JavaScript and jQuery Course

Session 12

Topics Covered

- Selectors / Filters
- Methods
- Event Methods
- Dom Manipulation Methods – not in text (extra)

Some of the most useful jQuery Selectors

```
[attribute]
[attribute=value]
:eq(n)
:even
:first
:first-child
:gt(n)
:header
:hidden
:last
:last-child
:lt(n)
:not(selector)
:nth-child(n)
:only—child
:parent
:odd
```

also known as filters

Selectors / Filters

Same as CSS

`:first-child`	`$("p:first-child")`	All `<p>` elements that are the first child of their parent
`:last-child`	`$("p:last-child")`	All `<p>` elements that are the last child element
`:nth-child(n)`	`$("p:nth-child(2)")`	All `<p>` elements that are the 2nd child element
`:first-of-type`	`$("li:first-of-type")`	All `<li>` elements that are the first `<li>` element of their parent
`:last-of-type`	`$("li:last-of-type")`	All <li> elements that are the last `<li>` element
`:nth-of-type(n)`	`$("li:nth-of-type(2)")`	All `<li>` elements that are the 2nd `<li>` element
`:only-child`	`$("p:only-child")`	All `<p>` elements that are the only child of their parent
`:only-of-type`	`$("p:only-of-type")`	All `<p>` elements that are the only child of its type

Selectors / Filters

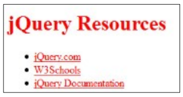
Like CSS Shortcut

`:eq(index)`	`$("ul li:eq(3)")`	The fourth element in the list (index starts at 0)
`:gt(number)`	`$("ul li:gt(3)")`	List elements with an index greater than 3
`:lt(number)`	`$("ul li:lt(3)")`	List elements with an index less than 3
`:not(selector)`	`$("input:not(selector)")`	All input elements that are not that selector
`:header`	`$(":header")`	All header elements (h1, h2,…)
`:animated`	`$(":animated")`	All animated elements
`:contains('string')`	`$(":contains('string')")`	Elements that contain the string
`:has('element')`	`$(":has('element')")`	Elements that have the element
`:empty`	`$(":empty")`	Elements with no child nodes
`:hidden`	`$("p:hidden")`	All hidden `<p>` elements
`:visible`	`$("table:visible")`	All visible tables

Form related Filters

`:input`	`$(":input")`	All input elements
`:text`	`$(":text")`	All input elements with `type="text"`
`:password`	`$(":password")`	`type="password"`
`:radio`	`$(":radio")`	`type="radio"`
`:checkbox`	`$(":checkbox")`	`type="checkbox"`
`:submit`	`$(":submit")`	`type="submit"`
`:reset`	`$(":reset")`	`type="reset"`
`:button`	`$(":button")`	`type="button"`
`:image`	`$(":image")`	`type="image"`
`:file`	`$(":file")`	`type="file"`
`:enabled`	`$(":enabled")`	All enabled input elements
`:disabled`	`$(":disabled")`	All disabled input elements
`:selected`	`$(":selected")`	All selected input elements
`:checked`	`$(":checked")`	All checked input elements

Attribute Selectors

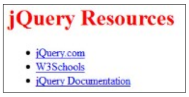


jQuery Resources

- jQuery.com
- W3Schools
- jQuery Documentation

```
$('[title]').css("color", "red");

<h1 title="resources">jQuery Resources</h1>

<ul>
  <li><a href="http://www.jquery.com"
        title="jQuery website">jQuery.com</a></li>

  <li><a
       href="http://www.w3schools.com/jquery/jquery_ref_selectors.asp"
       title="W3Schools jQuery reference">W3Schools</a></li>

  <li><a href="http://docs.jquery.com"
        title="jQuery documentation">jQuery Documentation</a></li>
</ul>
```

