

JQuery

Part II

jQuery - Events Handling

- We have the ability to create dynamic web pages by using events. Events are actions that can be detected by your Web Application. Following are the examples events:
 - A mouse click
 - A web page loading
 - Taking mouse over an element
 - Submitting an HTML form
 - A keystroke on your keyboard
- Binding event handlers
 - Using the jQuery Event Model, we can establish event handlers on DOM elements with the **bind()** method as follows:

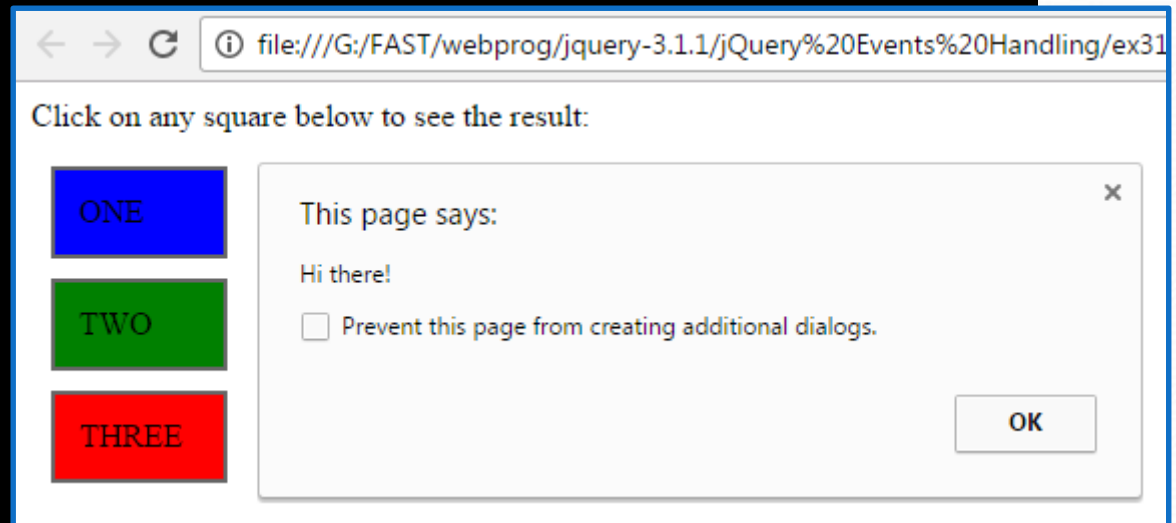
bind()

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
  $(document).ready(function() {
    $('div').bind('click', function( event ){
      alert('Hi there!');
    });
  });
</script>
<style>
  .div{padding:12px;
    border:2px;
    width:60px;}
</style>
```

.....

```
<p>Click on any square below to see the result:</p>
<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
```

.....



Removing event handlers

- Once an event handler is established, it remains in effect for the remainder of the life of the page.
- jQuery provides the `unbind()` command to remove an existing event handler. The syntax of `unbind()` is as follows:
 - `selector.unbind(eventType, handler)`
 - `selector.unbind(eventType)`
 - **eventType** – A string containing a JavaScript event type, such as `click` or `submit`.
 - **handler** – If provided, identifies the specific listener that's to be removed.

Event Manipulation Methods

bind(type, [data], fn)

Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.

off(events [, selector] [, handler(eventObject)])

This does the opposite of live, it removes a bound live event.

hover(over, out)

Simulates hovering for example moving the mouse on, and off, an object.

on(events [, selector] [, data], handler)

Binds a handler to an event (like click) for all current – and future – matched element. Can also bind custom events.

one(type, [data], fn)

Binds a handler to one or more events to be executed once for each matched element.

ready(fn)

Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.

trigger(event, [data])

Trigger an event on every matched element.

triggerHandler(event, [data])

Triggers all bound event handlers on an element.

unbind([type], [fn])

This does the opposite of bind, it removes bound events from each of the matched elements.

.on() / .off()

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
  $(document).ready(function() {
    function aClick() {
      $("div").show().fadeOut("slow"); }
    $("#bind").click(function () {
      $("#target").on("click", aClick).text("Can Click!");
    });
    $("#unbind").click(function () {
      $("#target").off("click", aClick).text("Does nothing...");
    }); });
```

