**Important Links:** jQuery **:** <https://jquery.com/>

jQuery CDN : <http://code.jquery.com/>

jQuery Events : <https://api.jquery.com/category/events/>

jQuery Events.data : <https://api.jquery.com/event.data/>

Public APIs: <https://github.com/public-apis/public-apis>

Jss/css selectors : <https://www.w3schools.com/jquery/jquery_ref_selectors.asp>

===================Step to follow when using JQuery in project====================

**Step1**. Include the minified version of this in the index.html from jQuery CDN site

|  |  |
| --- | --- |
|  | Copy the whole content from the minified link and use in Index.html <head>  <script   src="http://code.jquery.com/jquery-2.2.4.min.js"   integrity="sha256-BbhdlvQf/xTY9gja0Dq3HiwQF8LaCRTXxZKRutelT44="   crossorigin="anonymous"></script> |

**Step2**. Create the script.js file for the project and include

jQuery(function(){ … }) or $(function(){ … }) in the .js file

**Step3**. Add the reference of .js file in the index.html file in <head>

<script src="js/script.js"></script>

==============================effects ================================

**$(".red-box").fadeIn(); $(".green-box").fadeOut();**

**$(".blue-box").fadeTo(1000,0.8); $(".blue-box").fadeToggle();**

**//custom animation**

**$(".red-box").animate({**

**"margin-left": "+=20px",**

**},1000, "linear")**

**Or**

**$(".red-box").animate({marginLeft:"20px",opacity:"0.2"})**

**//custom animation with defined delay**

**$(".red-box").fadeTo(1000, 0.2);**

**$(".green-box").delay(1000).fadeTo(1000, 0.2);**

**$(".red-box").fadeTo(1000, 1).fadeTo(1000, 0.2);**

**//call back function**

**$(".red-box").fadeTo(1000, 0.2, function(){**

**alert("This is test");**

**});**

**========== Traverse =====================================**

$("#li-id").find("li").css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").children("li").css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").parents().css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").parents("div").css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").parent().css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").siblings().css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").siblings("p").css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").siblings(":header").css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").prev().css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").next().css("background-color", "rgba(180,180,30,0.8)");

$("#li-id").prev().next().css("background-color", "rgba(180,180,30,0.8)");

**================Filter=====================================**

$("#list").find("li").filter(":even").css("background-color", "rgb(180,180,30,0.8)");

$("#list").children("li").filter(":even").css("background-color", "rgb(180,180,30,0.8)");

// filter with callback:

$("li").filter(function(index){

return index%3 == 2;}).css("background-color", "rgba(180,180,30,0.8)");

// first() & last()

$("li").first().css("background-color", "rgb(180,180,30,0.8)");

$("li").last().css("background-color", "rgb(180,180,30,0.8)");

// positional : eq(index)

$("li").eq(4).css("background-color", "rgb(180,180,30,0.8)");

$("li").eq(-2).css("background-color", "rgb(180,180,30,0.8)");

// not()

$("li").not(":first").css("background-color", "rgb(180,180,30,0.8)");

$("li").not("#list ul li").css("background-color", "rgb(180,180,30,0.8)");

$("li").not(function(index){

return index%3 == 2;}).css("background-color", "rgba(180,180,30,0.8)");

**========== Add new element to Dom==============================**

=====**append**() & **prepend() ===**=== adds element as new **child** element

$("ul ul:first").append("<li>i`m the last sub item</li>");

Or

$("<li>i`m the last sub item</li>").appendTo($("ul ul:first"));

=========================

$("ul ul:first").prepend("<li>i`m the first sub item</li>");

Or

$("<li>i`m the last sub item</li>").prependTo($("ul ul:first"));

Or

$("<li>i`m the last sub item</li>").prependTo("ul ul:first");

=====**after**() & **before**()=== adds element as new **sibling** element

$(".red-box").after("<div> class='red-box' > Another Red Box </div>");

Or

$(".green-box").before("<div> class='green-box' > Another green Box </div>");

===insertAfter() & insertBefore()=============

=== callback function can also be passed to all these methods==

$(".green-box").before(function(){

return "<div> class='red-box' > Another Red Box </div>";

