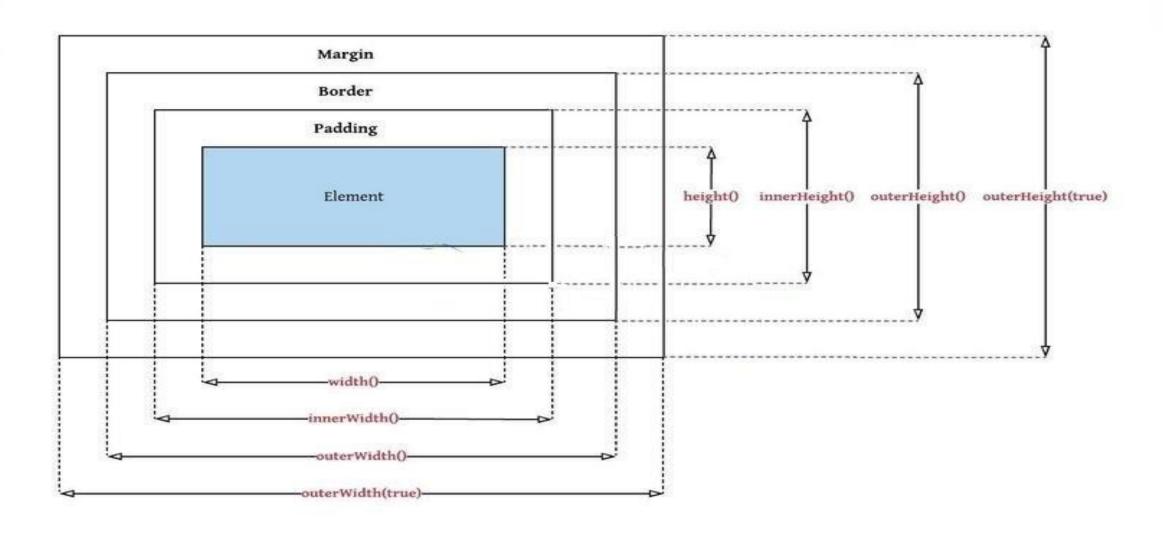


**Dimensions** 









jq38

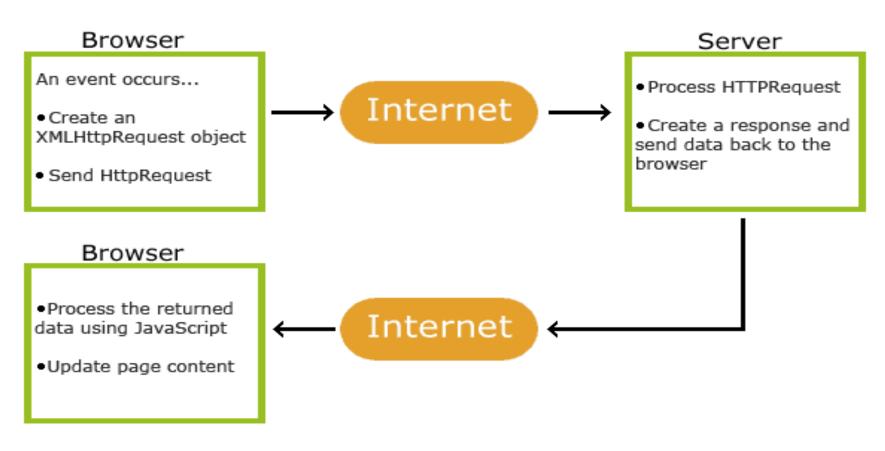


#### **Asynchronous JavaScript and XML**

Read data from server

Update a web page without reloading the page

Send data to server - in the background (asynchronous)





```
var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) { return this.responseText; }
    };
    xhttp.open("GET", "ajax_info.txt", true);
    xhttp.send();
```

onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest.
	0: request not initialized
	1: server connection established
	2: request received
	3: processing request
	4: request finished and response is ready
status	200: "OK"
	403: "Forbidden"
	404: "Page not found"



#### **Loading Simple Data**

[selector].load( URL, [data], [callback] )

```
$(document).ready(function() {
      $("# btn ").click(function(event){
        $("#mydiv").load('result.html');
      });
     });
<div id="mydiv">
    This is a DIV
</div>
<input type="button" value="Click" id="btn" />
```