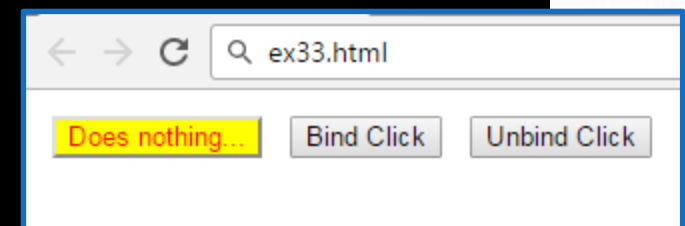
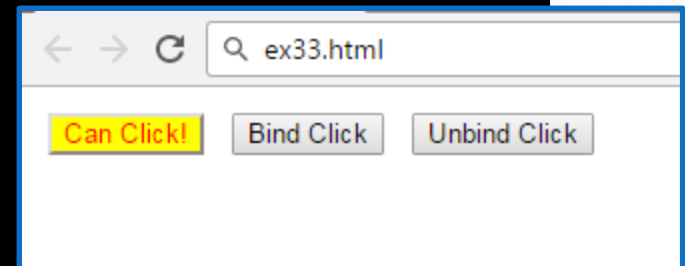
.....

```
button { margin:5px; }
button#target { color:red; background:yellow; }
```

.....

```
<button id = "target">Does nothing...</button>
<button id = "bind">Bind Click</button>
<button id = "unbind">Unbind Click</button>
<div style = "display:none;">Click!</div>
```

.....



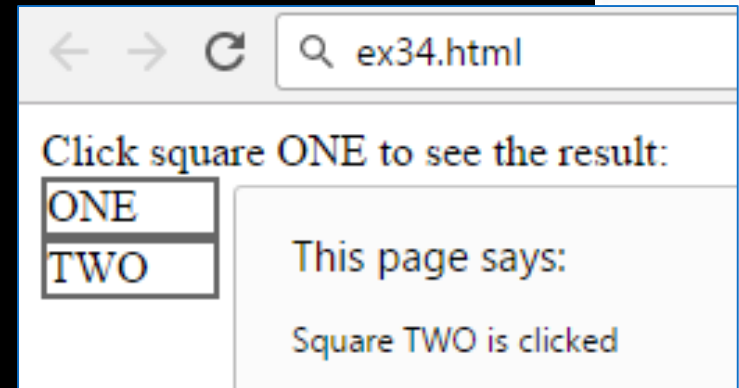
trigger()

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
    $("#div1").click( function () {
        $("#div2").trigger('click');
    });
    $("#div2").click( function () {
        alert( "Square TWO is clicked");
    });
});
</script>
<style>
.div{border:2px solid#666; width:60px;}
</style>
```

.....

```
<span>Click square ONE to see the result:</span>
<div class = "div" id = "div1" >ONE</div>
<div class = "div" id = "div2" >TWO</div>
```

.....



jQuery Effects

- Showing and Hiding elements
 - The commands for showing and hiding elements are pretty much what we would expect – `show()` to show the elements in a wrapped set and `hide()` to hide them.
- Syntax
- `[selector].show(speed, [callback]);`
- Here is the description of all the parameters
- `speed` – A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
- `callback` – This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

jQuery Effects - show / hide

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
```

```
    $(document).ready(function() {
        $("#show").click(function () {
            $(".mydiv").show( 1000 );
        });
        $("#hide").click(function () {
            $(".mydiv").hide( 1000 );
        }); });
```

```
</script>
```

```
<style>
```

```
    .mydiv{ border:2px solid #666; width:100px; height:100px;}
```

```
</style>
```

```
.....
```

```
<div class = "mydiv">
```

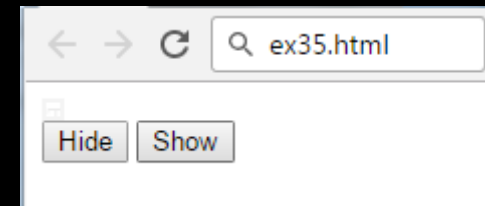
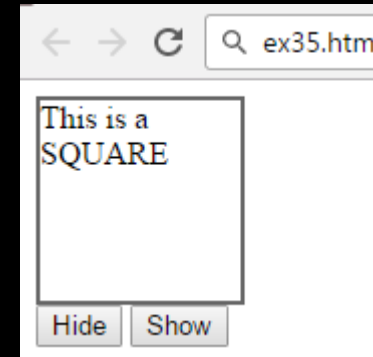
```
    This is a SQUARE
```

```
</div>
```

```
<input id = "hide" type = "button" value = "Hide" />
```

