AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology help us to load data from t he server without a browser page refresh.

If you are new with AJAX, I would recommend you go through our [Ajax Tutorial](http://www.tutorialspoint.com/ajax/index.htm) before proceeding further.

JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application.

## 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**

Here is the simple syntax for **load()** method −

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

Here is the description of all the parameters −

* **URL** − The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script which generates data dynamically or out of a database.
* **data** − This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the **POST** method. If omitted, the **GET** method is used.
* **callback** − A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

### **Example**

Consider the following HTML file with a small JQuery coding −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

$('#stage').load('/jquery/result.html');

});

});

</script>

</head>

<body>

<p>Click on the button to load /jquery/result.html file −</p>

<divid="stage"style="background-color:cc0;">

STAGE

</div>

<inputtype="button"id="driver"value="Load Data"/>

</body>

</html>

Here **load()** initiates an Ajax request to the specified URL **/jquery/result.html**file. After loading this file, all the content would be populated inside <div> tagged with ID *stage*. Assuming, our /jquery/result.html file has just one HTML line −

<h1>THIS IS RESULT...</h1>

When you click the given button, then result.html file gets loaded.

## 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**

Here is the simple syntax for **getJSON()** method −

**[selector].**getJSON( URL, [data], [callback] );

Here is the description of all the parameters −

* **URL** − The URL of the server-side resource contacted via the GET method.
* **data** − An object whose properties serve as the name/value pairs used to construct a query string to be appended to the URL, or a preformatted and encoded query string.
* **callback** − A function invoked when the request completes. The data value resulting from digesting the response body as a JSON string is passed as the first parameter to this callback, and the status as the second.

### **Example**

Consider the following HTML file with a small JQuery coding −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

$.getJSON('/jquery/result.json',function(jd){

$('#stage').html('<p> Name: '+ jd.name +'</p>');

$('#stage').append('<p>Age : '+jd.age+'</p>');

$('#stage').append('<p> Sex: '+jd.sex+'</p>');

});

});

});

</script>

</head>

<body>

<p>Click on the button to load result.json file −</p>

<divid="stage"style="background-color:#eee;">

STAGE

</div>

<inputtype="button"id="driver"value="Load Data"/>

</body>

</html>

Here JQuery utility method **getJSON()** initiates an Ajax request to the specified URL **result.json** file. After loading this file, all the content would be passed to the callback function which finally would be populated inside <div> tagged with ID *stage*. Assuming, our result.json file has following json formatted content −

{

"name": "Zara Ali",

"age" : "67",

"sex": "female"

}

When you click the given button, then result.json file gets loaded.

## 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.

### **Example**

This example demonstrate how can pass user input to a web server script which would send the same result back and we would print it −

<html>

<head>

<title>The jQuery Example</title>

<scripttype="text/javascript"

src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>

<scripttype="text/javascript"language="javascript">

$(document).ready(function(){

$("#driver").click(function(event){

var name = $("#name").val();

$("#stage").load('/jquery/result.php',{"name":name});

});

});

</script>

</head>

<body>

<p>Enter your name and click on the button:</p>

<inputtype="input"id="name"size="40"/><br/>

<divid="stage"style="background-color:cc0;">

STAGE

</div>

<inputtype="button"id="driver"value="Show Result"/>

</body>

</html>

Here is the code written in **result.php** script −

<?php

if( $\_REQUEST["name"]){

$name = $\_REQUEST['name'];

echo"Welcome ". $name;

}

?>

Now you can enter any text in the given input box and then click "Show Result" button to see what you have entered in the input box.

## JQuery AJAX Methods

You have seen basic concept of AJAX using JQuery. Following table lists down all important JQuery AJAX methods which you can use based your programming need −

