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

## jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing 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".

## The Document Ready Event

$(document).ready(function(){  
 $("button").click(function(){  
    $("p").hide();  
   });  
});

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

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

## The element Selector $("p")

## The #id Selector $("#test")

## The .class Selector$(".test")

Use the correct selector to hide all elements with an href attribute.

$("[href]").hide();

Use the correct selector to hide all odd table rows in a table.

$("tr:odd").hide();

## jQuery Syntax For Event Methods

In jQuery, most DOM events have an equivalent jQuery method.

To assign a click event to all paragraphs on a page, you can do this:

$("p").click();

The next step is to define what should happen when the event fires. You must pass a function to the event:

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

## The on() Method

The on() method attaches one or more event handlers for the selected elements.

## $("p").on({   mouseenter: function(){     $(this).css("background-color", "lightgray");   },   mouseleave: function(){     $(this).css("background-color", "lightblue");   },   click: function(){     $(this).css("background-color", "yellow");   } });

## 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*);**

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

## jQuery Method Chaining

Until now we have been writing jQuery statements one at a time (one after the other).

However, there is a technique called chaining, that allows us to run multiple jQuery commands, one after the other, on the same element(s).

$("#p1").css("color", "red")  
  .slideUp(2000)  
  .slideDown(2000);

## jQuery DOM Manipulation

One very important part of jQuery is the possibility to manipulate the DOM.

jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.

## Get Content

* text() - Sets or returns the text content of selected elements
* html() - Sets or returns the content of selected elements (including HTML markup)
* val() - Sets or returns the value of form fields

$("button").click(function(){

alert("Value: " + $("#test").val());

});

## Set Content

We will use the same three methods from the previous page to **set content**:

$("#btn1").click(function(){  
  $("#test1").text("Hello world!");  
});

## Set Attributes - attr()

The jQuery attr() method is also used to set/change attribute values.

$("button").click(function(){  
  $("#w3s").attr("href", "https://www.w3schools.com/jquery/");  
});

**Add New HTML Content**

We will look at four jQuery methods that are used to add new content:

* append() - Inserts content at the end of the selected elements
* prepend() - Inserts content at the beginning of the selected elements
* after() - Inserts content after the selected elements
* before() - Inserts content before the selected elements

**Remove Elements/Content**

To remove elements and content, there are mainly two jQuery methods:

* remove() - Removes the selected element (and its child elements) $("#div1").remove();
* empty() - Removes the child elements from the selected element $("#div1").empty(); $("p").remove(".test");

**Traversing Up the DOM Tree**

Three useful jQuery methods for traversing up the DOM tree are:

* parent()
* parents()
* parentsUntil()

$(document).ready(function(){  
  $("span").parentsUntil("div");  
});

**Traversing Down the DOM Tree**

Two useful jQuery methods for traversing down the DOM tree are:

* children()
* find()

$(document).ready(function(){  
  $("div").find("span");  
});

## jQuery and AJAX

## jQuery load() Method - The load() method loads data from a server and puts the returned data into the selected element.

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

The required URL parameter specifies the URL you wish to load.

The optional data parameter specifies a set of querystring key/value pairs to send along with the request.

The optional callback parameter is the name of a function to be executed after the load() method is completed.

$("button").click(function(){  
  $("#div1").load("demo\_test.txt", function(responseTxt, statusTxt, xhr){  
    if(statusTxt == "success")  
      alert("External content loaded successfully!");  
    if(statusTxt == "error")  
      alert("Error: " + xhr.status + ": " + xhr.statusText);  
  });  
});

* responseTxt - contains the resulting content if the call succeeds
* statusTxt - contains the status of the call
* xhr - contains the XMLHttpRequest object

**HTTP Request: GET vs. POST**

Two commonly used methods for a request-response between a client and server are: GET and POST.