});

== an element can also be passed direct from the Dom

$(".blue-box").after($(".red-box"));

**========== replace element/s to Dom============================**

=====**replaceWith() ===**===

**//can pass new element**

$("li").replaceWith("<li>replaced</li>");

**//can pass the string using function**

$("li").replaceWith(function(){

Return "<li>replaced</li>";

});

**// can pass the item**

var varItem = $("li:first")

$("p").replaceWith(varItem)

====replaceAll()=== opposite of SourceElement.replacewith(targetElement)

$("<div class='green-box'>More Green</div>").replaceAll(".red-box, .blue-box");

**======== remove,empty element/s from Dom==================**

**// remove() , detach() deletes the element and all its children**

**=====remove() ======**

$("li").remove();

// removes all the children of form which are not text type input, textarea and break line

$(“form”).children().not(“input:text, textarea, br”).remove();

**=====detach()======**

**//remembers the state, associated data and event handler, so when appended the same element, it restores the previous state**

var VarItem = $(“li”).detach();

$(“#content”).append(VarItem);

**=====empty()======**

**// empty() does not delete the element itself, but deletes all children and contents from the element**

**======== manipulate attributes and properties ==============**

**// all these works with first found element on the page///**

**=====attr() ======**

$("#spcl-link").attr(“href”); //fetches the existing value

$("#spcl-link").attr(“href”, “https://///,.....”); //updates the existing value with this new

**// attr() does not work with boolean properties or attributes.**

**ex : checked/selected/disabled**

**// prop() shoub be used in these case**

$("input:checkbox ").prop(“checked”);

**===== val() ====== used with input elements**

$("input:text").val();

$("input[type=’range’]").val(); //fetches the existing value

$("input:text").val(“new value”); //updates the existing value with this new

**======== manipulate css properties ==============**

**.hide() : sets the display property value as none for the element**

**=====css() ======**

$(".gallery").css(“display”); // returns the existing value

$(".gallery").css(“display”, “none”); // equivalent to .hide() : updates the display property as none

$(".red-box").css(“width”); // returns the width with units : 80px

$(".red-box").width(); // returns only the width without unit : 80

$(".red-box").css(“height”); // returns the height with units : 80px

$(".red-box").widheightth(); // returns only the height without unit : 80

$(".red-box").css(“background-color”, “#AA7700”); // sets the background color

$("p").css(“font-size”, “20px”); // sets the font size

$(".red-box").css(“width”, “+=20px”); // sets the additional 20px height

$(“p”).css([“color”,”line-height”,”font-size”]) // returns all properties in java script object format

$(".red-box").css(“user-select”, “none”); // disables user`s ability to select the item

//can also be used with call back function

$(".red-box").css(“user-select”, function(){

Return “none”; });

css() // it does not handle the "! Important" marks in css, for this css class selectors are needed

**======== manipulate css class ==============**

**=====addClass() ======**

$(“a”).addClass(“fancy-link”) // adds the class to the element

$(“a”).addClass(“class1 class2 class3”) // adds multiple classes to the element . {separated by space}

$(“li li”).addClass(function(index){

$(this).addClass(“item-”+ index)

});

$(“div”).addClass(function(index, currentClass){

If(currentClass == “dummy”) { return “red-box”; }

});

**=====removeClass() ======**

$(“.red-box”).removeClass(“.red-class”);

===ex: change class

$(“.red-box”).removeClass(“.red-class”).addClass(“green-class”);

**======== changing data of an element ==============**

**=====data() ======data(“somekey”,”data”)=====**

var gallery = $(“.gallery”);

var images = [“image1”, “image2”, “image3”];

gallery.data(“availableImages”, images); //assigns images array as data to .gallery element

gallery.data(“name”, “The gallery”); //assigns images array as data to .gallery element

gallery.data(“availableImages”) //fetches the value of provided key

gallery.data() //fetches the value of all data points/keys

**=====removeData(“key”) ======**

gallery.removeData(“name”) //removes the data for the provided key

**======== retrieving and changing data of an element ==========**

**// text() : ignores the html tags in the content**

**// html() : considers the html tags in the content**

$(“p:first”).text(); // fetches the value between the opening and closing tags// This is bold item.