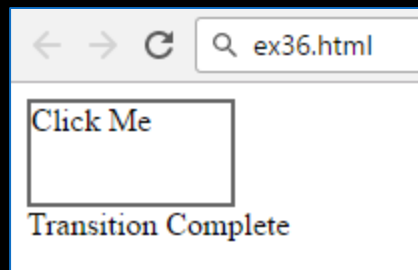
```
<input id = "show" type = "button" value = "Show" />
```

```
.....
```



Toggling the elements

- Here is the simple syntax for one of the toggle() methods:
- `[selector]..toggle([speed][, callback]);`
- Here is the description of all the parameters:
- speed – A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
- callback – This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.



JQuery Effect Methods

`animate(params, [duration, easing, callback])`

A function for making custom animations.

`fadeIn(speed, [callback])`

Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.

`fadeOut(speed, [callback])`

Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.

`fadeTo(speed, opacity, callback)`

Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.

`hide()`

Hides each of the set of matched elements if they are shown.

`hide(speed, [callback])`

Hide all matched elements using a graceful animation and firing an optional callback after completion.

`show()`

Displays each of the set of matched elements if they are hidden.

`show(speed, [callback])`

Show all matched elements using a graceful animation and firing an optional callback after completion.

`slideDown(speed, [callback])`

Reveal all matched elements by adjusting their height and firing an optional callback after completion.

`slideToggle(speed, [callback])`

Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.

`slideUp(speed, [callback])`

Hide all matched elements by adjusting their height and firing an optional callback after completion.

`stop([clearQueue, gotoEnd])`

Stops all the currently running animations on all the specified elements.

`toggle()`

Toggle displaying each of the set of matched elements.

`toggle(speed, [callback])`

Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.

`toggle(switch)`

Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).

`jQuery.fx.off`

Globally disable all animations. `jQuery.fx.off = true;`

jQuery Effects

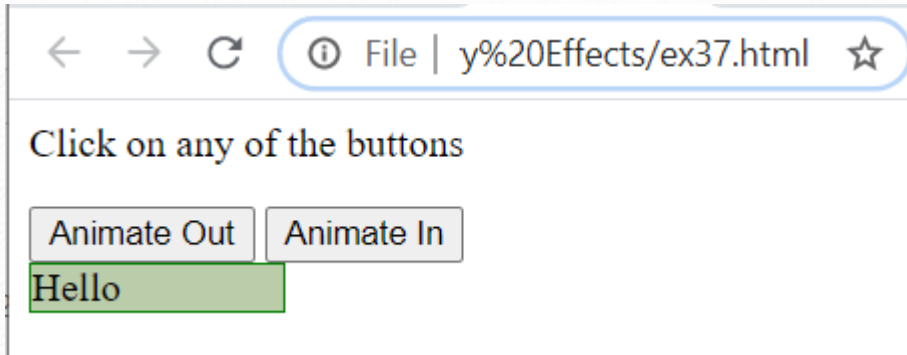
- Example 37
 - AnimateOut / AnimateIn
- Example 38
 - Opacity
- Example 39
 - jQueryUI - blinds
- Example 40
 - jQueryUI - bounce

jQuery Effects - animate

```
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
$("#out").click(function(){
    $("#block").animate({
        width: "70%",
        opacity: 0.4,
        marginLeft: "0.6in",
        fontSize: "3em",
        borderWidth: "10px"
    }, 1500 );
});
$("#in").click(function(){
    $("#block").animate({
        width: "100",
        opacity: 1.0,
        marginLeft: "0in",
        fontSize: "100%",
        borderWidth: "1px"
    }, 1500 );
});
});
</script>
....
```

```
<style>
div {background-color:#bca;
    width:100px;
    border:1px
    solid green;}
</style>
.....
<p>Click on any of the buttons</p>
<button id = "out"> Animate Out
</button>
<button id = "in"> Animate In
</button>
<div id = "block">Hello</div>
```