* **GET** - Requests data from a specified resource
* **POST** - Submits data to be processed to a specified resource

$.get(*URL,callback*);

$("button").click(function(){  
  $.get("demo\_test.asp", function(data, status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
});

$.post(*URL,data,callback*);

$("button").click(function(){  
  $.post("demo\_test\_post.asp",  
  {  
    name: "Donald Duck",  
    city: "Duckburg"  
  },  
  function(data, status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
});

$(".btnDownVote").click(function ()

{

$.ajax({

**url**: "/api/questions?answerid=" + UpBtn.attr("data-answerid") + "&value=" + value + "&UserID=@Session["CurrentUserID"]", **type**: "POST", **success**: function ()

{

DownBtn.parent().find(".votescount").text(parseInt(DownBtn.parent().find(".votescount").text()) + value);

DownBtn.attr("src", "/Images/down-color.png");

UpBtn.attr("src", "/Images/up-empty.png");

}

});

});

var sendInfo = {

Name: name,

Address: address,

Phone: phone

};

$.ajax({

**type**: "POST",

**url**: "/Home/Add",

**dataType**: "json",

**data**: sendInfo,

**success**: function (msg) {

if (msg) {

alert("Somebody" + name + " was added in list !");

location.reload(true);

} else {

alert("Cannot add to list !");

}

}

});

What is the difference between **JavaScript**and **jQuery**?

JavaScript is an independent language and can exist on its own. JQuery is a JavaScript library.

What is the difference between **$(window).load() vs $(document).ready()** function in jQuery? When to use $(window).load instead of $(document).ready()?

$(window).load()

The code which gets included inside $( window ).on( "load", function() { ... }) runs only once the entire page is ready (not only DOM).

$(document).ready()

The ready() method is used to make a function available after the document is loaded. Whatever code you write inside the $( document ).ready() method will run once the page DOM is ready to execute JavaScript code.

What is the difference between **prop**and **attr**in jQuery?

Both attr() and prop() are used to get or set the value of the specified property of an element attribute, but attr() returns the default value (Original state) of a property whereas **prop() returns the current value** (Current state).

What is the use of the **delay()** method in jQuery?

The jQuery delay() method is used **to delay the execution of functions in the queue**.

$("#div1").delay("slow").fadeIn();

Can you use multiple **document.ready()**function on the same page?

You can have multiple ones, but it's not always the neatest thing to do

What is**jQuery.holdReady()**function in jQuery?

holdReady() method **allows the caller to delay jQuery's ready event**.

What is the difference **.empty()**, **.remove()**, and **.detach()** in jQuery?

When you click on “**remove()**” button it will remove the yellow box with all content and elements inside the yellow box.

When you click on “**empty()**” button it will remove all content and child elements inside the yellow box only empty yellow box will display.

When you click on “**detach()**” button it will remove all content and child elements inside the yellow box and yellow box too but it keeps all data and event handlers of the removed elements which we can reuse it at a later time.

What is the difference between **$(this)** and "**this**" in jQuery?

this is the DOM object, whereas $(this) is the jQuery wrapper around same.

When using this, you can call DOM methods on it, but not jQuery methods. When using $(this), you can call jQuery methods on it, but not DOM methods.

What is the difference between **Map vs Grep** functions in jQuery?

The difference is that **the grep() method filters an array and returns the filtered array**, while map() method applies a function to each item in the array, thus returning a modified array.

Append to list and para.

$("#btn1").click(function(){

$("p").append(" <b>Appended text</b>.");

});

$("#btn2").click(function(){

$("ol").append("<li>Appended item</li>");

});

### [Access Session Values in JavaScript (Client Side)](https://www.aspdotnet-suresh.com/2017/02/aspnet-get-access-session-values-in-javascript-jquery-client-side.html)

Session["UserName"] = "Guest"; In C# code

In Client

<script type="text/javascript">

$(function() {

var name = 'Welcome '+' <%=Session["UserName"] %>'

$('#lbltxt').text(name)

});

</script>

### [Get All Images (img) Src in Div with Example](https://www.aspdotnet-suresh.com/2016/12/jquery-get-all-images-img-src-with-example.html)

<script type="text/javascript">

$(function () {

$('#btnGet').click(function () {

var imgs = $('#divimages').children('img').map(function () {

return $(this).attr('src')

}).get()

alert(imgs);

})

})