Attribute Selectors of a specific element

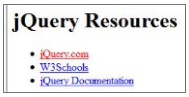


jQuery Resources

- jQuery.com
- W3Schools
- jQuery Documentation

```
$('h1[title]').css("color", "red");

<h1 title="resources">jQuery Resources</h1>

<ul>
  <li><a href="http://www.jquery.com"
        title="jQuery website">jQuery.com</a></li>

  <li><a
       href="http://www.w3schools.com/jquery/jquery_ref_selectors.asp"
       title="W3Schools jQuery reference">W3Schools</a></li>

  <li><a href="http://docs.jquery.com"
        title="jQuery documentation">jQuery Documentation</a></li>
</ul>
```

Attribute that has an exact value


jQuery Resources

- jQuery.com
- W3Schools
- jQuery Documentation

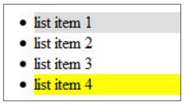
quotation marks

```
$('[title="jQuery website"]').css("color", "red");

<h1 title="resources">jQuery Resources</h1>

<ul>
  <li><a href="http://www.jquery.com"
        title="jQuery website">jQuery.com</a></li>

  <li><a
       href="http://www.w3schools.com/jquery/jquery_ref_selectors.asp"
       title="W3Schools jQuery reference">W3Schools</a></li>

  <li><a href="http://docs.jquery.com"
        title="jQuery documentation">jQuery Documentation</a></li>
</ul>
```

Like CSS Shortcut

Selector / Filter – :first :last

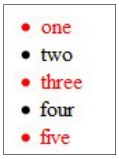
```
$(document).ready(function() {

    $('li:first').css('background','#dedede');
    $('li:last' ).css('background','#ff0');

}); //end ready

<ul>
   <li>list item 1</li>
   <li>list item 2</li>
   <li>list item 3</li>
   <li>list item 4</li>
</ul>
```


- list item 1
- list item 2
- list item 3
- list item 4

Like CSS Shortcut

Selector / Filter – :even

```
$(document).ready(function() {

    $("li:even").css("color", "red");

}); //end ready

<ul>
   <li>one</li>
   <li>two</li>
   <li>three</li>
   <li>four</li>
   <li>five</li>
</ul>
```

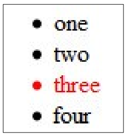

- one
- two
- three
- four
- five

starts counting at 0

Like CSS

Selector / Filter – :nth-child(n)

```
$(document).ready(function() {

    $("li:nth_child(3)").css("color", "red");

}); //end ready

<ul>
   <li>one</li>
   <li>two</li>
   <li>three</li>
   <li>four</li>
</ul>
```

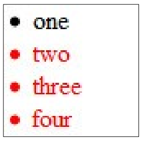

- one
- two
- three
- four

starts counting at 1

Like CSS

Selector / Filter – :not(some_selector)

```
$(document).ready(function() {

    $("li:not(:first)").css("color", "red");

}); //end ready

<ul>
   <li>one</li>
   <li>two</li>
   <li>three</li>
   <li>four</li>
</ul>
```



- one
- two
- three
- four

jQuery Methods

Methods can

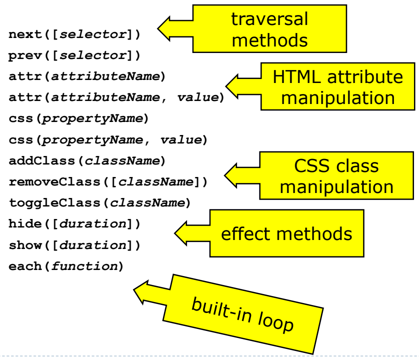
set (write / change)

and

get (read / return)

properties/values

Useful jQuery methods



```
next([selector])
prev([selector])
attr(attributeName)
attr(attributeName, value)
css(propertyName)
css(propertyName, value)
addClass(className)
removeClass([className])
toggleClass(className)
hide([duration])
show([duration])
each(function)
```

traversal methods

HTML attribute manipulation

CSS class manipulation

effect methods

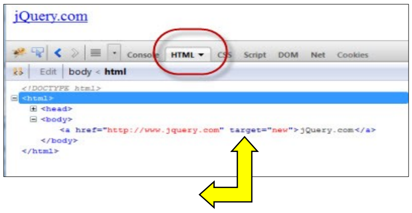
built-in loop

Like setAttribute()

attr(name, value) Method

Setting an attribute and value

`$("a").attr("target" , "new");`



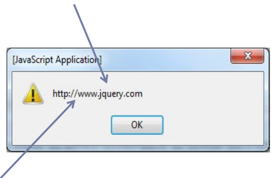
```
<!DOCTYPE html>
<html>
  <head>
  <body>
      <a href="http://www.jquery.com" target="new">jQuery.com</a>
  </body>
</html>
```

