Unit V – Introduction to jQuery

Infosys
be more

Copyright Guideline

Agenda

- Introduction
- jQuery Selectors
- jQuery HTML
- Event Handling
- Animations
- Effects
- DOM – jQuery DOM Traversing, DOM Manipulation
- jQuery AJAX

Introduction to jQuery

Infosys® | Education, Training and Assessment

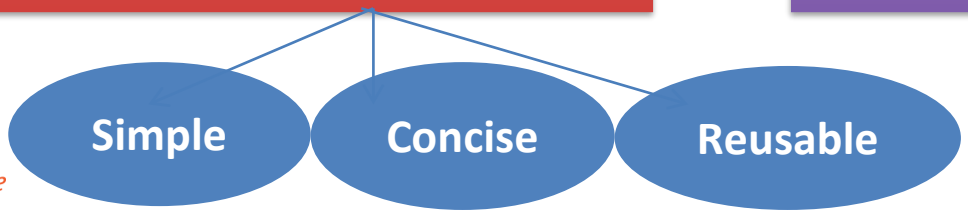
Why jQuery

Lightweight JavaScript Library

Created by John Resig (2006)

Java Scripting

Ajax based applications



Interesting Stats About jQuery

19,909,315 websites

216,123 top million most visited websites

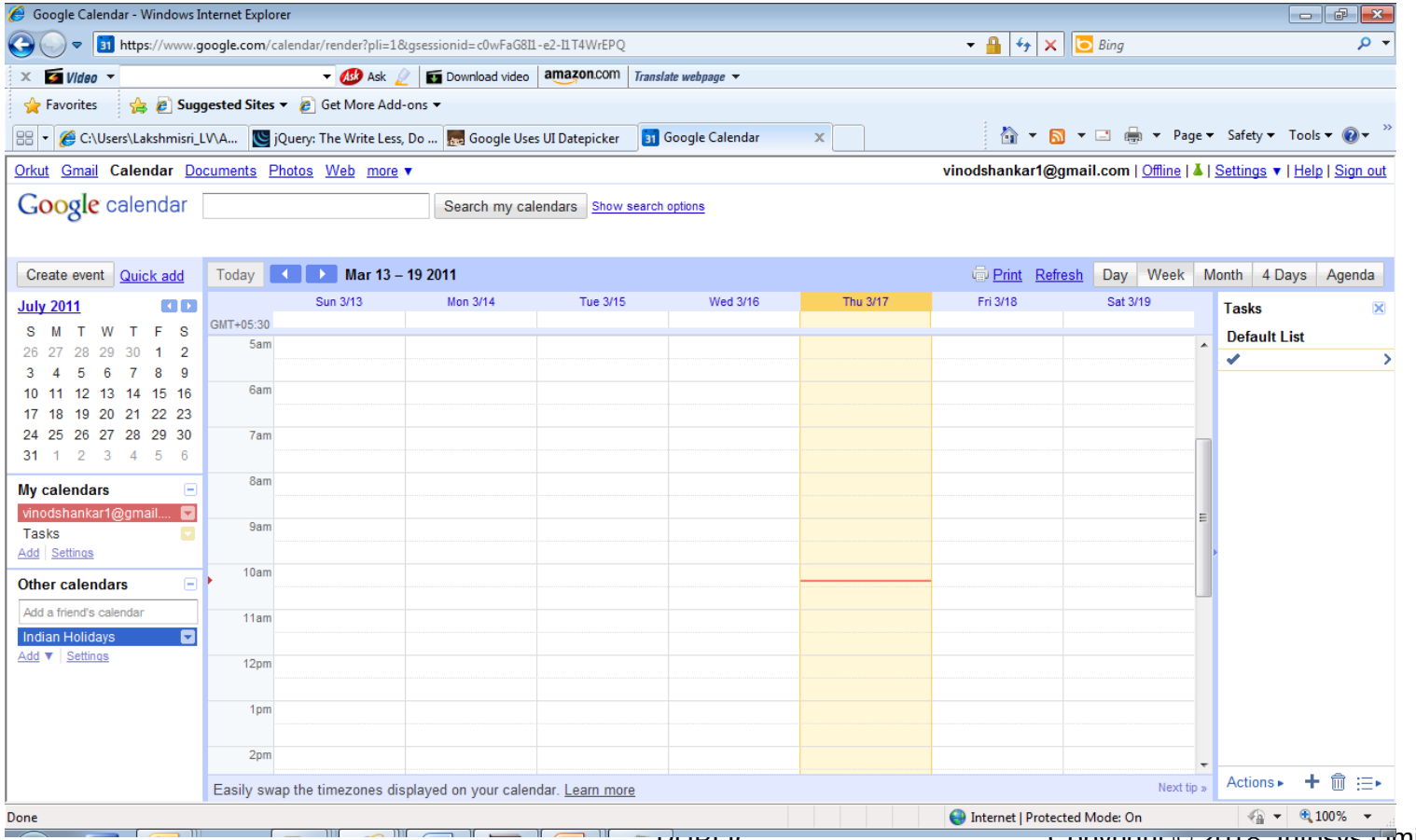
Most popular JavaScript library in use today. – Wikipedia