jQuery Effects - show / hide



jQuery Effects - fadeTo

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
  $(document).ready(function() {
    $("#more").click(function(){
      $(".target").fadeTo( 'slow', 0.7, function(){
        $(".log").text('More Opacity Transition Complete');
      });
    });
    $("#less").click(function(){
      $(".target").fadeTo( 'slow', 0.2, function(){
        $(".log").text('less Opacity Transition Complete');
      });
    });
  });
</script>
<style> p {background-color:#bca; width:200px;
border:1px solid green;} </style>
.....
<p>Click on any of the buttons</p>
<button id = "less"> Less Opacity </button>
<button id = "more"> More Opacity</button>
<div class = "target">
  <img src = "baby.png" alt = "baby" />
</div>
<div class = "log"></div>
```

← → ↻ ⓘ File | y%20Effects/ex38.html

Click on any of the buttons

Less Opacity

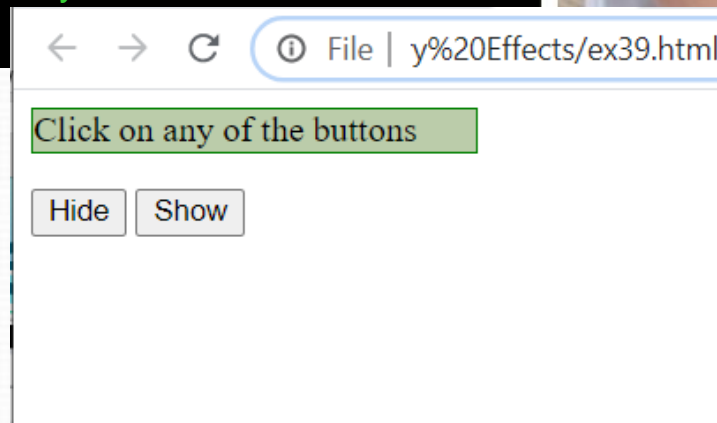
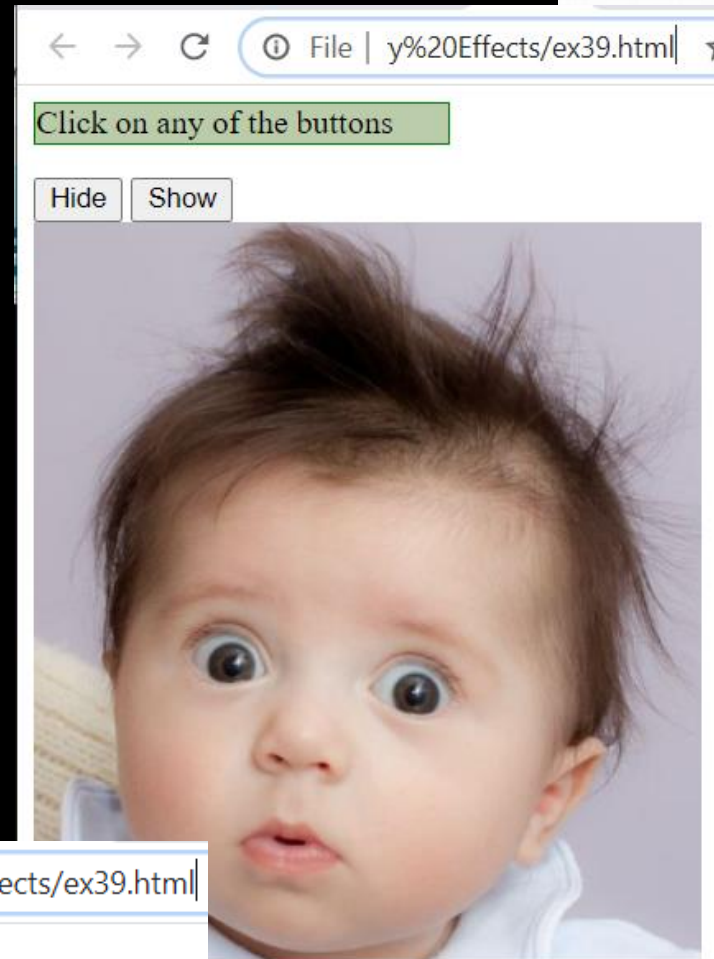
More Opacity



More Opacity Transition Complete

jQuery Effects - blind effect

