

2013

Arish Technology



[JQUERY OVERVIEW]

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: Write less, do more. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

TABLE OF CONTENTS

Introduction.....	3
What You Should Already Know	3
What is jQuery?.....	4
Why jQuery?.....	4
jQuery Install.....	4
Adding jQuery to Your Web Pages.....	4
Downloading jQuery	5
Alternatives to Downloading	5
Google	5
Microsoft.....	5
jQuery Syntax	6
jQuery Syntax	6
The Document Ready Event.....	6
jQuery Selectors	7
jQuery Selectors	7
The element Selector.....	7
The #id Selector	8
The .class Selector	9
More Examples of jQuery Selectors	10
jQuery Events.....	19
jQuery Event Functions.....	19
Functions In a Separate File	19
Functions In a Separate File	20
jQuery Name Conflicts	20
jQuery Events	21
jQuery Event Methods	22
jQuery Event Methods	22
jQuery Effects	25
jQuery hide() and show().....	25
jQuery toggle().....	27
jQuery Effects - Fade	28
jQuery Fading Methods	28
jQuery fadeIn() Method	28

jQuery fadeOut() Method	29
jQuery fadeTo() Method	30
jQuery Effects - Slide	31
jQuery slideDown(), slideUp(), and slideToggle()	31
jQuery slideDown().....	32
jQuery slideToggle ().....	33
jQuery slideDown().....	34
jQuery Effects - Animate	35
jQuery Animations	35
jQuery Callback Functions	37
jQuery Callback Functions	37
jQuery HTML/DOM Manipulation	39
Changing HTML Content.....	39
Adding HTML content.....	39
jQuery CSS Manipulation.....	42
jQuery css() Method	42
Return CSS Property.....	42
Set CSS Property and Value	43
Set Multiple CSS Properties	43
jQuery height() and width() Methods.....	44
jQuery AJAX.....	46
What is AJAX?	46
AJAX and jQuery	46
jQuery load() Method.....	47
jQuery \$.ajax() Method.....	47

LEARNING – JQUERY

INTRODUCTION

JQuery is a JavaScript Library.

JQuery greatly simplifies JavaScript programming.

JQuery is easy to learn.

Copy and paste code to my TryIt editor:

<http://www.maitrey.tk/blog/tryit/>

EXAMPLE #1

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>

<body>
<p>If you click on me, I will disappear.</p>
</body>
</html>
```

The purpose of jQuery is to make it much easier to use JavaScript on your website.

WHAT YOU SHOULD ALREADY KNOW

Before you start studying jQuery, you should have a basic knowledge of:

- HTML
- CSS
- JavaScript

WHAT IS JQUERY?

jQuery is a lightweight, "**write less, do more**", JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

jQuery takes a lot of common tasks that requires many lines of JavaScript code to accomplish, and wraps it into methods that you can call with a single line of code.

jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

Tip: In addition, jQuery has plugins for almost any task out there.

WHY JQUERY?

There are a lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.

The power, the plugins, and the syntax is what makes jQuery so great.

JQUERY INSTALL

ADDING JQUERY TO YOUR WEB PAGES

To use jQuery, you need to download the jQuery library (explained below), and include it on the pages you wish to use it.

The jQuery library is a single JavaScript file, and you reference to it using the HTML `<script>` tag:

```
<head>
<script src="jquery.js"></script>
</head>
```

Notice that the `<script>` tag should be inside the page's `<head>` section.

DOWNLOADING JQUERY

There are two versions of jQuery available for downloading:

- Production version - this is for your live website because it has been minified and compressed
- Development version - this is for testing and development (uncompressed and readable code)

Both versions can be downloaded from [jQuery.com](http://jquery.com).

Tip: Place the downloaded file in the same directory as the pages where you wish to use it. Or assign proper path to the script tag.

ALTERNATIVES TO DOWNLOADING

If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).

Both Google and Microsoft host jQuery.

To use jQuery from Google or Microsoft, use one of the following:

GOOGLE

```
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.0/jquery.min.js">
</script>
</head>
```

MICROSOFT

```
<head>
<script src="http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.8.0.js">
</script>
</head>
```

One big advantage of using the hosted jQuery from Google or Microsoft: Because the file comes from a common URL that other websites may use as well, chances are that when people reaches your website and their browser requests the jQuery framework, it may already be in the cache. Also, most CDN's will make sure that once a user requests a file from it, it's served from the server closest to them, so your European users won't have to get the file all the way from the US and vice versa.

JQUERY SYNTAX

With jQuery you select (query) HTML elements and perform "actions" on them.

JQUERY SYNTAX

The jQuery syntax is tailor made for **selecting** HTML elements and perform some **action** on the element(s).

Basic syntax is: `$(selector).action()`

- A \$ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

Examples:

`$(this).hide()` - hides the current element.

`$("p").hide()` - hides all <p> elements.

`$(".test").hide()` - hides all elements with class="test".

`$("#test").hide()` - hides the element with id="test".

jQuery uses CSS selector syntax.

You will learn more about the selector syntax in the next chapter of this tutorial.

THE DOCUMENT READY EVENT

```
$(document).ready(function(){  
  
    // jQuery methods go here...  
  