$(“p:first”).html(); //fetches the value between the opening and closing tags //This is <b>bold</b> item.

$(“p:first”).text(“<strong>Hello</strong>”); //adds text as content// <strong>Hello</strong>

$(“p:first”).html((“<strong>Hello</strong>”); // adds html as content //**Hello**

**Ex:**

var firstPar = $(“p:first”);

firstPar.html(firstPar.html() + “This is appended”);

**================ event handlers ========================**

`

========click event==== **click()**

$(".btn-click").click(function(){

  alert("Button was clicked")

});

$(".btn-click").click(function(event){

  console.log(event); //returns java script object

  alert("Button was clicked")

});

========hover event====

**hover(),** its same like Click event handler, but it fires 2 time in Jquery , when mouse enters and when leaves unlike in css (:hover) it fires only once . Can be solved using mouseenter()/mouseleaves()

==== mouse enter and mouse leave event========

**mouseenter() , mouseleave()**

var bluebox = $(“blue-box”);

|  |  |
| --- | --- |
| bluebox.mouseenter(function(){  $(this).fadeTo(500. 0.5);  }); | bluebox.mouseleave(function(){  $(this).fadeTo(500. 1);  }); |

This has problem of animation queue, can be solved using stop() method

|  |  |
| --- | --- |
| bluebox.mouseenter(function(){  $(this).stop().fadeTo(500. 0.5);  }); | bluebox.mouseleave(function(){  $(this).stop().fadeTo(500. 1);  }); |

Or Shorthand for hover event with mouseenter and mouseleave events

**==hover(handlerIn, handlerOut)**

Bluebox.hover(function(){

$(this).stop().fadeTo(500, 0.5)}, function(){

$(this).stop().fadeTo(500, 1)

});

========same handler for multiple event====

**on(“event”, function()) // events separated by space**

$(“html”).on(“click hover keydown”, function(){

Console.log(“same action for multiple events”)

});

========modularizing event handlers ====

function logEvent() {

console.log(“mouse was clicked or keydown pressed”);

}

$(“html”).on(“click hover keydown”, logEvent);

==================Delegated events ==========

$(“p”).click(function(){

$(this).slideUp();

});

$(“p”).append(“<p>This is newly added</p>”);

// delegated events

$(“#content”).on(“click”, “p”, function(){

$(this).slideUp();

});

$(“p”).append(“<p>This is newly added</p>”);

==================passing additional data to events ==========

$(“#btn-click”).click({

User: “user\_name”,

Domain: “abc.com”

}, function(event){

Greetuser(event.data);

});

function greetuser(userdata){

username = userdata.User;

useremail = userdata.domain;

alert(“Welcome back” + username + “from domain ” + userdomain + “!”);

}

==================keyboard handling: keyUp and KeyDown events ==========

**======keypress() -- evil,**

**===which**

$(“html”).keydown(function(event){

Console.log(event.which);

});

Var arrow\_right = 39;

$(“html”).keydown(function(event){

If(event.which == arrow\_right) {

$(“.blue-box”).stop().animate({

“margin-left” : “+=20px”

}, 50);

};

});

**================ form event handlers ========================**

**======focus() and blur() event**

Var inputfields = $(“input:text, input:password, textarea”)

Inputfields.focus(function(){

$(this).css(“box-shadow”, “0 0 4px #666”);

});

Inputfields.blur(function(){

$(this).css(“box-shadow”, “none”);

});

**======change() event : for check box, radio box, selectors**

===example for check box

$("#checkbox").change(function() {

var isChecked = $(this).is(":checked"); // or .prop("checked")

if (isChecked) {

// Add the checkbox' label to the selection and highlight both in green.