4,830 official jquery plugins

Who Is Using jQuery?

- Google
- DELL
- Bank Of America
- NBC
- mozilla.org
- Wordpress
- drupal.org
- Digg
- NETFLIX
- Major League BaseBall

Google Calendar



Need For Using jQuery

Simple, Fast and Concise

100 of Plug-ins

No Heavy Framework

Very Less Coding

Good in Handling AJAX

Total Size – 115 KB

Minified Size - 55 KB

Light Weighted

Cross Browser

CSS syntax & great documentation

Features of jQuery

- DOM element selection Functionalities
- Provides simple functions to traverse through the DOM elements and modify them.
- Event handling
- CSS manipulation
- Creating effects and animations in web pages.
- Simplifying Ajax interaction to web.
- Creating reusable JavaScript Plug-ins
- Extensibility

jQuery vs JavaScript

JavaScript	JQuery
Complex Java Script Functionality with more coding	Complex Java Script Functionality with very minimal coding
Lot of Iterative looping required for DOM Traversal	Very less Iterative looping required for DOM Traversal
Obscure Elements are not reachable.	Any obscure element can be reached out in a HTML page with simple code.

Example..

- Consider the following problem statement.
- All the textboxes which have the value “-” should be replaced by the value “Nil”.
- Lets see the code with and without jQuery.

WITHOUT JQUERY

```
var elements = document.getElementsByTagName("input");
for(var i = 0; i < elements.length; i++) {
    if(elements[i].type == "text")
        if(elements[i].value == "-") {
        alert(elements[i].value);
                elements[i].value="Nil";

     }
    }
```

WITH JQUERY

- $("input[type=text][value='-']").val("Nil");

  - As it is evident from the example above,
  - No iterative looping is necessary to access the DOM elements
  - Code is much simpler and concise.

http://Jquery.com/

- Download any latest version from http://Jquery.com/ in your system at the desired location.
- To use Jquery, include the following piece of code in your web page:

```
<script type=”text/javascript” src=”path/Jquery.js”></script>
```

What is '$' in jQuery

$	Jquery Namespace '$("Selector-Expression)' will return all the selected objects	$('p') = jQuery('p') This will select all paragraphs Equivalent DOM expression: document.getElementByTagName("p")

$(DOCUMENT).READY

This is almost similar to the JavaScript **window.onload** event just with the below mentioned vital difference.

Window.onLoad()

Code mentioned in window.onLoad event will be executed only when the entire content is loaded and visible to the user, including all images/style/animation effects, if any.

Document.ready()

In most of the cases, the JavaScript code execution should take place the moment DOM tree is created by the browser which does not necessarily need all the images to be loaded. If this is the case, they we can go for ready function of jQuery.

EXAMPLE


```
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
      <title>JSP Page</title>
      <script src="jquery.js"></script>
      <script language="JavaScript">
         $(document).ready(function()
         {
               $("a").click(function()
       {
          alert("Im being clicked");
        });
       });
      </script>
   </head>
<body>
      <a href="">This is dummy link</a>

   </body>
</html>
```

There is a shortcut to ready
$(document).ready(function(){});
Its as follows: $(function(){ });

jQuery.holdReady(hold)

- This is a feature introduced from jquery 1.6.
- The $.holdReady() method is used to delay the jQuery's ready event.
- This comes very handy when we need to dynamically load scripts before ready events are fired:

```
$.holdReady(true);
$.getScript("scriptBeforeReady.js", function() {
    $.holdReady(false);
// now the ready event can fire
});
```

Creating a jQuery program

The following are the steps involved to create a jQuery program:

- Get the HTML file ready
- Include the jQuery library(As discussed before)
- Add in the jQuery document ready state Handler function. This is the standard pattern used in most of the jQuery applications wherein the jQuery code resides.
- Write in the jQuery code within the ready state handler function

jQuery Selectors

Selectors

- jQuery supports DOM Element selections.It internally uses the cross-browser open source selector engine Sizzle(Since jQuery 1.4)
- $(“Filter Expression”) would return the elements that satisfy the given filter.
- Consider the expression $(‘p’).This would return all the elements which are of the type ‘p’.
- jQuery gives a wide variety of selector expressions

Basic Selectors

Selectors	Description
$("p#id")	This would select all the 'p' elements with the specified id.
$("p.className")	This would select all the 'p' elements that belong to the specified style-class.
$("input[type=text]")	This will select all the input elements of the type 'text'. This format can be used for an attribute based selection.

jQuery HTML

- Set or get the content and markup of the specified HTML elements
- This function cannot be applied on input elements like textbox, textarea etc.

Quick element construction

- This feature was introduced from jquery 1.4
- Using jQuery function, we can create elements and also add attributes and events at the same time.

```
jQuery("<div/>", {
 id: "test",
css: { height: "100px", width: "100px", backgroundColor: "cyan" },
mouseover: function() { $(this).css("backgroundColor", "silver");
}
}).appendTo("body");
```

More Functions

Functions	Description	Example
Val()	set/get the value of elements like textbox, text area etc. Not on HTML elements like div, span etc.	$(“#text1”).val(“initialValue”);
text()	Just inserts the text as it is.	$("div1").text("This is being created on the fly");
Attr()	attr() method returns undefined for all the undefined attributes.	var name = $(‘div’).attr(‘name’);
attr(attributeName,value)	The attr method can be used as a setter method It can also be used to set several attributes at once	$(‘img’).attr(‘title’,’A great photo at the Himalayas’); $('#pic1').attr({ alt: Nepalr', title: 'photo by Lakshmi' });

Learning each Function

- This comes handy when we need to invoke a function for every matched element.
- Ex:

```
$("div").each(function()
{
alert($(this).html())
});
```

- This would mean that every time the function is executed on the selected elements,' this' keyword points to the current element.

MAP function

- Map function can be used to invoke every matched element .
- The difference between each and map function is that map() function returns a jQuery Object containing all the return values.
- The array returned by map function is a jQuery-wrapped array. So generally, get() method is used to convert the jQuery-wrapped array to a basic array.

- Example:

```
$("input").map(function(){ return $(this).val(); }).get().join(", ") );
```

jQuery CSS

CSS Manipulation

- **Inline Styling**

  $(“p”).css(“background-color”,”red”);

- **For adding/removing class**

  $("p ").addClass("red");

- **Incase now if you want to change the class of <p> from “red” to “purple”**

  $("p"). removeClass("red").addClass("purple”);

RELATIVE CSS

- This is a feature introduced from jquery 1.6
- It is possible to modify the CSS properties relatively using the CSS function.
  // move 20px to the right

```
$("#element").css(“right", "+=20px");
```

jQuery Event Handling

Event Handling

- We have many event handlers in jQuery like load, blur, click, unload etc.
- $(this) is the object which should be used to invoke Jquery methods on that element (source of the event).
- Some of the functions are:
  - Click()
  - Keypress()
  - Mouseover()
  - Mouseout()
  - Hover()
  - Scroll()

Event Handling-Continued..


```
$("a").click(function(e){
e.preventDefault();
alert("I'm being clicked");
});
```

- preventDefault as the name suggests will stop the event to behave in its default ways. For example, in the above case, preventDefault function will stop the link from navigating to the url specified in its href attribute.

jQuery DOM Transverse

Infosys® | Education, Training and Assessment

Attribute Selector

- Contains: $('[attribute*="value"]')
- Contains Word: $('[attribute~="value"]')
- Starts With: ('[attribute^="value"]')
- Ends With: ('[attribute$="value"]')
- Equals: $("[attribute='value']"))
- Not Equals: $('[attribute!="value"]')
- Prefix: $(":eq(index)")
- Has: :has()

Advanced Selection

- Contains:

  $(“div:contains(‘abc’)”) ").append("Appending");

- Not Selector

  $("div").not("#div1").append("Appending");

- Has selector

  $("div:has(p)").append("Appending");

- Combining has and not(does not have)

  $("div").not(“:has(p)”).append("Appending");

Attribute Based Selection

Attribute Based Selection is possible in jQuery using the syntax below:

Eg: $(“input[type=text]”)

Syntax: [attribute!=value]

This will give an array of elements which do not have the value specified.