#### **Getting JSON Data**

\$.getJSON( URL, [data], [callback] )

```
$(document).ready(function() {
             $("# btn ").click(function(event){
      $.getJSON('result.json', function(jd) {
                $('#mydiv ').html(' First Name: ' + jd.fname + '');
                $('#mydiv ').append('Last Name : ' + jd.lname+ '');
                $('#mydiv ').append(' Email: ' + jd.email+ '');
          });
      });
      <div id="mydiv">
           This is a DIV
       </div>
©2019 Clayfin <input type="button" value="Click" id="btn" />
```





ajaxStart( callback )	Attach a function to be executed whenever an AJAX request begins and there is none already active.
ajaxSend( callback )	Attach a function to be executed before an AJAX request is sent.
ajaxComplete( callback )	Attach a function to be executed whenever an AJAX request completes.
ajaxSuccess( callback )	Attach a function to be executed whenever an AJAX request completes successfully.
ajaxError( callback )	Attach a function to be executed whenever an AJAX request fails.
ajaxStop( callback )	Attach a function to be executed whenever all AJAX requests have ended.



#### **Getting data from the Server using GET**

```
$.get( url, [data], [callback], [type] )
```

```
$(document).ready(function() {
     $("# btn ").click(function(event){
       $.get( "/result", { name: "Karthik" }, function(data) {
           $('#mydiv ').html(data);
       );
     });
});
<div id="mydiv">
     This is a DIV
</div>
<input type="button" value="Click" id="btn" />
```



#### Posting data to the Server using POST

```
$.post( url, [data], [callback], [type] )
```

```
$(document).ready(function() {
    $("# btn ").click(function(event){
       $.post( "/result",{ name: "Karthik" }, function(data) {
           $('#mydiv ').html(data);
          });
       });
});
<div id="mydiv">
     This is a DIV
</div>
<input type="button" value="Click" id="btn" />
```





# Posting data to the Server using \$.ajax

\$.ajax(options)

_		
	url	A string containing the URL to which the request is sent.
	TVDE	A string defining the HTTP method to use for the request (GET or POST). The default value is GET.
	data	A map or string that is sent to the server with the request.
	success	A callback function that is executed if the request succeeds.
	error	A callback function that is executed if the request fails.
	timeout	Number of milliseconds after which the request will time out in failure.
	complete	A callback function that executes whenever the request finishes.
	async	A Boolean indicating whether to perform the request asynchronously. The default value is true.
	beforeSend	A callback function that is executed before the request is sent.
	CONTONTIVAD	A string containing a MIME content type to set for the request. The default value is application/x-www-form-urlencoded.
	dataType	A string defining the type of data expected back from the server (xml, html, json, or script).
	username	A username to be used in response to an HTTP access authentication request.
yfin	password	A password to be used in response to an HTTP access authentication request.
_		



#### Posting data to the Server using \$.ajax

```
$(document).ready(function() {
     $("# btn ").click(function(event){
        $.ajax( { url : '/result', success : function(data) {
           $('#mydiv').html(data);
          });
        });
});
<div id="mydiv">
     This is a DIV
</div>
<input type="button" value="Click" id="btn" />
```



#### **Global Settings for AJAX**

\$.ajaxSetup(options)

```
$(document).ready(function() {
    $("# btn ").click(function(event){
       $.ajaxSetup({ url : '/result' });
       $.ajax( {success:function(data) {
           $('#mydiv').html(data);
          });
        });
});
<div id="mydiv">
     This is a DIV
</div>
<input type="button" value="Click" id="btn" />
```