`<a href="http://www.jquery.com">jQuery.com</a>`

attr(name) Method

Getting a value for an HTML attribute

```
var theHref = $("a").attr("href");
alert(theHref);
```



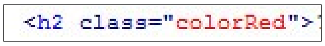
http://www.jquery.com

```
<a href="http://www.jquery.com">jQuery.com</a>
```

Like className property

addClass(class_name) Method

Adding a CSS class

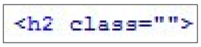


<h2 class="colorRed">

```
<style>
  .colorRed {color: red}
</style>

$('h2').addClass('colorRed');

<h2></h2>
```

removeClass(class_name) Method

Removing a CSS class

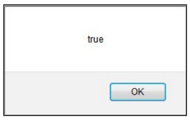


<h2 class="">

```
<style>
  .colorRed {color: red}
</style>

$('h2').removeClass('colorRed');

<h2></h2>
```

hasClass(class_name) Method

Testing to see if the class exists



true

```
<style>
  .colorRed {color: red}
</style>

alert( $('h2').hasClass('colorRed') );

<h2 class="colorRed"></h2>
```

toggleClass(class_name) Method

Add and remove a class

```
$('#btn').click(function() {
  $('h2').toggleClass('colorRed');
});  //end click
```

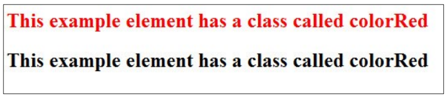
Effects all <h2> elements

toggleClass(class_name) Method

Adds and removes the class

```
$('h2').click(function() {
    $(this).toggleClass('colorRed');
});  //end click
```

Effects current <h2> element



This example element has a class called colorRed

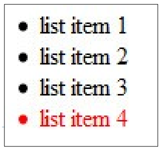
This example element has a class called colorRed

Like nextElementSibling

next() Traversal Method

Get the next sibling

```
$('li.third').next().addClass('colorRed');

<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third">list item 3</li>
  <li>list item 4</li>
</ul>
```

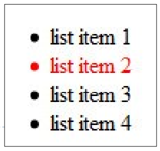


- list item 1
- list item 2
- list item 3
- list item 4

prev() Traversal Method

Get the sibling immediately before

```
$('li.third').prev().addClass('colorRed');

<ul>
  <li>list item 1</li>
  <li>list item 2</li>
  <li class="third">list item 3</li>
  <li>list item 4</li>
</ul>
```



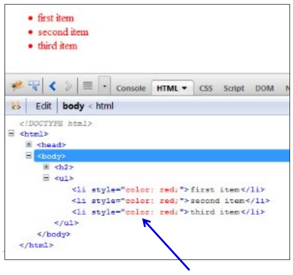
- list item 1
- list item 2
- list item 3
- list item 4

each(function) Method – takes a function

```
$("li").each(function() {

  $(this).css("color" , "red");
```

Refers to each instance of the selector

```
});   // end each
```



- first item
- second item
- third item

```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h2>
    <ul>
        <li style="color: red;">first item</li>
        <li style="color: red;">second item</li>
        <li style="color: red;">third item</li>
      </ul>
    </body>
</html>
```

hide() Method

Hides the selector (`display: none`)

```
$('#btn1').click(function() {

    $('#para1').hide();

}); // end click


<input type="button"
       id="btn1"
       value="Click Me">
```

show() Method

Shows the selector (`display: block`)

