

## USING BASIC JQUERY SELECTORS

---

- CSS-style selectors and filters are based on familiar CSS syntax, and work pretty much the same way as CSS does
- The CSS selectors listed here correspond directly to their CSS counterparts

SELECTOR	PURPOSE
tagname	Finds all elements that are named tagname
#identifier	Finds all elements with ID of identifier
.className	Finds all elements that have class attribute with the value of className
tag.className	Gets elements of type tag that have a class attribute with the value of className
tag#id.className	Retrieves the tag element that has an ID of id and a class attribute with the value of className
*	Finds all of the elements on the page

## USING BASIC JQUERY SELECTORS

---

- Using jQuery vs. using the plain browser DOM

```
<html>
<head>
<title>Document</title>
</head>
<body>
<ul id="list1">
    <li class="a">item 1</li>
    <li class="a">item 2</li>
    <li class="b">item 3</li>
    <li class="b">item 3</li>
</ul>
<p>This is paragraph 1</p>
<p>This is paragraph 2</p>
<p>This is paragraph 3</p>
<p>This is paragraph 4</p>
</body>
</html>
```

Get all <p> tags (DOM):  
document.getElementsByTagName ("p");

Get all <p> tags (jQuery):  
\$("p");

Get the tag with id "list1" (DOM):  
document.getElementById("list1");

Get the tag with id "list1" (jQuery):  
\$("#list1");

Get all <li> tags with class "a" (jQuery):  
\$("li.a");

Get all tags with class "b", but only if they are inside a <ul> (jQuery):  
\$("ul .b");

## USING BASIC JQUERY SELECTORS

---

- The hierarchy and combination selectors allow you to get a little more advanced in selecting page content
- You can select elements based on hierarchical relationships or on a series of common criteria

SELECTOR	PURPOSE
<code>selector, selector, ...</code>	Finds all of the specified selectors
<code>.class1.class2</code>	Finds all elements with both <code>.class1</code> and <code>.class2</code> applied
<code>parent &gt; child</code>	Finds all <code>child</code> elements that are direct children of elements of type <code>parent</code>
<code>ancestor descendant</code>	Finds all <code>descendant</code> elements that are contained within elements of type <code>ancestor</code>
<code>prev + next</code>	Finds all <code>next</code> elements that are next to a <code>prev</code> element
<code>prev ~ siblings</code>	Finds all <code>sibling</code> elements that come after <code>prev</code> and match the <code>siblings</code> selector

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Document</title>
<script type="text/javascript" src="../jquery-1.3.2.js"></script>
<script type="text/javascript">
    $("document").ready(function() {
    });
</script>
<style type="text/css">
.a { color: Navy; }
.b { color: Maroon; }
</style>
</head>
<body>
    <ul id="list1">
        <li class="a">item 1</li>
        <li class="a">item 2</li>
        <li class="b">item 3</li>
        <li class="b">item 4</li>
    </ul>
<p class="a">This is paragraph 1</p>
<p>This is paragraph 2</p>
<p class="b">This is paragraph 3</p>
<p>This is paragraph 4</p>
</body>
</html>

```

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

```

]<script type="text/javascript">
    $("document").ready(function() {
        $("p").css("border", "3px solid red");
    });
-</script>

<script type="text/javascript">
    $("document").ready(function() {
        $(".a").css("border", "3px solid red");
    });
</script>

```

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```
<script type="text/javascript">
    $("document").ready(function() {
        $("#list1").css("border", "3px solid red");
    });
</script>
```

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```
<script type="text/javascript">
    $("document").ready(function() {
        $("p.b").css("border", "3px solid red");
    });
</script>
```

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```
<script type="text/javascript">
    $("document").ready(function() {
        $("p, li.b").css("border", "3px solid red");
    });
</script>
```

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```
<script type="text/javascript">
    $("document").ready(function() {
        $("ul li.a").css("border", "3px solid red");
    });
</script>
```

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```


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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4