[attribute*=value]

That is if a part of the attribute value contains the specified value.

[attribute^=value]

Attribute value starts with the given value

[attribute$=value]

Attribute value ends with the given value

Using +

‘+’ is the operator which is used to select any field element of the specified type which is exactly beside the field of a particular type.

For example:

```
<label>user</label>

<input type="text" maxlength="30" />

<input type="text" maxlength="30" />
```

$("label+input") would select **only the first textbox!**

Using '~'

- '~' is used to select all elements which are next to the specified element
- This is as against '+' filter , where only the exact successive elements will be selected.

- Lets consider the same example

```
<label>user</label>
<input type="text" maxlength="30" />
<input type="text" maxlength="30" />
```

- In this case, both the textboxes will be selected by using $("label~input").

Parent-Child

- '>' is the filter used for selecting on the basis of parent-child hierarchy.
- Consider the example

```
$("#test  >p").html("new -value");
```

In this case, the paragraph(<p>) which is the immediate child of element having id 'test', gets selected.

Ancestor-Descendant

Consider this piece of code..

```
<div> I'm a div element
    <p>I'm a para
    <input type="text" value="Im not set"/>
    <input type="text" value="Im also not set"/>
   </p>
</div>
```

Say for example, I need to set the textboxes in the above example, using div.

Ancestor-Descendant (Cont.)

- In that case, we cannot use $("div >input").val("Im set");
- Since the input element is not the immediate child of div.

  So in this case, whenever we have the child in the hierarchy but may not necessarily be the immediate child, then we should not be using the '>' symbol.

- The code to select the textboxes would be

$("div input").val("Im set“);

- This is equivalent to **$(“div").find(“input"). val(“Im set”);**

Find Parents

- You can find out the parent elements by using the parents function

Ex: $(this).parents("a").addClass("red");

Position based Selection

- **First Selector :** This would select the first element satisfying the given condition.

  Ex: $("td:first").css("background-color","red");

- **Last Selector :** This would select the last element satisfying the given condition.

- **Even:** It selects all the elements with even indicies, starting from 0.

- **Odd:** Selects all the odd indexed elements.

Position based Selection (Cont.)

Eq-

This is used to select the element with the specified index

For Example,

```
$("td:eq('2')")
```

- Similar to this we can use **gt** for greater than the specified index, **lt** etc.

jQuery Effects

Infosys® | Education, Training and Assessment

Basic Effects

Basic:

- Hide: .hide()
- Show: .show()
- Toggle: .toggle()

Fading:

- Fadein: .fadeIn()
- Fadeout: .fadeOut([duration], [easing], [callback])
- Fade Toggle: .fadeToggle([duration], [easing], [callback])
- Fade To .fadeTo()

Slide Effects

- Slide Up: .fadeIn()
- Slide Down: .fadeOut([duration], [easing], [callback])
- Slide Toggle: .fadeToggle([duration], [easing], [callback])

Hide/Display

- **show()**: This is used to enable the display of the specified element.
- **hide() :** This is used to hide the display of the specified element.
- **toggle():** This function is used to toggle the display.
- **toggleClass:** This function is used to toggle the style of the specified element between two CSS classes.

Example..