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**jQuery.ajax( options )**](http://www.tutorialspoint.com/jquery/ajax-jquery-ajax.htm)  Load a remote page using an HTTP request.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#driver").click(function(event){  $.ajax({  url:'result.html',  success:function(data){  $('#stage').html(data);  }  });  });  });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:blue;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 2 | [**jQuery.ajaxSetup( options )**](http://www.tutorialspoint.com/jquery/ajax-jquery-ajaxsetup.htm)  Setup global settings for AJAX requests.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  // Do global setting.  $.ajaxSetup({  url:"result.html"  });    $.ajax({  success:function(data){  $('#stage').html(data);  }  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:#cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 3 | [**jQuery.get( url, [data], [callback], [type] )**](http://www.tutorialspoint.com/jquery/ajax-jquery-get.htm)  Load a remote page using an HTTP GET request.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.get(  "result.php",  { name:"Zara"},  function(data){  $('#stage').html(data);  }  );  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file −</p>    <spanid="stage"style="background-color:#cc0;">  STAGE  </span>    <div><inputtype="button"id="driver"  value="Load Data"/></div>    </body>    </html> |
| 4 | [**jQuery.getJSON( url, [data], [callback] )**](http://www.tutorialspoint.com/jquery/ajax-jquery-getjson.htm)  Load JSON data using an HTTP GET request.  {  "name":"Zara Ali",  "age":"67",  "sex":"female"  }  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.getJSON('result.json',function(jd){  $('#stage').html('<p> Name: '+ jd.name +'</p>');  $('#stage').append('<p>Age : '+jd.age+'</p>');  $('#stage').append('<p> Sex: '+jd.sex+'</p>');  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:#cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 5 | [**jQuery.getScript( url, [callback] )**](http://www.tutorialspoint.com/jquery/ajax-jquery-getscript.htm)  Loads and executes a JavaScript file using an HTTP GET request.  functionCheckJS(){  alert("This is JavaScript");  }  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $.getScript('result.js',function(jd){  // Call custom function defined in script  CheckJS();  });  });    });  </script>  </head>    <body>    <p>Click on the button to load result.js file −</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 6 | [**jQuery.post( url, [data], [callback], [type] )**](http://www.tutorialspoint.com/jquery/ajax-jquery-post.htm)  Load a remote page using an HTTP POST request.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "result.php",  { name:"Zara"},  function(data){  $('#stage').html(data);  }  );    });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file −</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 7 | [**load( url, [data], [callback] )**](http://www.tutorialspoint.com/jquery/ajax-load.htm)  Load HTML from a remote file and inject it into the DOM.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  $("#driver").click(function(event){  $('#stage').load('result.html');  });  });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage"style="background-color:cc0;">  STAGE  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 8 | [**serialize( )**](http://www.tutorialspoint.com/jquery/ajax-serialize.htm)  Serializes a set of input elements into a string of data.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  $age = $\_REQUEST['age'];  echo"<br />Your age : ". $age;  $sex = $\_REQUEST['sex'];  echo"<br />Your gender : ". $sex;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "/jquery/serialize.php",  $("#testform").serialize(),    function(data){  $('#stage1').html(data);  }    );    varstr= $("#testform").serialize();  $("#stage2").text(str);  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>  <br/>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <formid="testform">  <table>    <tr>  <td><p>Name:</p></td>  <td><inputtype="text"name="name"size="40"/></td>  </tr>    <tr>  <td><p>Age:</p></td>  <td><inputtype="text"name="age"size="40"/></td>  </tr>    <tr>    <td><p>Sex:</p></td>    <td><selectname="sex">  <optionvalue="Male"selected>Male</option>  <optionvalue="Female"selected>Female</option>  </select></td>    </tr>    <tr>  <tdcolspan="2">  <inputtype="button"id="driver"value="Load Data"/>  </td>  </tr>    </table>    </form>    </body>    </html> |
| 9 | [**serializeArray( )**](http://www.tutorialspoint.com/jquery/ajax-serializearray.htm)  Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.  <?php  if( $\_REQUEST["name"]){  $name = $\_REQUEST['name'];  echo"Welcome ". $name;  $age = $\_REQUEST['age'];  echo"<br />Your age : ". $age;  $sex = $\_REQUEST['sex'];  echo"<br />Your gender : ". $sex;  }  ?>  Following is a simple example a simple showing the usage of this method −  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){    $.post(  "/jquery/serialize.php",  $("#testform").serializeArray(),  function(data){  $('#stage1').html(data);  }  );    var fields = $("#testform").serializeArray();  $("#stage2").empty();    jQuery.each(fields,function(i, field){  $("#stage2").append(field.value+" ");  });    });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <br/>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <formid="testform">    <table>    <tr>  <td><p>Name:</p></td>  <td><inputtype="text"name="name"size="40"/></td>  </tr>    <tr>  <td><p>Age:</p></td>  <td><inputtype="text"name="age"size="40"/></td>  </tr>    <tr>  <td><p>Sex:</p></td>  <td><selectname="sex">  <optionvalue="Male"selected>Male</option>  <optionvalue="Female"selected>Female</option>  </select></td>  </tr>    <tr>  <tdcolspan="2">  <inputtype="button"id="driver"value="Load Data"/>  </td>  </tr>    </table>    </form>    </body>    </html> |