```
<script src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $("#hide").click(function(){
            $(".target").hide( "blind", {direction: "horizontal"}, 1000 );
        });
        $("#show").click(function(){
            $(".target").show( "blind", {direction: "horizontal"}, 1000 );
        });
    });
</script>
<style> p {background-color:#bca; width:200px;
border:1px solid green;} </style>
.....
<p>Click on any of the buttons</p>
<button id = "hide"> Hide </button>
<button id = "show"> Show</button>
<div class = "target">
    <img src = "baby.png" alt = "baby" />
</div>
```



effect

```
<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $("#button").click(function(){
            $(".target").effect( "bounce", {times:3}, 300 );
        });
    });
</script>
<style> p {background-color:#bca; width:200px;
border:1px solid green;} </style>
.....
<p>Click the button</p>
<button id = "button"> Bounce </button>
<div class = "target"> <img src = "baby.png" alt = "baby" />
</div>
```



jQuery - Ajax

- AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology help us to load data from the server without a browser page refresh.
- Loading simple data
- This is very easy to load any static or dynamic data using JQuery AJAX. JQuery provides **load()** method to do the job:
- Syntax
- `[selector].load(URL, [data], [callback]);`

JQuery AJAX Methods

jQuery.ajax(options)

Load a remote page using an HTTP request.

jQuery.ajaxSetup(options)

Setup global settings for AJAX requests.

jQuery.get(url, [data], [callback], [type])

Load a remote page using an HTTP GET request.

jQuerygetJSON(url, [data], [callback])

Load JSON data using an HTTP GET request.

jQuery.getScript(url, [callback])

Loads and executes a JavaScript file using an HTTP GET request.

jQuery.post(url, [data], [callback], [type])

Load a remote page using an HTTP POST request.

load(url, [data], [callback])

Load HTML from a remote file and inject it into the DOM.

serialize()

Serializes a set of input elements into a string of data.

serializeArray()

Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.

JQuery AJAX Events

ajaxComplete(callback)

Attach a function to be executed whenever an AJAX request completes.

ajaxStart(callback)

Attach a function to be executed whenever an AJAX request begins and there is none already active.

ajaxError(callback)

Attach a function to be executed whenever an AJAX request fails.

ajaxSend(callback)

Attach a function to be executed before an AJAX request is sent.

ajaxStop(callback)

Attach a function to be executed whenever all AJAX requests have ended.

ajaxSuccess(callback)

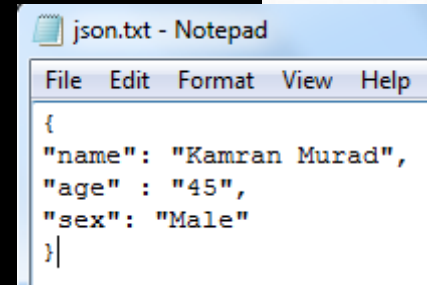
Attach a function to be executed whenever an AJAX request completes successfully.

Getting JSON data

- There would be a situation when server would return JSON string against your request. JQuery utility function `getJSON()` parses the returned JSON string and makes the resulting string available to the callback function as first parameter to take further action.
- Syntax
- `[selector].getJSON(URL, [data], [callback]);`

getJSON

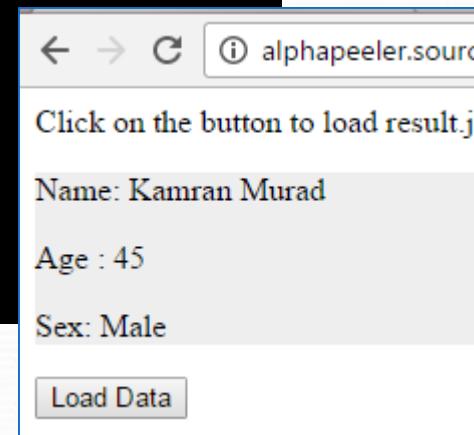
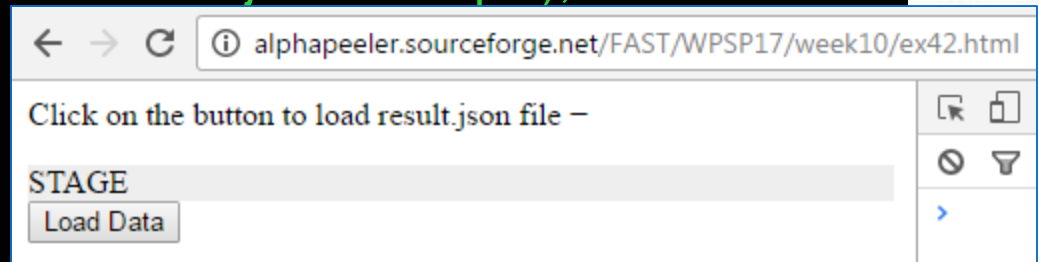
```
<script type = "text/javascript" src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $("#driver").click(function(event){
            $.getJSON('json.txt', function(jd) {
                $('#stage').html('<p> Name: ' + jd.name + '</p>');
                $('#stage').append('<p>Age : ' + jd.age+ '</p>');
                $('#stage').append('<p> Sex: ' + jd.sex+ '</p>');
            });
        });
    });
    ....
    <p>Click on the button to load result.json file -</p>
    <div id = "stage" style = "background-color:#eee;">
        STAGE
    </div>
    <input type = "button" id = "driver" value = "Load Data" />
    .....
```



json.txt - Notepad

File Edit Format View Help