$(this).add("label[for='checkbox']").css("box-shadow", "0 0 4px #181");

} else {

$(this).add("label[for='checkbox']").css("box-shadow", "0 0 4px #811");

}

});

=======example for drop down

$("#selection").change(function(){

var selectedoption= $(this).find(":selected").text();

alert(selectedoption);

});

**====== handling form submit events=============**

**//submit() , trim(), event.preventDefault()**

$("#form").submit(function(event) {

var textarea = $("#message");

if (textarea.val().trim() === "") {

textarea.css("box-shadow", "0 0 4px #811");

event.preventDefault();

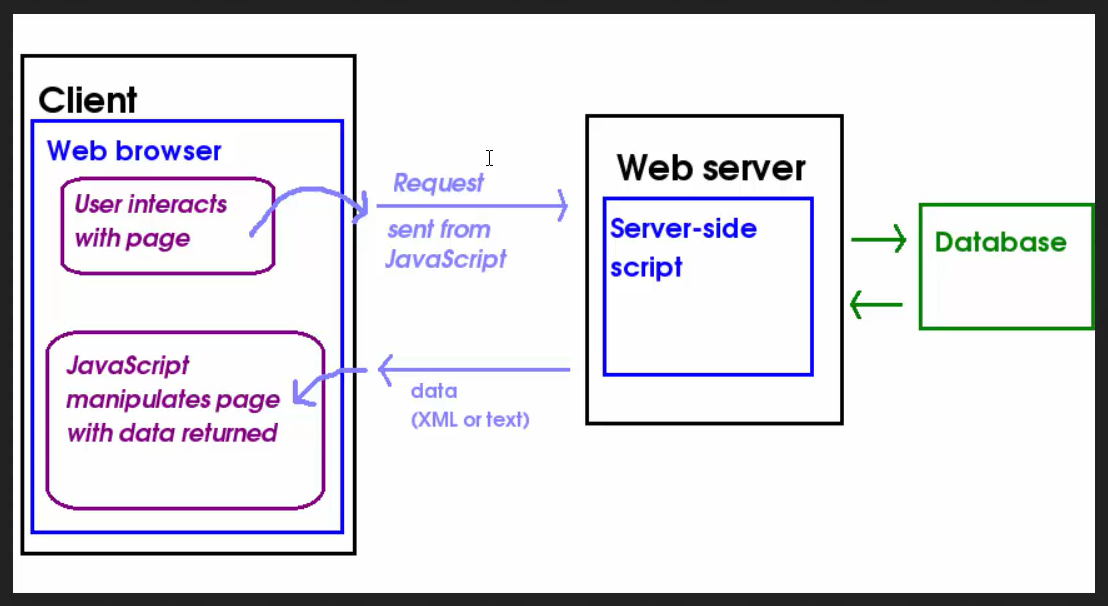
} else {

// Nothing to do, let the form be submitted.

}

});

**================ Ajax with jQuery ========================**



Asynchronous java script and xml: Page continues loading and does not wait for the API /AJAX call

// $.load(), $.get(), $.post(), $.ajax(), $.getJSON()

**>> load() from the same server**

|  |
| --- |
| $(function () { |
|  |  |
|  | // You can load files from the server (or local file system in this case) with |
|  | // jQuery using its load() function. |
|  | $("#code").load("js/script.js"); |
|  |  |
|  | // You can also handle different responses, for instance if an error occurred. |
|  | $("#code").load("idontexist.php", function(response, status) { |
|  | console.log(response); // empty (normally contains the fetched contents) |
|  | console.log(status); // error |
|  |  |
|  | if (status === "error") { |
|  | alert("Could not find idontexist.php"); |
|  | } |
|  | }); |
|  |  |
|  | }); |

**>> getJSON() to query API for JSON**

$(function () {

  var flickerAPI = "https://www.flickr.com/services/feeds/photos\_public.gne?jsoncallback=?";

  $.getJSON(flickerAPI,{

    //options

  }).done(function(data){

    //success

    console.log(data);

  }).fail(function(){

    //alert("Ajax call failed")

  }).always(function(){

// it will always be executed

});

});

$.each(data.items, function(index,value)) >> if data was array

$.each(data.items, function(key,value)) >> if data was object

Converts thae first letter to upper case

Var name = user.name.charAt(0).toUpperCase() + user.name.slice(1);