## JQuery AJAX Events

You can call various JQuery methods during the life cycle of AJAX call progress. Based on different events/stages following methods are available −

You can go through all the [AJAX Events](http://www.tutorialspoint.com/jquery/ajax-events.htm).

|  |  |
| --- | --- |
| **S.N.** | **Methods & Description** |
| 1 | [**ajaxComplete( callback )**](http://www.tutorialspoint.com/jquery/ajaxcomplete.htm)  Attach a function to be executed whenever an AJAX request completes.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  $('#stage1').load('result.html');  });  $(document).ajaxComplete(function(event, request, settings){  $("#stage2").html("<h1>Request Complete.</h1>");  });    });  </script>  </head>  <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 2 | [**ajaxStart( callback )**](http://www.tutorialspoint.com/jquery/ajaxstart.htm)  Attach a function to be executed whenever an AJAX request begins and there is none already active.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){  /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage1').load('result.html');  });    /\* Gets called when request starts \*/    $(document).ajaxStart(function(){  count++;  $("#stage2").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,set){  count++;  $("#stage3").html("<h1>Completes,Count:"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 3 | [**ajaxError( callback )**](http://www.tutorialspoint.com/jquery/ajaxerror.htm)  Attach a function to be executed whenever an AJAX request fails.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    $("#driver").click(function(event){  /\* Assume result.text does not exist. \*/  $('#stage1').load('/jquery/result.text');  });  $(document).ajaxError(function(event, request, settings ){  $("#stage2").html("<h1>Error in loading page.</h1>");  });    });  </script>  </head>  <body>    <p>Click on the button to load result.text file:</p>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 4 | [**ajaxSend( callback )**](http://www.tutorialspoint.com/jquery/ajaxsend.htm)  Attach a function to be executed before an AJAX request is sent.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;    $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });  /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt,req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes, Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 5 | [**ajaxStop( callback )**](http://www.tutorialspoint.com/jquery/ajaxstop.htm)  Attach a function to be executed whenever all AJAX requests have ended.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt,req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request complete \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes, Count :"+ count +"</h1>");  });    /\* Gets called when all requests are ended \*/  $(document).ajaxStop(function(event,request,settings){  count++;  $("#stage4").html("<h1>Stops, Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <divid="stage4"style="background-color:blue;">  STAGE - 4  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
| 6 | [**ajaxSuccess( callback )**](http://www.tutorialspoint.com/jquery/ajaxsuccess.htm)  Attach a function to be executed whenever an AJAX request completes successfully.  <h1>THIS IS RESULT...</h1>  Following is a simple example a simple showing the usage of this method.  <html>  <head>  <title>The jQuery Example</title>  <scripttype="text/javascript"  src="http://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js"></script>    <scripttype="text/javascript"language="javascript">  $(document).ready(function(){    /\* Global variable \*/  var count =0;  $("#driver").click(function(event){  $('#stage0').load('result.html');  });    /\* Gets called when request starts \*/  $(document).ajaxStart(function(){  count++;  $("#stage1").html("<h1>Starts, Count :"+ count +"</h1>");  });    /\* Gets called when request is sent \*/  $(document).ajaxSend(function(evt,req,set){  count++;  $("#stage2").html("<h1>Sends, Count :"+ count +"</h1>");  $("#stage2").append("<h1>URL :"+set.url +"</h1>");  });    /\* Gets called when request completes \*/  $(document).ajaxComplete(function(event,request,settings){  count++;  $("#stage3").html("<h1>Completes,Count:"+ count +"</h1>");  });    /\* Gets called when request is stopped \*/  $(document).ajaxStop(function(event,request,settings){  count++;  $("#stage4").html("<h1>Stops, Count :"+ count +"</h1>");  });    /\* Gets called when all request completes successfully \*/  $(document).ajaxSuccess(function(event,request,settings){  count++;  $("#stage5").html("<h1>Success,Count :"+ count +"</h1>");  });    });  </script>  </head>    <body>    <p>Click on the button to load result.html file:</p>    <divid="stage0"style="background-color:blue;">  STAGE - 0  </div>    <divid="stage1"style="background-color:blue;">  STAGE - 1  </div>    <divid="stage2"style="background-color:blue;">  STAGE - 2  </div>    <divid="stage3"style="background-color:blue;">  STAGE - 3  </div>    <divid="stage4"style="background-color:blue;">  STAGE - 4  </div>    <divid="stage5"style="background-color:blue;">  STAGE - 5  </div>    <inputtype="button"id="driver"value="Load Data"/>    </body>    </html> |
|  |  |