});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready, before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){  
  
    // jQuery methods go here...  
  
});
```

Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.

JQUERY SELECTORS

jQuery selectors are one of the most important parts of the jQuery library.

JQUERY SELECTORS

jQuery selectors allow you to select and manipulate HTML element(s).

With jQuery selectors you can find elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All type of selectors in jQuery, start with the dollar sign and parentheses: \$().

THE ELEMENT SELECTOR

The jQuery element selector selects elements based on their tag names.

You can select all <p> elements on a page like this:

```
$("p")
```

Example

When a user clicks on a button, all <p> elements will be hidden:

```
$(document).ready(function(){  
    $("button").click(function(){
```



```
    $("p").hide();
  });
});
```

EXAMPLE #2

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>

<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

THE #ID SELECTOR

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the element:

```
$("#test")
```

Example

When a user clicks on a button, the element with id="test" will be hidden:

EXAMPLE #3

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
```

```

$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
</script>
</head>

<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

THE .CLASS SELECTOR

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

```
$(".test")
```

Example

When a user clicks on a button, the elements with class="test" will be hidden:

EXAMPLE #4

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
</script>
</head>
<body>

<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>

```

```

<button>Click me</button>
</body>
</html>

```

Please this is a request to go through all the examples below.

MORE EXAMPLES OF JQUERY SELECTORS

Syntax	Description	Example
<code>\$("*")</code>	Selects all elements	EX #5
<code>\$(this)</code>	Selects the current HTML element	EX #6
<code>\$(".intro")</code>	Selects all <p> elements with class="intro"	EX #7
<code>\$(".p:first")</code>	Selects the first <p> element	EX #8
<code>\$(".ul li:first")</code>	Selects the first element of the first 	EX #9
<code>\$(".ul li:first-child")</code>	Selects the first element of every 	EX #10
<code>\$(".href")</code>	Selects all elements with an href attribute	EX #11
<code>\$(".a[target='_blank']")</code>	Selects all <a> elements with a target attribute value equal to "_blank"	EX #12
<code>\$(".a[target!='_blank']")</code>	Selects all <a> elements with a target attribute value NOT equal to "_blank"	EX #13
<code>\$(".:button")</code>	Selects all <button> elements and <input> elements of type="button"	EX #14
<code>\$(".tr:even")</code>	Selects all even <tr> elements	EX #15

EXAMPLES #5:

```

$("*")

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $(".button").click(function(){
        $(".*").hide();
    });
});

```

```
});  
</script>  
</head>  
<body>  
<h2>This is a heading</h2>  
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>  
<button>Click me</button>  
</body>  
</html>
```

EXAMPLES #6:

\$(this)

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="jquery.js"></script>  
<script>  
$(document).ready(function(){  
  $("button").click(function(){  
    $(this).hide();  
  });  
});  
</script>  
</head>  
<body>  
<h2>This is a heading</h2>  
<p>This is a paragraph.</p>  
<p>This is another paragraph.</p>  
<button>Click me</button>  
</body>  
</html>
```

EXAMPLE #7

\$("p.intro")

```
<!DOCTYPE html>  
<html>  
<head>  
<script src="jquery.js"></script>  
<script>  
$(document).ready(function(){  
  $("button").click(function(){  
    $("p.intro").hide();  
  });  
});  
</script>  
</head>  
<body>
```

```
<h2 class="intro">This is a heading</h2>
<p class="intro">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

EXAMPLE #8:

```
$("#p:first")
```

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    $("#p:first").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

EXAMPLE #9

```
$("#ul li:first")
```

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    $("#ul li:first").hide();
  });
});
</script>
</head>
<body>
<p>List 1:</p>
<ul>
  <li>Coffee</li>
  <li>Milk</li>
```

```

    <li>Tea</li>
  </ul>
<p>List 2:</p>
<ul>
  <li>Coffee</li>
  <li>Milk</li>
  <li>Tea</li>
</ul>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #10

```

$("ul li:first-child")