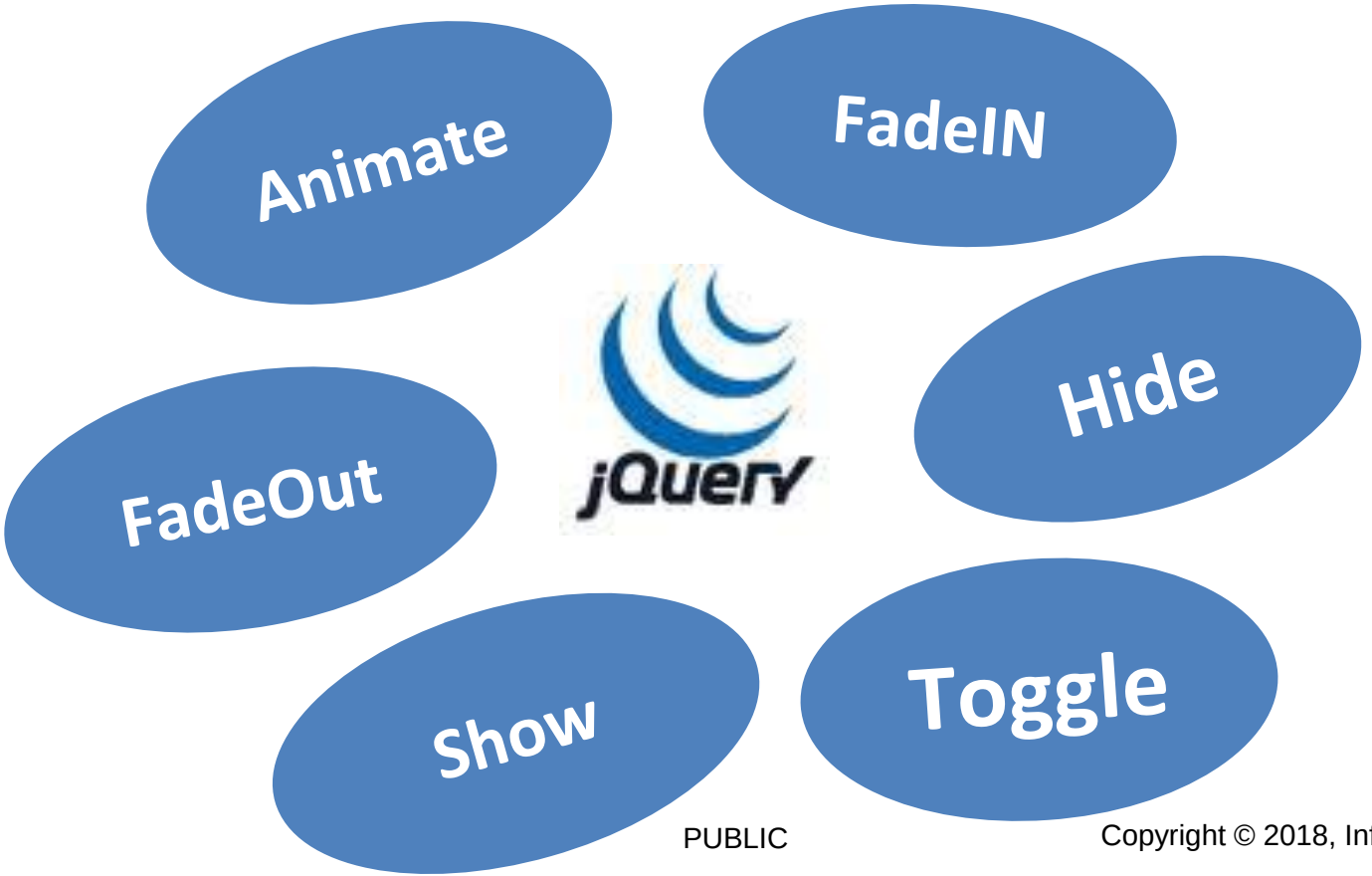

```
$(function()
 {
    $("#check1").click(
    function()
    {
    $("#disp").toggle();
    });
});
```

To toggle between the classes,

```
$("#second").toggleClass("highlight");
```

jQuery Animations

Some Animation Functions In jQuery



setTimeout function

setTimeout(function() { ... }, 300000);

- The function specified within the setTimeOut function will execute after the specified amount of time.

setInterval

setInterval('repeatthis()', 1000);

- The function specified within the setInterval function will repeatafter the specified amount of time.

Animate Function

- Using animate() function, the different style properties can be animated. The extent to which animation is required can be specified. Example: height, width, top, opacity, font-size, etc.
- The speed at which animation is supposed to happen can also be optionally specified.
- Syntax:

  (selector).animate({styles}, speed, easing, callback)

- Example:

```
$('#test').animate({ height: 400, width: 300 }, "slow", function(){
   alert(Im getting animated!');
});
```

Chaining property

- Using chaining property of jQuery, it is possible to club several functions together.

- **$(".start").click(function(){**

```
$("#element1").animate({height: 400, width: 300 }, "slow“,)
.slideUp("fast")
.slideDown("slow")

});
```

Plug-ins

- Plugins are files which contain a set of user defined functions that are repeatedly used in an application They can be used by simply including the files names.