```

`<script type="text/javascript">
 $("document").ready(function() {
 $("ul + p").css("border", "3px solid red");
 });
</script>`

```


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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4


```

`<script type="text/javascript">
 $("document").ready(function() {
 $("#list1 ~ p").css("border", "3px solid red");
 });
</script>`

## JQUERY SELECTORS AND FILTERS: USING FILTERS

Range: Bogon:Ads, Spyware  
etc

- Filters work with selectors to provide even more fine-grained control over how elements are selected in the document
- jQuery filters fall into six different categories

<b>Basic</b>	<b>Content</b>	<b>Visibility</b>
Provides basic filtering, like getting the first, last, and even- and odd-numbered items in a returned set	Filters a set of elements based on the content, like whether an element contains a particular string	Filters a set of elements using the visibility setting of each element as a test
<b>Attribute</b>	<b>Child</b>	<b>Form</b>
Examines a given attribute on an element to determine whether it should be filtered out	Selects elements based upon their relationship with their parent element	Provides specialized filters that operate on form elements

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

## USING BASIC JQUERY FILTERS

---

- You can refine a selector by including elements that match certain conditions, like position or index

FILTER	PURPOSE
:first	Selects only the first instance of the selector's returned set
:last	Selects only the last instance of the selector's returned set
:even	Selects only even-numbered elements in the selector's returned set
:odd	Selects only odd-numbered elements in the selector's returned set
:eq(n)	Filters out elements that are not positioned at the given index
:gt(n)	Includes elements that are past the given index
:lt(n)	Includes elements that are before the given index
:header	Selects all header elements (H1, H2, H3, etc)
:animated	Selects all elements that are currently being animated in some way
:not(selector)	Includes elements that do not match the given selector

```
<script type="text/javascript">
    $("document").ready(function() {
        $("p:first").css("border", "3px solid red");
    });
</script>
```

This is paragraph 1  
This is paragraph 2  
This is paragraph 3  
This is paragraph 4

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

```
|<script type="text/javascript">
|    $("document").ready(function() {
|        //$("p:first").css("border", "3px solid red");
|        $("p:last").css("border", "3px solid red");
|    );
|</script>
```

This is paragraph 1  
This is paragraph 2  
This is paragraph 3  
This is paragraph 4

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

```


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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4



---



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



This is paragraph 1



This is paragraph 2



This is paragraph 3



This is paragraph 4


```

```

        • item 1
        • item 2
        • item 3
        • item 4

This is paragraph 1
This is paragraph 2
This is paragraph 3
This is paragraph 4

$("p:gt(1)").css("border", "3px solid red");

```

• item 1  
• item 2  
• item 3  
• item 4

```

This is paragraph 1
This is paragraph 2
This is paragraph 3
This is paragraph 4

$("p:not(p:eq(2))").css("border", "3px solid red");

```

## JQUERY SELECTORS AND FILTERS: ATTRIBUTE FILTERS

---

- You can filter the results of a selector statement based on attribute content

FILTER	PURPOSE
[attribute]	Includes elements in the result set if they have the specified <i>attribute</i>
[attribute=value]	Includes elements in the result set if they have the specified <i>attribute</i> and it has the given <i>value</i>
[attribute!=value]	Includes elements in the result set only if they have the specified <i>attribute</i> and it doesn't have the given <i>value</i>
[attribute^=value]	Includes elements that have the specified <i>attribute</i> and it starts with the specified <i>value</i>
[attribute\$=value]	Includes elements that have the specified <i>attribute</i> and it ends with the specified <i>value</i>
[attribute*=value]	Includes elements that have the specified <i>attribute</i> and it contains the specified <i>value</i>
[attrFilter1][attrFilterN]	Includes elements that match all of the specified attribute filters

```
<body>
  <ul id="list1">
    <li class="a">item 1</li>
    <li class="a">item 2</li>
    <li class="b">item 3</li>
    <li class="b">item 4</li>
  </ul>
  <p class="a">This is paragraph 1</p>
  <p id="para1">This is paragraph 2</p>
  <p class="b">This is paragraph 3</p>
  <p id="para4" lang="en-us">This is paragraph 4</p>
</body>
```