```

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("ul li:first-child").hide();
  });
});
</script>
</head>
<body>
<p>List 1:</p>
<ul>
  <li>Coffee</li>
  <li>Milk</li>
  <li>Tea</li>
</ul>
<p>List 2:</p>
<ul>
  <li>Coffee</li>
  <li>Milk</li>
  <li>Tea</li>
</ul>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #11

```

$("[href]")

```

```

<!DOCTYPE html>

```

```

<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("[href]").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p><a href="http://www.google.com">Google</a></p>
<p><a href="https://www.google.com/search?q=maitrey684">Maitrey684 - Google</a></p>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #12

```

$("[a[target='_blank']")

```

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("a[target='_blank']").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p><a href="http://www.google.com" target="_blank">Google</a></p>
<p><a href="https://www.google.com/search?q=maitrey684">Maitrey684 - Google</a></p>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #13

```

$("[a[target!='_blank']")

```

```

<!DOCTYPE html>

```

```

<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("a[target!='_blank']").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p><a href="http://www.google.com" target="_blank">Google</a></p>
<p><a href="https://www.google.com/search?q=maitrey684">Maitrey684 - Google</a></p>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #14

`$(":button")`

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(":button").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

EXAMPLE #15

`$("tr:even")` - Change **even** to **odd** and visa-versa

```

<!DOCTYPE html>
<html>
<head>

```



```
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("tr:even").css("background-color","red");
});
</script>
</head>
<body>
<h1>Welcome to My Web Page</h1>
<table>
<tr>
  <th>Company</th>
  <th>Contact</th>
  <th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Christina Berglund</td>
<td>Sweden</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
<tr>
<td>Ernst Handel</td>
<td>Roland Mendel</td>
<td>Austria</td>
</tr>
<tr>
<td>Island Trading</td>
<td>Helen Bennett</td>
<td>UK</td>
</tr>
<tr>
<td>Königlich Essen</td>
<td>Philip Cramer</td>
<td>Germany</td>
</tr>
<tr>
<td>Laughing Bacchus Winecellars</td>
<td>Yoshi Tannamuri</td>
<td>Canada</td>
</tr>
```

```

<tr>
<td>Magazzini Alimentari Riuniti</td>
<td>Giovanni Rovelli</td>
<td>Italy</td>
</tr>
<tr>
<td>North/South</td>
<td>Simon Crowther</td>
<td>UK</td>
</tr>
</table>
</body>
</html>

```

Most important table to remember:

Selector	Example	Selects
*	\$("#*")	All elements
#ID	\$("#lastname")	The element with id=lastname
.CLASS	\$(".intro")	All elements with class="intro"
ELEMENT	\$("p")	All p elements
.CLASS.CLASS	\$(".intro.demo")	All elements with the classes "intro" and "demo"
:first	\$("p:first")	The first p element
:last	\$("p:last")	The last p element
:even	\$("tr:even")	All even tr elements
:odd	\$("tr:odd")	All odd tr elements
:eq(INDEX)	\$("ul li:eq(3)")	The fourth element in a list (index starts at 0)
:gt(NO)	\$("ul li:gt(3)")	List elements with an index greater than 3
:lt(NO)	\$("ul li:lt(3)")	List elements with an index less than 3
:not(SELECTOR)	\$("input:not(:empty)")	All input elements that are not empty
:header	\$(":header")	All header elements h1, h2 ...
:animated	\$(":animated")	All animated elements
:contains(TEXT)	\$(":contains('Indusa')")	All elements which contains the text
:empty	\$(":empty")	All elements with no child (elements) nodes
:hidden	\$("p:hidden")	All hidden p elements
:visible	\$("table:visible")	All visible tables
S1,S2,S3	\$("th,td,.intro")	All elements with matching selectors
[ATTRIBUTE]	\$("[href]")	All elements with a href attribute
[ATTRIBUTE=VALUE]	\$("[href='default.htm']")	All elements with a href attribute value equal to "default.htm"
[ATTRIBUTE!=VALUE]	\$("[href!='default.htm']")	All elements with a href attribute value not equal to "default.htm"

[ATTRIBUTE \$= VALUE]	\$("[href\$='.jpg']")	All elements with a href attribute value ending with ".jpg"
[<i>attribute</i> ^= <i>value</i>]	\$("[href^='jquery_']")	All elements with a href attribute value starting with "jquery_"
:input	\$(":input")	All input elements
:text	\$(":text")	All input elements with type="text"
:password	\$(":password")	All input elements with type="password"
:radio	\$(":radio")	All input elements with type="radio"
:checkbox	\$(":checkbox")	All input elements with type="checkbox"
:submit	\$(":submit")	All input elements with type="submit"
:reset	\$(":reset")	All input elements with type="reset"
:button	\$(":button")	All input elements with type="button"
:image	\$(":image")	All input elements with type="image"
:file	\$(":file")	All input elements with 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

JQUERY EVENTS

jQuery is tailor made to handle events.