</script>

</head>

<body>

<b> Get All the Images in Div </b>

<div id="divimages">

<img src="test.png" />

<img src="test2.jpg" />

</div>

### [Asp.Net MVC Redirect to Another View or Controller Action Method from View](https://www.aspdotnet-suresh.com/2016/10/aspnet-mvc-redirect-to-another-view-controller-action-method-from-view.html)

<script type="text/javascript">

$(function () {

$("#btnRedirect").click(function(){

window.location.href = "@Url.Action("Action Name", "Controller Name")";

});

});

</script>

### [jQuery Check if Div Element has Class or Not Example](https://www.aspdotnet-suresh.com/2016/07/jquery-check-if-div-element-has-class-not-example.html)

<script type="text/javascript">

$(function(){

$('#divdetails').addClass("purplcls");

$('#divdetails').click(function () {

if($(this).hasClass("purplcls")){

$(this).removeClass("purplcls");

$(this).addClass("blucls");

}

else{

$(this).removeClass("blucls");

$(this).addClass("purplcls");

}

})

})

</script>

### [jQuery Show Bootstrap Modal Popup on Button Click with Example](https://www.aspdotnet-suresh.com/2016/07/jquery-show-bootstrap-modal-popup-on-button-click-with-example.html)

Need Bootstrap and jquery for this CDN links

<script type="text/javascript">

$(function () {

$("#btnShow").click(function(){

$('#demoModal').modal('show');

});

});

</script>

</head>

<body>

<!--Button to Trigger Modal-->

<div style="text-align:center; margin-top:10%">

<button id="btnShow" class="btn btn-primary btn-lg">Show Modal Popup</button>

</div>

<!-- Modal -->

<div class="modal fade" id="demoModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

<h4 class="modal-title" id="myModalLabel">Bootstrap Modal Popup</h4>

</div>

<div class="modal-body">Hi, Welcome to Aspdotnet-Suresh.com</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

<button type="button" class="btn btn-primary">Save changes</button>

</div>

</div>

</div>

</div>

</body>

</html>

**◆**$("h1").text('Butterfly');

**◆**$("p").html('This is a <strong>butterfly</strong> in its natural habitat:');

**◆**$("img").attr("src",'https://www.kasandbox.org/programming-images/animals/butterfly\_monarch.png')

**◆**$("h1").css("color","orange")

**◆**var spiders = $("<p>");

spiders.text("new animal with shark and spider");

$("body").append(spiders);

spiders.addClass("result");

**◆**var $pic = $("#picture"); The $pic variable below stores a jQuery collection object. Denoted by $

var pic = $pic[0]; Get DOM element

**◆**var $heading = $(heading);

Convert into jQuery collection var heading = document.getElementById('title');

**◆**Using either a for loop or the each() method, find each paragraph, send its text through the toPigLatin function, and set the paragraph to that text.

var $paragraphs = $("p"); //Create Jquery collection object

$paragraphs.each(function(index, element) {

var $paragraph = $(element); //Single jQuery object of each paragraph

$paragraph.text(toPigLatin($paragraph.text())); //set text by passing thorugh the PigLatin func

});

**◆Mark spot/dot on image on clicking**

$("#dog-pic").on("click", function(event) {

var $dot = $("<div></div>");

$dot.addClass("dot");

$dot.appendTo("body")

.css("position", "absolute")

.css("top", event.pageY + "px")

.css("left", event.pageX + "px");

});

**◆Explain bind() vs live() vs delegate() methods in Jquery**

**The bind()** method will not attach events to those elements which are added after DOM is loaded while **live()** and **delegate()** methods attach events to the future elements also.

$(document).ready(function(){

$("#myTable").find("tr").live("click",function(){

alert($(this).text());

});

});

**What Do We Use The Delay() Method In JQuery?**

The delay() method is mainly used to delay the execution of functions in a queue. It is mainly used to make a delay between the queued jQuery effects.

$(selector).delay (speed, queueName)

**Write A JQuery Code Selector That Is Used For Querying All The Elements Whose ID Ends With The String “IB”?**

$(“[id$=’IB’]”)