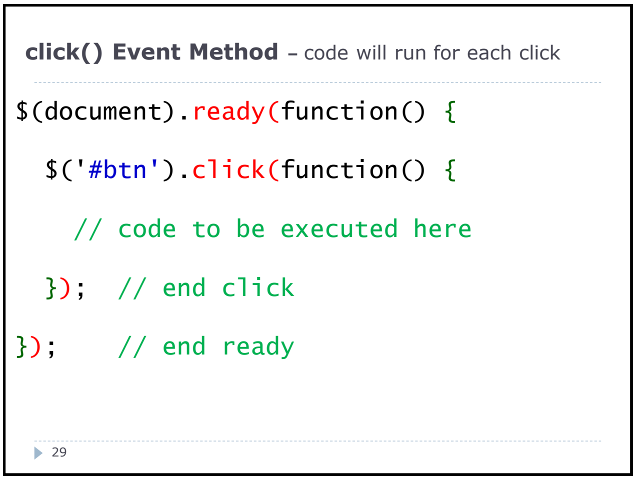
```
$('#btn2').click(function() {

    $('#para1').show();

}); // end click


<input type="button"
       id="btn2"
       value="Click Me">
```

Useful jQuery event methods

```
ready(handler)
click(handler)
dblclick(handler)
mouseenter(handler)
mouseleave (handler)
mouseover(handler)
mouseout(handler)
hover(handlerIn, handlerOut)
toggle(handlerIn, handlerOut)
event.preventDefault()
```

Compound event / 2 functions

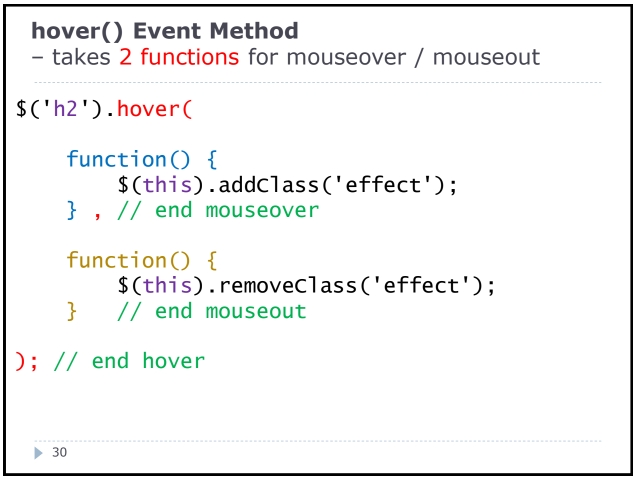
Compound event / 2 functions
Deprecated version 1.7



click() Event Method – code will run for each click

```
$(document).ready(function() {

  $('#btn').click(function() {

    // code to be executed here

  });  // end click

});    // end ready
```

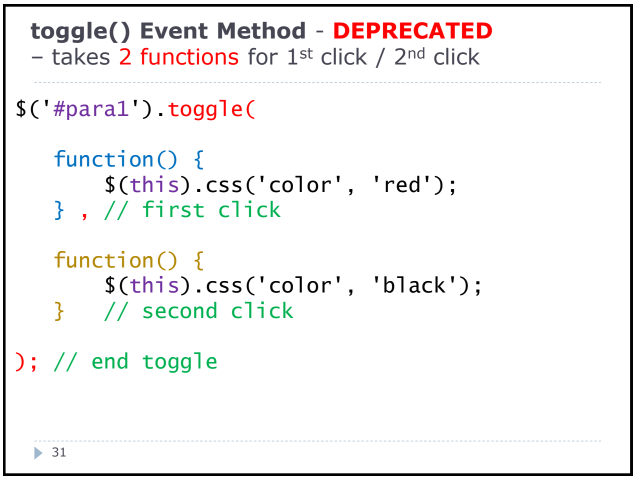
29



hover() Event Method
– takes 2 functions for mouseover / mouseout

```
$('h2').hover(

    function() {
        $(this).addClass('effect');
    } , // end mouseover

    function() {
        $(this).removeClass('effect');
    }   // end mouseout

); // end hover
```