JQUERY EVENT FUNCTIONS

The jQuery event handling methods are core functions in jQuery.

Event handlers are methods that are called when "something happens" in HTML. The term **"triggered (or "fired") by an event"** is often used.

It is common to put jQuery code into event handler methods in the <head> section:

EXAMPLE #16

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

In the example above, a function is called when the click event for the button is triggered:

```
$("#button").click(function() {
    ...some code...
});
```

FUNCTIONS IN A SEPARATE FILE

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, put your jQuery functions in a separate .js file.

When we demonstrate jQuery here, the functions are added directly into the <head> section, However, sometimes it is preferable to place them in a separate file, like this (refer to the file with the src attribute):

FUNCTIONS IN A SEPARATE FILE

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, put your jQuery functions in a separate .js file.

When we demonstrate jQuery here, the functions are added directly into the <head> section, However, sometimes it is preferable to place them in a separate file, like this (refer to the file with the src attribute):

EXAMPLE

```
<head>
<script src="jquery.js"></script>
<script src="my_jquery_functions.js"></script>
</head>
```

JQUERY NAME CONFLICTS

jQuery uses the \$ sign as a shortcut for jQuery.

Some other JavaScript libraries also use the dollar sign for their functions.

The jQuery **noConflict()** method specifies a custom name (like jq), instead of using the dollar sign.

EXAMPLE #17

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
var jq=jQuery.noConflict();
jq(document).ready(function(){
  jq("button").click(function(){
    jq("p").hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
```

```
</body>  
</html>
```

JQUERY EVENTS

Here are some examples of event methods in jQuery:

Event Method	Description
<code>\$(document).ready(function)</code>	Binds a function to the ready event of a document (when the document is finished loading)
<code>\$(selector).click(function)</code>	Triggers, or binds a function to the click event of selected elements
<code>\$(selector).dblclick(function)</code>	Triggers, or binds a function to the double click event of selected elements
<code>\$(selector).focus(function)</code>	Triggers, or binds a function to the focus event of selected elements
<code>\$(selector).mouseover(function)</code>	Triggers, or binds a function to the mouseover event of selected elements

JQUERY EVENT METHODS

JQUERY EVENT METHODS

Event methods trigger, or bind a function to an event for all matching elements.

Trigger example:

`$("#button").click()` - triggers the click event for a button element.

Binding example:

`$("#button").click(function(){ $("#img").hide() })` - binds a function to the click event.

The following table lists all the methods used to handle events.

Method	Description
<code>bind()</code>	Add one or more event handlers to matching elements
<code>blur()</code>	Triggers, or binds a function to the blur event of selected elements
<code>change()</code>	Triggers, or binds a function to the change event of selected elements
<code>click()</code>	Triggers, or binds a function to the click event of selected elements
<code>dblclick()</code>	Triggers, or binds a function to the dblclick event of selected elements
<code>delegate()</code>	Add one or more event handlers to current, or future, specified child elements of the matching elements
<code>die()</code>	Remove all event handlers added with the <code>live()</code> function
<code>error()</code>	Triggers, or binds a function to the error event of selected elements
<code>event.currentTarget</code>	The current DOM element within the event bubbling phase
<code>event.data</code>	Contains the optional data passed to <code>jQuery.fn.bind</code> when the current executing handler was bound
<code>event.isDefaultPrevented()</code>	Returns whether <code>event.preventDefault()</code> was called for the event object
<code>event.isImmediatePropagationStopped()</code>	Returns whether <code>event.stopImmediatePropagation()</code> was called for the event object
<code>event.isPropagationStopped()</code>	Returns whether <code>event.stopPropagation()</code> was called for the event object
<code>event.pageX</code>	The mouse position relative to the left edge of the document
<code>event.pageY</code>	The mouse position relative to the top edge of the document

event.preventDefault()	Prevents the default action of the event
event.relatedTarget	The other DOM element involved in the event, if any
event.result	This attribute contains the last value returned by an event handler that was triggered by this event, unless the value was undefined
event.stopImmediatePropagation()	Prevents other event handlers from being called
event.stopPropagation()	Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event
event.target	The DOM element that initiated the event
event.timeStamp	This attribute returns the number of milliseconds since January 1, 1970, when the event is triggered
event.type	Describes the nature of the event
event.which	Which key or button was pressed for a key or button event
focus()	Triggers, or binds a function to the focus event of selected elements
focusin()	Binds a function to the focusin event of selected elements
focusout()	Binds a function to the focusout event of selected elements
hover()	Binds one or two functions to the hover event of selected elements
keydown()	Triggers, or binds a function to the keydown event of selected elements
keypress()	Triggers, or binds a function to the keypress event of selected elements
keyup()	Triggers, or binds a function to the keyup event of selected elements
live()	Add one or more event handlers to current, or future, matching elements
load()	Triggers, or binds a function to the load event of selected elements
mousedown()	Triggers, or binds a function to the mouse down event of selected elements
mouseenter()	Triggers, or binds a function to the mouse enter event of selected elements
mouseleave()	Triggers, or binds a function to the mouse leave event of selected elements
mousemove()	Triggers, or binds a function to the mouse move event of selected elements
mouseout()	Triggers, or binds a function to the mouse out event of selected elements
mouseover()	Triggers, or binds a function to the mouse over event of selected elements
mouseup()	Triggers, or binds a function to the mouse up event of selected elements
one()	Add one or more event handlers to matching elements. This handler can only be triggered once per element

ready()	Binds a function to the ready event of a document (when an HTML document is ready to use)
resize()	Triggers, or binds a function to the resize event of selected elements
scroll()	Triggers, or binds a function to the scroll event of selected elements
select()	Triggers, or binds a function to the select event of selected elements
submit()	Triggers, or binds a function to the submit event of selected elements
toggle()	Binds two or more functions to the toggle between for the click event for selected elements
trigger()	Triggers all events bound to the selected elements
triggerHandler()	Triggers all functions bound to a specified event for the selected elements
unbind()	Remove an added event handler from selected elements
undelegate()	Remove an event handler to selected elements, now or in the future
unload()	Triggers, or binds a function to the unload event of selected elements

JQUERY EFFECTS

JQUERY EFFECTS

Hide, Show, Toggle, Slide, Fade, and Animate.

JQUERY HIDE() AND SHOW()

With jQuery, you can hide and show HTML elements with the `hide()` and `show()` methods:

Example:

```
$("#hide").click(function(){  
    $("p").hide();  
});
```

```
$("#show").click(function(){  
    $("p").show();  
});
```

Syntax:

```
$(selector).hide(speed,callback);
```

```
$(selector).show(speed,callback);
```

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", "normal", or milliseconds.

The optional callback parameter is the name of a function to be executed after the hide (or show) function completes. You will learn more about the callback parameter in the next chapter of this tutorial.

The following example demonstrates the speed parameter with `hide()`:

Example:

```
$("#button").click(function(){  
    $("p").hide(1000);  
});
```

EXAMPLE #18:

Simple `hide()` method.

```
<!DOCTYPE html>  
<html>
```

```

<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
</script>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body>
</html>

```

EXAMPLE #19:

Hide() with effect

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $(".ex .hide").click(function(){
        $(this).parents(".ex").hide("slow");
    });
});
</script>
<style type="text/css">
div.ex
{
background-color:#e5eccc;
padding:7px;
border:solid 1px #c3c3c3;
}
</style>
</head>
<body>
<h3>Island Trading</h3>
<div class="ex">
<button class="hide">Hide me</button>
<p>Contact: Helen Bennett<br />
Garden House Crowther Way<br />
London</p>
</div>
<h3>Paris spécialités</h3>
<div class="ex">
<button class="hide">Hide me</button>

```

```
<p>Contact: Marie Bertrand<br />
265, Boulevard Charonne<br />
Paris</p>
</div>
</body>
</html>
```

JQUERY TOGGLE()

With jQuery, you can toggle between the `hide()` and `show()` methods with the `toggle()` method.

Shown elements are hidden and hidden elements are shown:

Example:

```
$("#button").click(function(){
    $("#p").toggle();
});
```

Syntax:

```
$(selector).toggle(speed,callback);
```

The optional speed parameter can take the following values: "slow", "fast", "normal", or milliseconds.

The optional callback parameter is the name of a function to be executed after the `toggle()` method completes.

EXAMPLE #20:

```
Toggle()
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        $("#p").toggle();
    });
});
</script>
</head>
<body>
<button>Toggle</button>
<p>This is a paragraph with little content.</p>
<p>This is another small paragraph.</p>
</body>
</html>
```

JQUERY EFFECTS - FADE

With jQuery you can fade elements in and out of visibility.

JQUERY FADING METHODS

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

- `fadeIn()`
- `fadeOut()`
- `fadeTo()`

JQUERY FADEIN() METHOD

The jQuery `fadeIn()` method is used to fade in a hidden element.

Syntax:

`$(selector).fadeIn(speed,callback);`

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is the name of a function to be executed after the fading completes.