```
{
  "name": "Kamran Murad",
  "age" : "45",
  "sex": "Male"
}
```



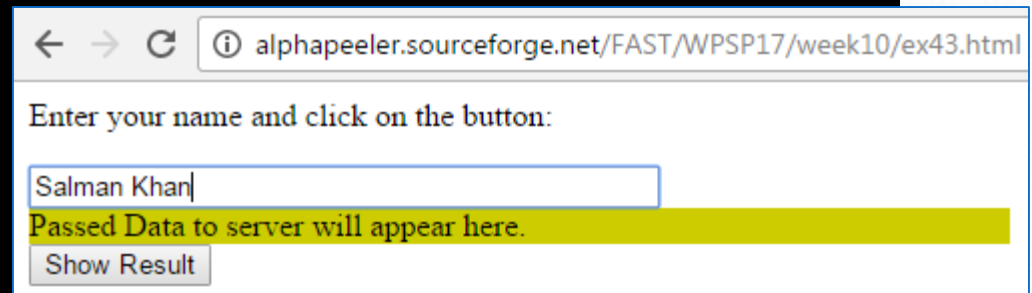
Passing data to the Server

- Many times you collect input from the user and you pass that input to the server for further processing. JQuery AJAX made it easy enough to pass collected data to the server using **data** parameter of any available Ajax method.

Passing data to the Server

```
<script type = "text/javascript" src = "jquery-3.6.0.js"></script>
<script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
        $("#driver").click(function(event){
            var name = $("#name").val();
            $("#stage").load('ex43.php', {"name":name} );
        });
    }); </script>
.....
```

```
<p>Enter your name and click on the button:</p>
<input type = "input" id = "name" size = "40" /><br />
<div id = "stage" style = "background-color:cc0;">
    Passed Data to server will appear here.
</div>
<input type = "button" id = "driver" value = "Show Result" />
.....
```



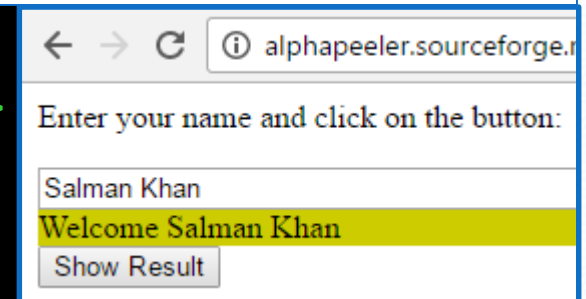
← → ↻ ⓘ alphapeeler.sourceforge.net/FAST/WPSP17/week10/ex43.html

Enter your name and click on the button:

Salman Khan

Passed Data to server will appear here.

Show Result



← → ↻ ⓘ alphapeeler.sourceforge.net/FAST/WPSP17/week10/ex43.html

Enter your name and click on the button:

Salman Khan

Welcome Salman Khan

Show Result

Passing data to the Server

- ex43.php

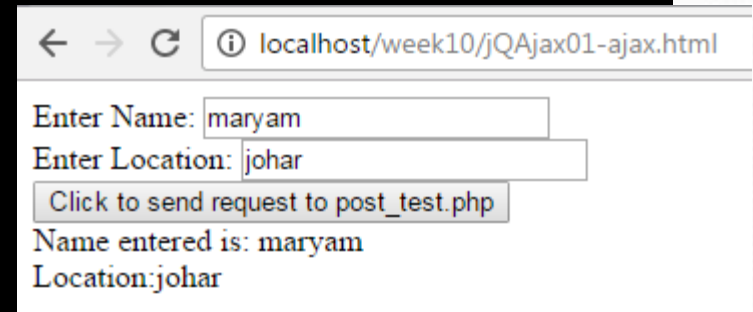
```
<?php
if( $_REQUEST["name"] ){