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```

        • item 1
        • item 2
        • item 3
        • item 4

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

        • item 1
        • item 2
        • item 3
        • item 4

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

```

## JQUERY SELECTORS AND FILTERS: CONTENT AND VISIBILITY FILTERS

---

- You can examine the content of selected elements or their visibility property to determine whether they should be included or excluded from the final set

CONTENT FILTER	PURPOSE
:contains(text)	Filters the selection to only include elements that contain the text string
:empty	Filters the selection to only include empty elements
:has(selector)	Matches elements that contain at least one element that has the specified selector
:parent	Matches all elements that are parents (i.e. they contain at least one other element, including text)
VISIBILITY FILTER	PURPOSE
:visible	Filters the selection to only include visible elements
:hidden	Filters the selection to only include hidden elements

## JQUERY SELECTORS AND FILTERS: CHILD FILTERS

---

- You can refine a selector by examining the relationship each element has with its parent element

FILTER	PURPOSE
:nth-child( <i>index</i> ) :nth-child(even) :nth-child(odd) :nth-child( <i>equation</i> )	Matches elements at <i>index</i> , or even or odd increments, or who match an equation of the form $Xn+M$ (for example, $2n$ or $3n+1$ )
:first-child	Matches elements who are the first child of their parent
:last-child	Matches elements who are the last child of their parent
:only-child	Matches elements who are the only child of their parent

```
12<style type="text/css">
13  .a { color: Navy; }
14  .b { color: Maroon; }
15</style>
16</head>
17<body>
18  <ul id="list1">
19    <li class="a">item 1</li>
20    <li class="a">item 2</li>
21    <li class="b">item 3</li>
22    <li class="b">item 3</li>
23  </ul>
24  <p class="a">This is paragraph 1</p>
25  <p id="para1">This is paragraph 2</p>
26  <p class="b">This is paragraph 3</p>
27  <p>This is paragraph 4</p>
```

- item 1
- item 2
- item 3
- item 3

This is paragraph 1

This is paragraph 2

This is paragraph 3

```
$(“p:contains(3)”).css(“border”, “3px solid red”); This is paragraph 4
```

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

```
$(“p:parent”).css(“border”, “3px solid red”); This is paragraph 4
```

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

```
$(“ul:has(li[class=a])”).css(“border”, “3px solid red”); This is paragraph 4
```

- item 1

- item 2

- item 3

- item 4 

This is paragraph 1

This is paragraph 2

This is paragraph 3

`$(“ul li:nth-child(3)”).css(“border”, “3px solid red”);` This is paragraph 4

- item 1

- item 2

- item 3

- item 4

This is paragraph 1

This is paragraph 2

This is paragraph 3

`$(“ul li:last-child”).css(“border”, “3px solid red”);` This is paragraph 4

- item 1

- item 2

- item 3

- item 4 

This is paragraph 1

This is paragraph 2

This is paragraph 3

`$(“ul li:nth-child(2n)”).css(“border”, “3px solid red”);` This is paragraph 4

## JQUERY SELECTORS AND FILTERS: FORM SELECTORS

---

- You can use form selectors to deal with form elements
- They work like other selectors but start with a colon (:) like a regular filter

SELECTOR	PURPOSE
:input	Finds all input, select, textarea, and button elements
:text	Finds all text elements
:password	Finds all password elements
:radio	Finds all radio button elements
:checkbox	Finds all checkbox elements
:submit	Finds all submit elements
:reset	Finds all reset elements
:image	Finds all image elements
:button	Finds all button elements
:file	Finds all file upload elements

## JQUERY SELECTORS AND FILTERS: FORM SELECTORS

---

- You can perform additional filtering of form elements, such as whether items are checked, selected, or enabled

SELECTOR	PURPOSE
:enabled	Matches all form elements that are enabled
:disabled	Matches all form elements that are disabled
:checked	Matches all form elements that are checked (radiobuttons and checkboxes)
:selected	Matches all form elements that are selected

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

# Example Form Document

First Name

Last Name

Gender  M  F

What products are you interested in?

Widgets  
 Hibbity Jibbities  
 SplashBangers  
 Whatzits

Comments:

Optional life story file

## Example Form Document

First Name

Last Name

Gender  M  F

What products are you interested in?

Widgets  
 Hibbity Jibbities  
 SplashBangers  
 Whatzits

Comments:

Optional life story file