The following example demonstrates the `fadeIn()` method with different parameters:

Example:

```
$("#button").click(function(){
    $("#div1").fadeIn();
    $("#div2").fadeIn("slow");
    $("#div3").fadeIn(3000);
});
```

EXAMPLE #21:

FadeIn() example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="jquery.js"></script>
```

```
<script>
```

```

$(document).ready(function(){
    $("button").click(function(){
        $("#div1").fadeIn();
        $("#div2").fadeIn("slow");
        $("#div3").fadeIn(3000);
    });
});
</script>
</head>
<body>
<p>Demonstrate fadeIn() with different parameters.</p>
<button>Click to fade in boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;display:none;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;display:none;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;display:none;background-color:blue;"></div>
</body>
</html>

```

JQUERY FADEOUT() METHOD

The jQuery fadeOut() method is used to fade out a visible element.

Syntax:

```
$(selector).fadeOut(speed,callback);
```

The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional callback parameter is the name of a function to be executed after the fading completes.

The following example demonstrates the fadeOut() method with different parameters:

Example

```

$("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
});

```

EXAMPLE #22:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>

```

```

<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut();
    $("#div2").fadeOut("slow");
    $("#div3").fadeOut(3000);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeOut() with different parameters.</p>
<button>Click to fade out boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>

```

JQUERY FADETO() METHOD

The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).

Syntax:

```
$(selector).fadeTo(speed,opacity,callback);
```

The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1).

The optional callback parameter is the name of a function to be executed after the function completes.

The following example demonstrates the fadeTo() method with different parameters:

EXAMPLE:

```

$("button").click(function(){
  $("#div1").fadeTo("slow",0.15);
  $("#div2").fadeTo("slow",0.4);
  $("#div3").fadeTo("slow",0.7);
});

```

EXAMPLE #23:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").fadeOut("slow",0.15);
    $("#div2").fadeOut("slow",0.4);
    $("#div3").fadeOut("slow",0.7);
  });
});
</script>
</head>
<body>
<p>Demonstrate fadeTo() with different parameters.</p>
<button>Click to fade boxes</button>
<br><br>
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>

```

JQUERY EFFECTS - SLIDE

The jQuery slide methods gradually change the height for selected elements.

JQUERY SLIDEDOWN(), SLIDEUP(), AND SLIDETOGGLE()

The jQuery slide methods gradually change the height for selected elements.

jQuery has the following slide methods:

```
$(selector).slideDown(speed,callback);
```

```
$(selector).slideUp(speed,callback);
```

```
$(selector).slideToggle(speed,callback);
```

The optional speed parameter can take the following values: "slow", "fast", "normal", or milliseconds.

The optional callback parameter is the name of a function to be executed after the function completes.

Examples

slideDown() Example

```
$(".flip").click(function(){
    $(".panel").slideDown();
});
```

EXAMPLE #24:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $(".flip").click(function(){
        $(".panel").slideDown("slow");
    });
});
</script>
<style type="text/css">
div.panel,p.flip
{
margin:0px;
padding:5px;
text-align:center;
background:#e5eccc;
border:solid 1px #c3c3c3;
}
div.panel
{
height:120px;
display:none;
}
</style>
</head>
<body>
<div class="panel">
<p>Because time is valuable, we deliver quick and easy learning.</p>
<p>At Indusa, you can study everything you need to learn, in an accessible and handy format.</p>
</div>
<p class="flip">Show Panel</p>
</body>
</html>
```

slideUp() Example

```
$(".flip").click(function(){
    $(".panel").slideUp()
});
```

EXAMPLE #25

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $(".flip").click(function(){
        $(".panel").slideUp("slow");
    });
});
</script>
<style type="text/css">
div.panel,p.flip
{
margin:0px;
padding:5px;
text-align:center;
background:#e5eccc;
border:solid 1px #c3c3c3;
}
div.panel
{
height:120px;
}
</style>
</head>
<body>
<div class="panel">
<p>Because time is valuable, we deliver quick and easy learning.</p>
<p>At Indusa, you can study everything you need to learn, in an accessible and handy format.</p>
</div>
<p class="flip">Hide Panel</p>
</body>
</html>
```

slideToggle() Example

```
$(".flip").click(function(){
    $(".panel").slideToggle();
});
```

EXAMPLE #26:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
$(".flip").click(function(){
    $(".panel").slideToggle("slow");
});
});
</script>
<style type="text/css">
div.panel,p.flip
{
margin:0px;
padding:5px;
text-align:center;
background:#e5eccc;
border:solid 1px #c3c3c3;
}
div.panel
{
height:120px;
display:none;
}
</style>
</head>
<body>
<div class="panel">
<p>Because time is valuable, we deliver quick and easy learning.</p>
<p>At Indusa, you can study everything you need to learn, in an accessible and handy format.</p>
</div>
<p class="flip">Show/Hide Panel</p>
</body>
</html>
```

JQUERY EFFECTS - ANIMATE

JQUERY ANIMATIONS

jQuery's syntax for making custom animations is:

```
$(selector).animate({params},duration,easing,callback);
```

The key parameter here is **params** that defines the CSS properties to be animated. Multiple properties can be animated at the same time:

```
animate({width:"70%",opacity:0.4,marginLeft:"0.6in",fontSize:"3em"});
```

The optional duration parameter specifies the speed of the animation, and can take the following values: "slow", "fast", "normal", or milliseconds.