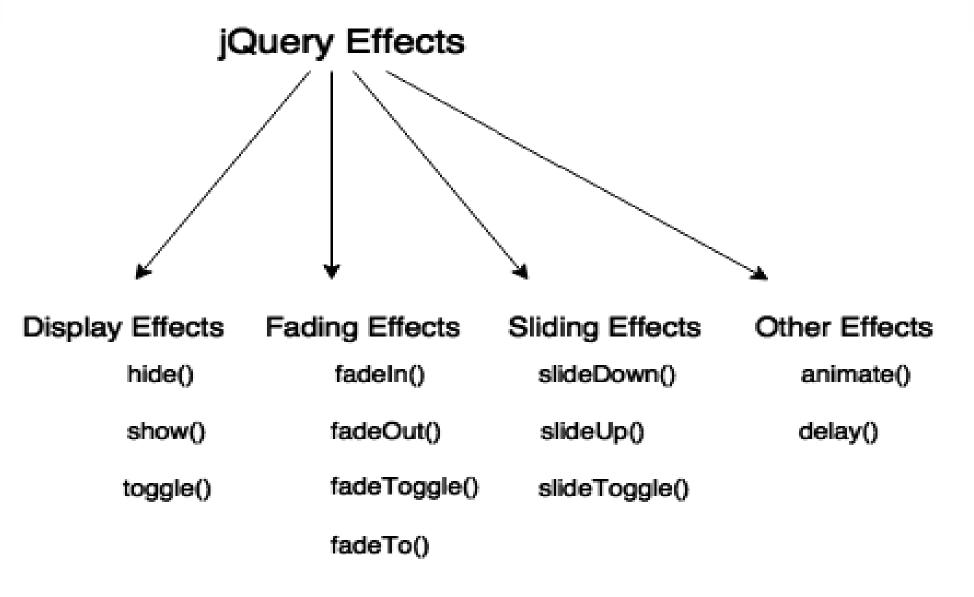




**Animations** 









	Displays each of the set of matched elements if they are
show()	hidden.
	Show all matched elements using a graceful animation and
<pre>show( speed, [callback] )</pre>	firing an optional callback after completion.
	Hides each of the set of matched elements if they are
hide( )	shown.
	Hide all matched elements using a graceful animation and
hide(speed, [callback])	firing an optional callback after completion.
toggle()	Toggle displaying each of the set of matched elements.
	Toggle displaying each of the set of matched elements using
	a graceful animation and firing an optional callback after
toggle(speed, [callback])	completion.
	Toggle displaying each of the set of matched elements based
	upon the switch (true shows all elements, false hides all
toggle( switch )	elements).





	Fade in all matched elements by adjusting their opacity
fadeIn( speed, [callback] )	and firing an optional callback after completion.
	Fade out all matched elements by adjusting their
	opacity to 0, then setting display to "none" and firing
fadeOut( speed, [callback] )	an optional callback after completion.
	Fade the opacity of all matched elements to a
	specified opacity and firing an optional callback after
fadeTo( speed, opacity, callback )	completion.
	Hide all matched elements by adjusting their height
	and firing an optional callback after completion.
	Reveal all matched elements by adjusting their height
	and firing an optional callback after completion.
	Toggle the visibility of all matched elements by
	adjusting their height and firing an optional callback
	after completion.







**Plug-in APIs** 



Clayfin

A plug-in is piece of code written in a standard JavaScript file. These files provide useful jQuery methods which can be used along with jQuery library methods.

jQuery Plug-ins

- By extending the Jquery's prototype object you enable all jQuery objects to inherit any methods that you add. whenever you call jQuery() you're creating a new jQuery object, with all of jQuery's methods inherited
- There are plenty of jQuery plug-in available which you can download from repository link at <a href="https://jquery.com/plugins">https://jquery.com/plugins</a>



## How to develop a Plug-in

#### jQuery.fn.methodName = methodDefinition;

- methodName is the name of new method and methodDefinition is actual method definition
- Any methods or functions you attach must have a semicolon (;) at the end
- Your method must return the jQuery object, unless explicity noted otherwise
- You should use this.each to iterate over the current set of matched elements
- Prefix the filename with jquery, follow that with the name of the plugin and conclude with .js
- For example, if we write a plugin that we want to name custom, our JavaScript filename for this plugin should be jquery.custom.js
- The use of the **jquery**. prefix eliminates any possible name collisions with files intended for use with other libraries
- Always attach the plugin to jQuery directly instead of \$, so users can use a custom alias via noConflict() method



## warning method for debugging purposes

#### jquery.warning.js

```
jQuery.fn.warning = function() {
 return this.each(function() {
   alert('Tag Name:"' + $(this).prop("tagName") + "".');
 });
};
$("div").warning();
$("p").warning();
```







**Best Practices** 





- Use \$( document ).ready() Avoid \$( window ).load() unless it is really necessary
- Use Method Chaining wherever possible
- Cache wherever possible
- Don't Abuse \$(this)
   \$('#someAnchor').click(function() {
   alert( \$(this).attr('id') ); // Use Vanilla JavaScript alert( this.id );
   });
- Be specific on the right-hand side of your selector, and less specific on the left
- Avoid Reflow ( refers to layout changes in a webpage ) . Use detach() instead of remove() to avoid expensive Reflows
- adding multiple rows to a table one by one it'll cause a lot of reflows. So it's better
  to take the table out the DOM tree, add the rows to it and put it back to the DOM
- Use Promises wherever possible for Nested AJAX Calls
- Lazy load scripts when needed \$.getScript("js/custom.js", callback);



```
$('#elem').hide();
$('#elem').html('bla');
$('#elem').otherStuff();
//Method Chaining
$('#elem')
 .hide()
 .html('bla')
 .otherStuff();
//Caching
var elem = $('#elem');
elem.hide();
elem.html('bla');
elem.otherStuff();
```



```
// This is bad
$.each(myArray, function(i, item) {
        var newListItem = '' + item + '';
        $('#myList').append(newListItem);
});
// Better Way
var myHtml = ";
$.each(myArray, function(i, item) {
        myHtml += '' + item + '';
});
$('#myList').html(myHtml);
```



# **Thank You**

www.clayfin.com