    $name =
    $_REQUEST['name'];
    echo "Welcome ". $name;
}
?>
```

.ajax ()

```
<script type = "text/javascript" src = "jquery-3.6.0.js"></script>
<script>
$(document).ready(function(){
$("#loaddata").click(function(){
txtname=$("#txtinput").val();
txtlocation=$("#txtlocation").val();
$.ajax({url:"post_test.php",data:{name:txtname, location: txtlocation
},success: function(ajaxresult){
$("#ajaxrequest").html(ajaxresult);
}});
});
});
```

```
....
<body>
Enter Name: <input type="text" id="txtinput"><br />
Enter Location: <input type="text" id="txtlocation"><br />
<button id="loaddata">Click to send request to post_test.php</button>
<div id="ajaxrequest"></div>
....
```

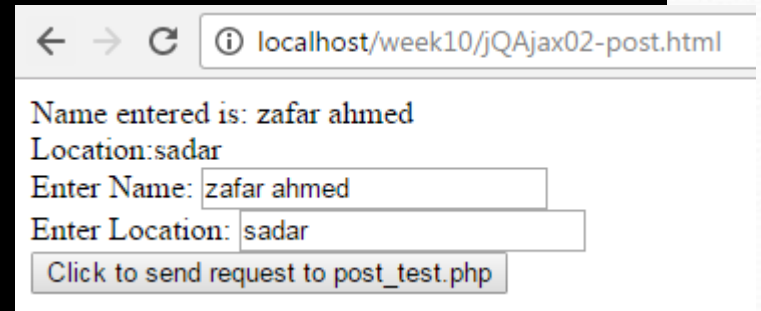


.post ()

```
<script type = "text/javascript" src = "jquery-3.6.0.js"></script>
<script>
$(document).ready(function(){
$("#loaddata").click(function(){
txtname=$("#txtinput").val();
txtlocation=$("#txtlocation").val();
$.post("post_test2.php",{ name:txtname, location: txtlocation
},function(ajaxresult){
$("#postrequest").html(ajaxresult);
});
});
});
```

.....

```
<body>
<div id="postrequest"></div>
Enter Name: <input type="text" id="txtinput"><br />
Enter Location: <input type="text" id="txtlocation"><br />
<button id="loaddata">Click to send request to post_test2.php</button>
</body>
</html>
```



← → ↻ ⓘ localhost/week10/jQAjax02-post.html

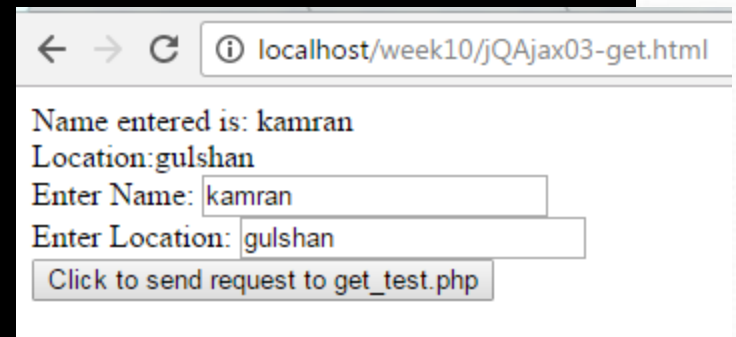
Name entered is: zafar ahmed
Location:sadar
Enter Name:
Enter Location:

.get()

```
<script type = "text/javascript" src = "jquery-3.6.0.js"></script>
<script>
$(document).ready(function(){
$("#loaddata").click(function(){
txtname=$("#txtname").val();
txtlocation=$("#txtlocation").val();
$.get("get_test.php",{ name:txtname, location: txtlocation
},function(ajaxresult){
$("#getrequest").html(ajaxresult);
});
});
});
```

.....