- Example:

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

Keypoints

Key points to remember while writing plugin:

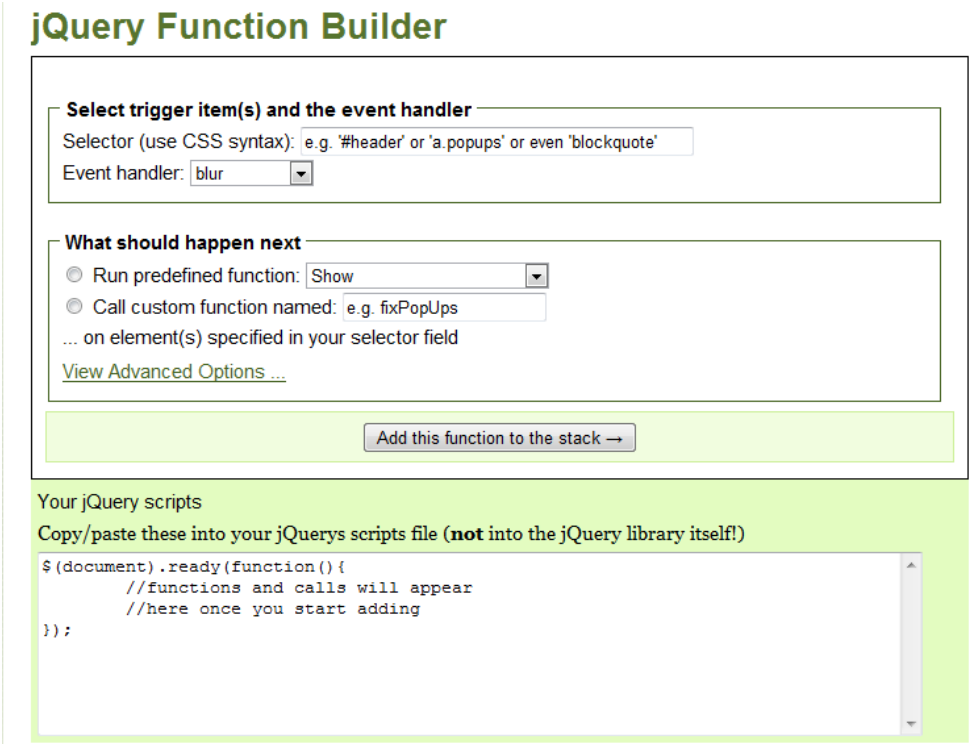
- Plugin Name should be in the following format**: Jquery.[name of plugin].js**

  Ex: jquery.test.js

- All the user defined functions should be attached to the Jquery.fn object. At the end of your function there should be a semicolon (;)

jQuery Builder Tool

- http://accessify.com/tools-and-wizards/developer-tools/jquery-builder/default.php



jQuery AJAX

Infosys® | Education, Training and Assessment

jQuery AJAX

- Ajax is a powerful technology for building Web 2.0 applications today.
- jQuery offers variety of methods which makes Ajax calls very simple to implement.
- jQuery Ajax support is available as part of its core library itself.

Ajax()

- $.ajax(options) - is used to perform an AJAX (asynchronous HTTP) request.

```
$("button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
        $("#div1").html(result);
    }});
});
```

Load()

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

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

- Example:

```
$("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);
    });
});
```

Ajax Get() and Post()

- The jQuery get() and post() methods are used to request data from the server with an HTTP GET or POST request.

- jQuery $.get() Method - requests data from the server with an HTTP GET request.

Syntax: $.get(*URL,callback*);

Example:

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

Ajax Get() and Post() (Cont.)

- jQuery $.post() Method - requests data from the server using an HTTP POST request.

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

Example:

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

Summary

- You are now knowledgeable on:
  - Implementation of jQuery
    - Selectors
    - HTML
    - Event Handling
    - Animations
    - Effects
    - DOM – jQuery DOM Traversing, DOM Manipulation
    - AJAX

Reference

- **Links:**
  - https://www.w3schools.com/jquery/default.asp
  - https://www.tutorialrepublic.com/jquery-tutorial/
  - https://www.tutorialspoint.com/jquery/index.htm
  - https://www.javatpoint.com/jquery-tutorial
  - www.jquery-tutorial.net/
  - www.tutorialsteacher.com/jquery/jquery-tutorials
  - https://www.codeschool.com/courses/try-jquery

- **Videos:**
  - JQuery Tutorial - YouTube

Thank You