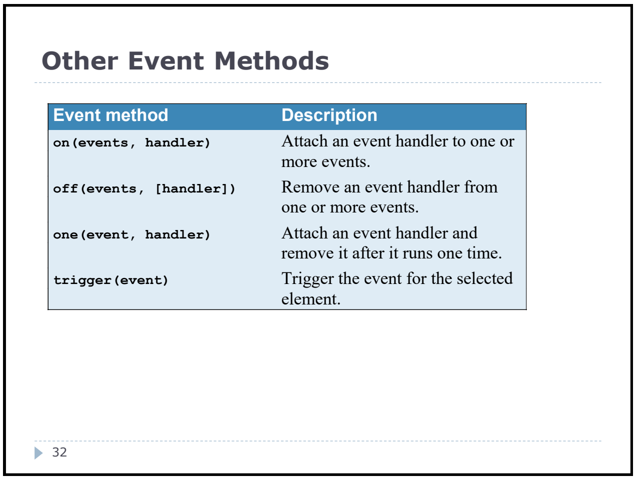
30



toggle() Event Method - DEPRECATED
– takes 2 functions for 1st click / 2nd click

```
$('#para1').toggle(

    function() {
        $(this).css('color', 'red');
    } , // first click

    function() {
        $(this).css('color', 'black');
    }   // second click

); // end toggle
```

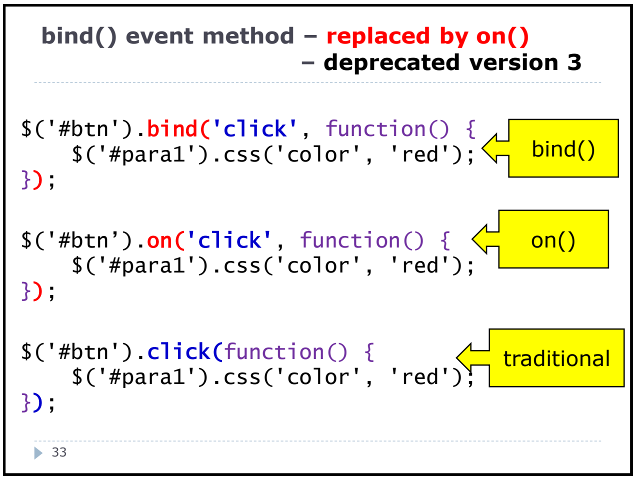
31



Other Event Methods

Event method	Description
`on(events, handler)`	Attach an event handler to one or more events.
`off(events, [handler])`	Remove an event handler from one or more events.
`one(event, handler)`	Attach an event handler and remove it after it runs one time.
`trigger(event)`	Trigger the event for the selected element.

32



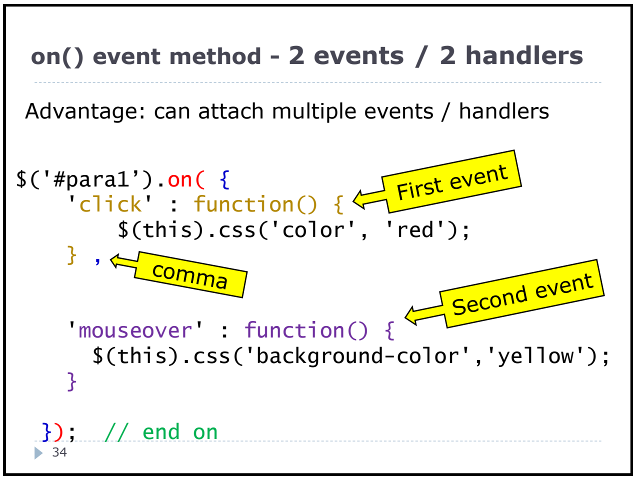
bind() event method – replaced by on() – deprecated version 3

```
$('#btn').bind('click', function() {
    $('#para1').css('color', 'red');
});

$('#btn').on('click', function() {
    $('#para1').css('color', 'red');
});

$('#btn').click(function() {
    $('#para1').css('color', 'red');
});
```

bind()

on()