```
<body>
<div id="getrequest"></div>
Enter Name: <input type="text" id="txtname"><br />
Enter Location: <input type="text" id="txtlocation"><br />
<button id="loaddata">Click to send request to get_test.php</button>
```

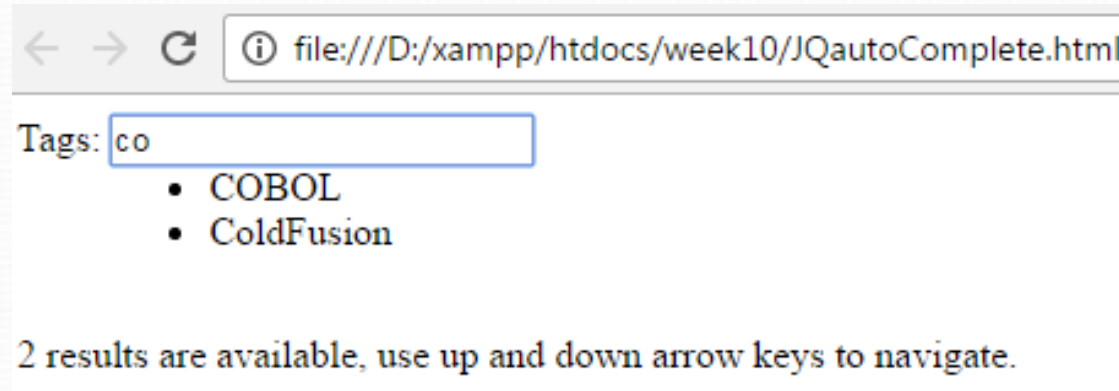


Auto-completing text fields

```
.....  
<meta name="viewport" content="width=device-width, initial-scale=1">  
<link rel="stylesheet" href="jquery-ui.css">  
<script src="jquery-1.12.4.js"></script>  
<script src="jquery-ui.js"></script>  
<script>  
$( function() {  
  var availableTags = [  
    "ActionScript", "AppleScript", "Asp",  
    "BASIC", "C", "C++", "Clojure", "COBOL",  
    "ColdFusion", "Erlang", "Fortran",  
    "Groovy", "Haskell", "Java", "JavaScript",  
    "Lisp", "Perl", "PHP", "Python", "Ruby",  
    "Scala", "Scheme"  
  ];  
  $( "#tags" ).autocomplete({  
    source: availableTags  
  });  
} );  
</script>
```

```
<body>  
  
<div class="ui-widget">  
  <label for="tags">Tags:  
</label>  
  <input id="tags">  
</div>  
  
</body>  
</html>
```

Using autocompleters



- Other examples:
- <https://www.petefreitag.com/item/605.cfm>
- <http://www.pontikis.net/blog/jquery-ui-autocomplete-step-by-step>
- <https://jqueryui.com/autocomplete/>

jQuery photo gallery

- <http://photoswipe.com/>
- <http://manos.malihu.gr/sideways-jquery-fullscreen-image-gallery/>
- <https://codecanyon.net/item/jquery-carousel-evolution/490018>

jQuery image sliders

- <http://www.pixedelic.com/plugins/camera/>
- <https://codecanyon.net/item/aviaslider-jquery-slideshow/105090>
- <http://www.jssor.com/>

Practice Tasks

- Read and utilize the JQuery traversing by creating your own mini code snippets for all of the following:
jQuery Traversing
https://www.w3schools.com/jquery/jquery_traversing.asp
(upgrade all samples of code to jquery-3.6.0)
- jQuery Traversing – **Ancestors**
https://www.w3schools.com/jquery/jquery_traversing_ancestors.asp
- jQuery Traversing – **Descendants**
[https://www.w3schools.com/jquery/jquery_traversing_descendants.a
sp](https://www.w3schools.com/jquery/jquery_traversing_descendants.asp)
- jQuery Traversing – **Siblings**
https://www.w3schools.com/jquery/jquery_traversing_siblings.asp
- jQuery Traversing - **Filtering**
https://www.w3schools.com/jquery/jquery_traversing_filtering.asp