The optional callback parameter is the name of a function to be executed after the function completes.

Examples

The example below animates the height, width, and opacity of a <div> element:

EXAMPLE

```
$("#button").click(function(){
    $("#div").animate({height:300,opacity:0.4},"slow");
    $("#div").animate({width:300,opacity:0.8},"slow");
    $("#div").animate({height:100,opacity:0.4},"slow");
    $("#div").animate({width:100,opacity:0.8},"slow");
});
```

EXAMPLE #27:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        $("#div").animate({height:300,opacity:0.4},"slow");
        $("#div").animate({width:300,opacity:0.8},"slow");
        $("#div").animate({height:100,opacity:0.4},"slow");
        $("#div").animate({width:100,opacity:0.8},"slow");
    });
});
</script>
```

```

</head>
<body>
<button>Start Animation</button>
<br /><br />
<div style="background:#98bf21;height:100px;width:100px;">
</div>
</body>
</html>

```

The example below moves a <div> element and increases the font size of the text:

EXAMPLE

```

$("button").click(function(){
    $("div").animate({left:"100px"},"slow");
    $("div").animate({fontSize:"3em"},"slow");
});

```

EXAMPLE #28:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({left:"100px"},"slow");
        $("div").animate({fontSize:"3em"},"slow");
    });
});
</script>
</head>
<body>
<button>Start Animation</button>
<br /><br />
<div style="background:#98bf21;height:100px;width:200px;position:relative;">HELLO</div>
</body>
</html>

```

By default, an HTML element cannot be moved. The position is static.

To manipulate the position of an HTML element, you must first set the CSS position property of the element to relative, fixed, or absolute.

JQUERY CALLBACK FUNCTIONS

JQUERY CALLBACK FUNCTIONS

A callback function is executed after the current effect is 100% finished.

JQUERY CALLBACK FUNCTIONS

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.

To prevent this, you can create a callback function.

A callback function is executed after the current effect is finished.

Typical syntax: `$(selector).hide(speed,callback);`

Examples

The example below has a callback parameter that is a function that will be executed after the hide effect is completed:

Example with Callback

```
$("#button").click(function(){
    $("#p").hide("slow",function(){
        alert("The paragraph is now hidden");
    });
});
```

EXAMPLE #29:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        $("#p").hide("slow",function(){
            alert("The paragraph is now hidden");
        });
    });
});
</script>
</head>
```

```
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body>
</html>
```

The example below has no callback parameter, and the alert box will be displayed before the hide effect is completed:

Example without Callback

```
$("#button").click(function(){
    $("#p").hide(1000);
    alert("The paragraph is now hidden");
});
```

EXAMPLE #30:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("#button").click(function(){
        $("#p").hide(1000);
        alert("The paragraph is now hidden");
    });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body>
</html>
```

JQUERY HTML/DOM MANIPULATION

jQuery contains powerful methods for changing and manipulating HTML elements and attributes.

CHANGING HTML CONTENT

The jQuery `html()` method changes the content (innerHTML) of the selected HTML elements.

EXAMPLE

```
$("p").html("Indusa");
```

EXAMPLE #31:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").html("Indusa");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

ADDING HTML CONTENT

The jQuery **append()** method adds content at the end of the selected HTML elements.

EXAMPLE

```
$("p").append(" Indusa.");
```

EXAMPLE #32:

```
<!DOCTYPE html>
<html>
```



```

<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").append(" <b>Indusa</b>.");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

The jQuery **prepend()** method adds content at the beginning of the selected HTML elements.

EXAMPLE

```

$("p").prepend("Indusa. ");

```

EXAMPLE #33:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").prepend("<b>Indusa</b>. ");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

The jQuery **after()** method inserts HTML content after the selected HTML elements.

EXAMPLE

```

$("p").after("Indusa");

```

EXAMPLE #34:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").after("Indusa");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

The jQuery **before()** method inserts HTML content before the selected HTML elements.

EXAMPLE