traditional



on() event method - 2 events / 2 handlers

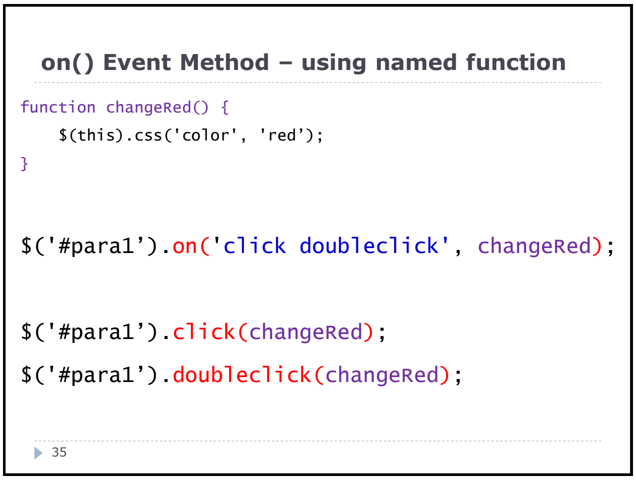
Advantage: can attach multiple events / handlers

```
$('#para1').on( {
    'click' : function() {
        $(this).css('color', 'red');
    } ,

    'mouseover' : function() {
      $(this).css('background-color','yellow');
    }

  });  // end on
```

First event

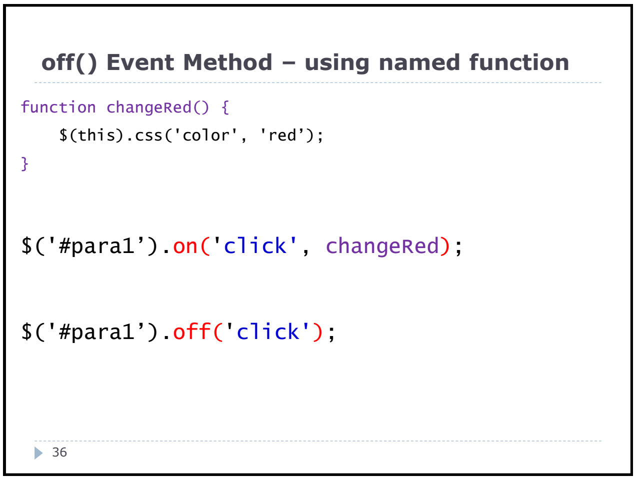
comma

Second event



on() Event Method – using named function

```
function changeRed() {
    $(this).css('color', 'red');
}


$('#para1').on('click doubleclick', changeRed);


$('#para1').click(changeRed);

$('#para1').doubleclick(changeRed);
```



off() Event Method – using named function

```
function changeRed() {
    $(this).css('color', 'red');
}


$('#para1').on('click', changeRed);


$('#para1').off('click');
```

one() Event Method – code will run once

```
$(document).ready(function() {

  $('#btn').one('click', function() {

    // code to be executed here

  });  // end one

});    // end ready
```

DOM Manipulation Methods – not in text

Write to the page

append() method
This method will add the new content to the **end** of the existing content.

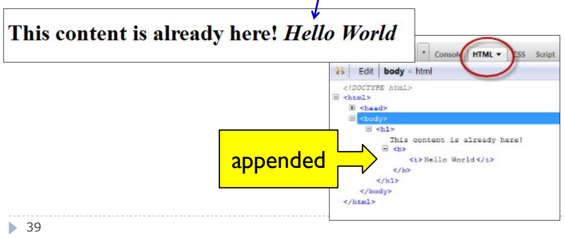
prepend() method
This method will add the new content to the **beginning** of the existing content.

Note: html() replaces existing content!

append() Method – at the end

$("h1").append(" <b><i>Hello World</i></b>");

<h1>This content is already here!</h1>



This content is already here! *Hello World*

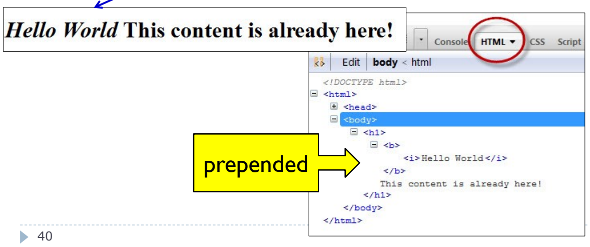
```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h1>
      This content is already here!
      <b>
        <i>Hello World</i>
      </b>
    </h1>
  </body>
</html>
```

appended

prepend() Method – at the beginning

$("h1").prepend("<b><i>Hello World</i></b> ");