```
$( "form :input" ).css( "border", "3px solid red" );
```

Created with

 **nitroPDF®** professional

download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

## Example Form Document

First Name

Last Name

Gender  M  F

What products are you interested in?

Widgets  
 Hibbity Jibbities  
 SplashBangers  
 Whatzits

Comments:

Optional life story file

```
$(“form :text”).css(“border”, “3px solid red”);
```

## Example Form Document

First Name

Last Name

Disabled Text Field

Gender  M  F

What products are you interested in?

Widgets  
 Hibbity Jibbities  
 SplashBangers  
 Whatzits

Comments:

Optional life story file

## Example Form Document

First Name

Last Name

Disabled Text Field

Gender  M  F

What products are you interested in?

Widgets  
 Hibbity Jibbities  
 SplashBangers  
 Whatzits

Comments:

Optional life story file

```
$(“form :text:enabled”).css(“border”, “3px solid red”);
```

Created with

 **nitroPDF®** professional

download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

# Example Form Document

First Name	<input type="text"/>
Last Name	<input type="text"/>
Disabled Text Field	<input type="text" value="td"/>
Gender	<input checked="" type="radio"/> M <input type="radio"/> F
What products are you interested in?	<input checked="" type="checkbox"/> Widgets <input type="checkbox"/> Hibbity Jibbities <input checked="" type="checkbox"/> SplashBangers <input type="checkbox"/> Whatzits
Comments:	<input type="text"/>
Optional life story file	<input type="file"/> <input type="button" value="Browse..."/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>

## Example Form Document

First Name	<input type="text"/>
Last Name	<input type="text"/>
Disabled Text Field	<input type="text" value="td"/>
Gender	<input checked="" type="radio"/> M <input checked="" type="radio"/> F
What products are you interested in?	<input checked="" type="checkbox"/> Widgets <input type="checkbox"/> Hibbity Jibbities <input checked="" type="checkbox"/> SplashBangers <input type="checkbox"/> Whatzits
Comments:	<input type="text"/>
Optional life story file	<input type="file"/> <input type="button" value="Browse..."/>
	<input type="button" value="submit"/> <input type="button" value="reset"/>

`$(“form :checked”).css(“border”, “3px solid red”);`

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

## Example Form Document

First Name

Last Name

Disabled Text Field

Gender  M  F  
 Widgets  
 Hibbity Jibbies  
 SplashBangers  
 Whatzits

What products are you interested in?

Comments:

Optional life story file

```
$( "form :checkbox:checked" ).css("border", "3px solid red");
```

## TRAVERSING DOCUMENT INFORMATION

---

- You can traverse the information returned from a document easily

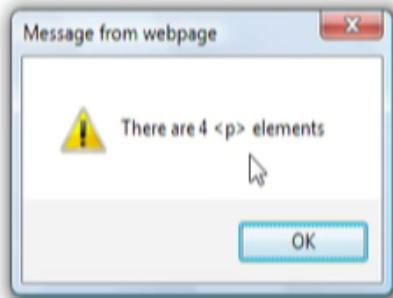
FUNCTION / PROPERTY	PURPOSE
size(), length	The number of elements in the jQuery result set
get()	Returns an array of all matched DOM elements. Useful if you need to operate on the DOM elements themselves instead of using built-in jQuery functions
get(index)	Access a single matched DOM element at a specified index in the matched set
find(expression)	Searches for descendent elements that match the specified expression
each(fn)	Execute a function within the context of every matched element

Created with

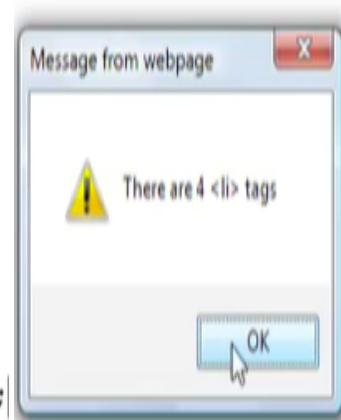


download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