```

$("p").before("W3Schools");

```

EXAMPLE #35:

```

<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").before("Indusa");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>

```

JQUERY CSS MANIPULATION

JQUERY CSS MANIPULATION

JQUERY CSS() METHOD

jQuery has one important method for CSS manipulation: `css()`

The `css()` method has three different syntaxes, to perform different tasks:

- `css(property)` - Return CSS property value
- `css(property,value)` - Set CSS property and value
- `css({properties})` - Set multiple CSS properties and values

RETURN CSS PROPERTY

Use `css(property)` to return the specified CSS property value of the FIRST matched element:

EXAMPLE

```
$("#p").css("background");
```

EXAMPLE #36:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    alert($("#p").css("background"));
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p style="background:#ff0000">This is a paragraph.</p>
<p style="background:#00ff00">This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

SET CSS PROPERTY AND VALUE

Use `css(property,value)` to set the specified CSS property for ALL matched elements:

EXAMPLE

```
$("#p").css("background","yellow");
```

EXAMPLE #37:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    $("#p").css("background","yellow");
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

SET MULTIPLE CSS PROPERTIES

Use `css({properties})` to set multiple CSS properties for ALL matched elements:

EXAMPLE

```
$("#p").css({"background":"yellow","font-size":"200%"});
```

EXAMPLE #38:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("#button").click(function(){
    $("#p").css({"background":"yellow","font-size":"200%"});
  });
});
</script>
```

```
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

JQUERY HEIGHT() AND WIDTH() METHODS

jQuery also has two important methods for size manipulation:

- height()
- width()

The height() method returns the height of the FIRST matched element.

The height() method sets the height for ALL matched element:

EXAMPLE

```
$("div").height("150px");
```

EXAMPLE #39:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").height("150px");
  });
});
</script>
</head>
<body>
<div style="background:yellow;height:40px;width:100px">HELLO</div>
<div style="background:yellow;height:80px;width:100px">W3SCHOOLS</div>
<button>Click me</button>
</body>
</html>
```

The width() method returns the width of the FIRST matched element.

The width() method sets the width for ALL matched element:

EXAMPLE

```
$("div").width("300px");
```

EXAMPLE #40:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").width("300px");
    });
});
</script>
</head>
<body>
<div style="background:yellow;height:40px;width:100px">HELLO</div>
<div style="background:yellow;height:80px;width:100px">W3SCHOOLS</div>
<button>Click me</button>
</body>
</html>
```

JQUERY AJAX

JQUERY AJAX

jQuery has a rich library of methods for AJAX development.

jQuery AJAX Example

EXAMPLE #41:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").load('test1.txt');
  });
});
</script>
</head>
<body>
<div><h2>Let AJAX change this text</h2></div>
<button>Change Content</button>
</body>
</html>
```

WHAT IS AJAX?

AJAX = Asynchronous JavaScript and XML.

AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

AJAX AND JQUERY

jQuery provides a rich set of methods for AJAX web development.

With jQuery AJAX, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post.

And you can load remote data directly into the selected HTML elements of your web page!

JQUERY LOAD() METHOD

The jQuery load() method is a powerful AJAX function. It loads data from a server and puts the returned data into the selected element.

The load() method has the following syntax:

```
$(selector).load(URL,data,callback);
```

The required *URL* parameter specifies the URL to send the request to.

EXAMPLE

Change the content of a <div> element using an AJAX request:

```
$("div").load('test1.txt');
```

The optional data parameter specifies data to send to the server along with the request.

The optional callback parameter is the name of a function to be executed after the request is completed.

JQUERY \$.AJAX() METHOD

The jQuery \$.ajax() method also performs an AJAX request.

Syntax:

```
$.ajax(options);
```

\$.ajax() offers more functionality than load(), get(), and post(), but it is also more difficult to use.

The *options* parameter takes name:value pairs defining URL, passwords, data types, filters, character sets, timeout and error functions.

EXAMPLE

Change the content of a <div> element using an AJAX request:

```
$.ajax({url:"test1.txt",success:function(result){  
    $("div").html(result);  
}});
```


EXAMPLE #42:

```
<!DOCTYPE html>
<html>
<head>
<script src="jquery.js"></script>
<script>
$(document).ready(function(){
$("button").click(function(){
$.ajax({url:"test1.txt",success:function(result){
$("div").html(result);
}});
});
});
</script>
</head>
<body>
<div><h2>Let AJAX change this text</h2></div>
<button>Change Content</button>
</body>
</html>
```



Maitrey Patel

patel_maitrey684@yahoo.co.in | www.maitrey.tk

Connect me @     

Contact me:  patelmaitrey684  maitrey684  patel_maitrey684