<h1>This content is already here!</h1>



***Hello World* This content is already here!**

```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h1>
      <b>
        <i>Hello World</i>
      </b>
      This content is already here!
    </h1>
  </body>
</html>
```

prepended

DOM Manipulation Methods – not in text

appendTo() method

This method takes a selector as a parameter and uses that selector to place the content.

prependTo() method

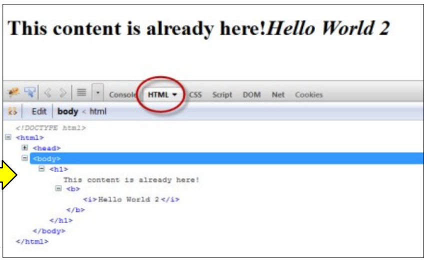
This method takes a selector as a parameter and uses that selector to place the content.

appendTo() Method

Requires a selector

```
$("<b><i>Hello World 2</i></b>").appendTo("h1");
```

```
<h1>This content is already here!</h1>
```



This content is already here!*Hello World 2*

```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h1>
      This content is already here!
      <b>
        <i>Hello World 2</i>
      </b>
    </h1>
  </body>
</html>
```

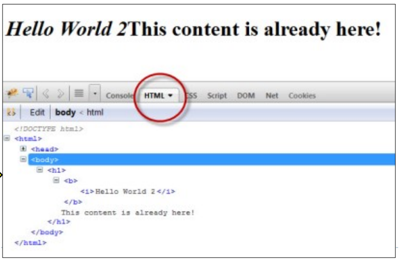
appended to <h1>

prependTo() Method

Requires a selector

```
$("<b><i>Hello World 2</i></b>").prependTo("h1");
```

```
<h1>This content is already here!</h1>
```



***Hello World 2*This content is already here!**

```
<!DOCTYPE html>
<html>
  <head>
  <body>
    <h1>
      <b>
        <i>Hello World 2</i>
      </b>
      This content is already here!
    </h1>
  </body>
</html>
```

prepended to <h1>

DOM Manipulation Methods – not in text

before() method

This method will add the new content before the selection

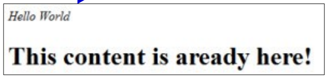
after() method

This method will add the new content after the selection

before() Method – before the selector

```
$("h1").before("<p><i>Hello World</i></p>");
```

```
<h1>This content is already here!</h1>
```



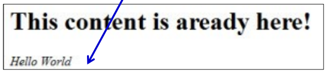
Hello World

This content is aready here!

after() Method – before the selector

```
$("h1").after("<p><i>Hello World</i></p>");
```

```
<h1>This content is already here!</h1>
```



This content is aready here!

Hello World

DOM Manipulation Methods – not in text

insertBefore() method

This method takes a selector as a parameter and uses that selector to place the content.

insertAfter() method

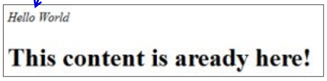
This method takes a selector as a parameter and uses that selector to place the content.

insertBefore() Method

Requires a selector

```
$("<p><i>Hello World</i></p>").insertBefore("h1");
```

```
<h1>This content is already here!</h1>
```



Hello World

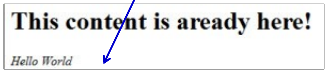
This content is aready here!

insertAfter() Method

Requires a selector

```
$("<p><i>Hello World</i></p>").insertAfter("h1");
```

```
<h1>This content is already here!</h1>
```



This content is aready here!

Hello World

DOM Manipulation Methods – not in text

remove() method – similar to delete

This method will remove the selected element.

detach() method – similar to cut / paste

This method will remove but keep a copy of the selected element. You must then do something with it.

clone() method – similar to copy / paste

This method will clone or copy the selection. You must then do something with it.