```
alert("There are " + $("p").length + " <p> elements");
```



```
var elems = $('li').get();
alert("There are " + elems.length + " <li> tags");
```



```
alert($('li').get(0));
```



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

This is paragraph 1

This is paragraph 2

This is paragraph 3

```
$( "ul" ).find("li.b").css("border", "3px solid red"); This is paragraph 4
```

```
var leftmargin = 0;
var border = 3;
$("p").each(function() {
    $(this).css("border", border + "px solid red");
    $(this).css("margin-left", leftmargin);
    border += 2;
    leftmargin += 10;
});
```

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

This is paragraph 1

This is paragraph 2

This is paragraph 3

This is paragraph 4

## JQUERY STATEMENT CHAINING

---

- One of jQuery's most powerful features is its ability to chain multiple functions together to perform several operations in one line of code

```
$(selector).fn1().fn2().fn3();
```

Statement chain

# Example: Automatic Insertion of Icons for PDF Links

This example demonstrates how to examine the contents of a link to see if it points to a PDF file.

- [Link #1](#)
- Named Anchor Link
- [Link #2](#)
- [Link #3](#)
- [Link #4](#)
- [Link #5](#)
- [Link #6](#)
- [Link #7](#)
- [Email Link](mailto:joe@joe.com)

```
<ul class="navlist" id="navlinks">
    <li><a href="someurl.html">Link #1</a></li>
    <li><a name="#anchor1">Named Anchor Link</a></li>
    <li><a href="someurl.html">Link #2</a></li>
    <li><a href="someurl.pdf">Link #3</a></li>
    <li><a href="someurl.html">Link #4</a></li>
    <li><a href="someurl.html">Link #5</a></li>
    <li><a href="someurl.pdf">Link #6</a></li>
    <li><a href="someurl.html">Link #7</a></li>
    <li><a href="mailto:joe@joe.com">Email Link</a></li>
</ul>
.
.
.
$( "a[href$=.pdf]" ).after("<img src='images/small pdf icon.gif' align='absbottom' />")
```

# Example: Anchors

This example demonstrates how to use anchors.

- [Link #1](#)
- Named Anchor Link
- [Link #2](#)
- [Link #3](#) 
- [Link #4](#)
- [Link #5](#)
- [Link #6](#) 
- [Link #7](#)
- [Email Link](#)

## MANIPULATING CONTENT: OVERVIEW

---

- Once you've used selectors and filters to retrieve web page content, you usually want to do something with it
- Sometimes you want to create new content to dynamically add into the page
- jQuery has functions for creating, copying, deleting, and moving content around, as well as wrapping page content in other content
- jQuery provides cross-browser support for working with CSS, including positioning and sizing information

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

## CREATING, SETTING, AND GETTING CONTENT

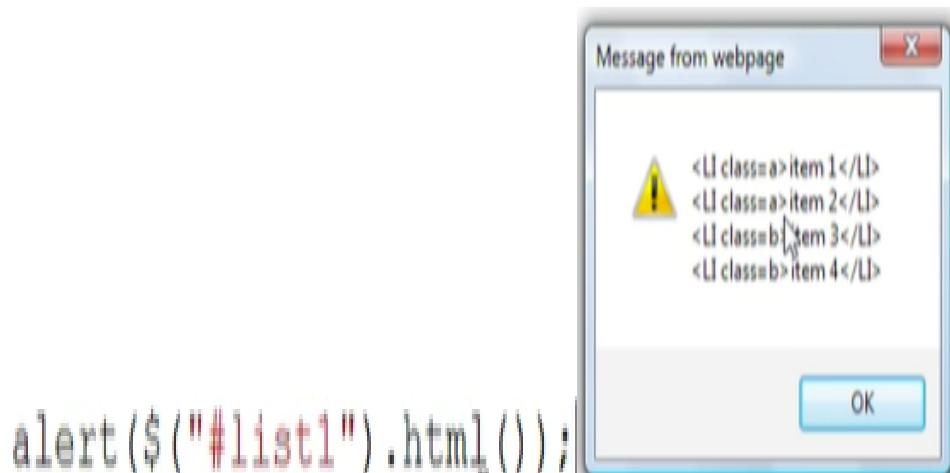
---

- To create new HTML content, you simply pass a string containing new HTML to the `$()` function:

```
var newHeader = $("<h1>My New Header</h1>");  
var myStr = "<h1>My New Header</h1>";  
var newHeader = $(myStr);
```

- In addition to this method, you can use the `html()` and `text()` methods to get and set content on

FUNCTION	PURPOSE
<code>html()</code>	Returns the HTML content of the first matched element
<code>html(newcontent)</code>	Sets the HTML content of every matched element
<code>text()</code>	Returns the text content of the first matched element
<code>text(newtext)</code>	Sets the text content for all matched elements



```
        • This is a new list item  
  
This is paragraph 1   
  
This is paragraph 2  
  
This is paragraph 3  
  
This is paragraph 4  
  
        • item 1  
        • item 2  
        • item 3  
        • item 4  
  
This is paragraph 1  
  
This is a new paragraph  
  
This is paragraph 3  
This is paragraph 4  
  
        • item 1  
        • item 2  
        • item 3  
        • item 4  
  
This is paragraph 1  
  
This is paragraph 2  
  
This is paragraph 3  
  
This is the last paragraph
```

## MANIPULATING ATTRIBUTES

---

- To inspect or change the value of attribute on elements, use jQuery's **attr** functions

FUNCTION	PURPOSE
attr(name)	Accesses property on the first matched element. This method makes it easy to retrieve a property value from the first matched element. If the element does not have an attribute with such a name, undefined is returned
attr(properties)	Sets a series of attributes on all matched elements using an object notation syntax. This is the best used for setting large numbers of properties at once <code>\$( "img" ).attr({ src: "/images/hat.gif", title: "jQuery", alt: "jQuery Logo" });</code>
attr(key, value)	Sets a single property to a value on all matched elements
attr(key, fn)	Sets a single property to a computed value, on all matched elements. Instead of supplying a string value, a function is provided that computes the value of the attribute
removeAttr(name)	Removes the named attribute from all matched elements

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)



```
<a href="images/Grass.jpg">
```

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

This open the image in new window

```
$( "a" ).removeAttr("href");
```

removes the link attribute

```
$( "img" ).attr({src:"images/Spring.jpg", alt:$});
```

changes the image

## INSERTING CONTENT

---

- jQuery provides several functions for inserting content into the document, both before and after existing page elements

FUNCTION	PURPOSE
append(content)	Appends content to the inside of every matched element
appendTo(selector)	Appends all of the matched elements to another, specified, set of elements
prepend(content)	Prepends content to the inside of every matched element
prependTo(selector)	Prepends all the matched elements to another, specified, set of elements
after(content)	Inserts content after each of the matched elements
before(content)	Inserts content before each of the matched elements
insertAfter(selector)	Inserts all of the matched elements after another, specified, set of elements
insertBefore(selector)	Inserts all the matched elements before another, specified, set of elements

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)