**Q. What is JavaScript?**

**ANS**:

JavaScript is a client-side as well as server side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript is also an Object Oriented Programming language.

**Q. What is the use of is NaN function?**

**ANS: isNan** function returns true if the argument is not a number otherwise it is false.

**Q. What are the types used in JavaScript?**

**ANS:** String, Number, Boolean, Function, Object, Null, Undefined.

**Q. How to get textbox value in JavaScript?**

**ANS:**

<!DOCTYPE html>

<html>

<body>

Full name: <input type="text" id="txtFullName"

name="FirstName" value="Vikas Ahlawat">

</body>

</html>

var name = document.getElementById('txtFullName').value;

**Q. How will you get the Checkbox status whether it is checked or not?**

**ANS:**

var status = document.getElementById('checkbox1').checked;

alert(status);

**Q. How do you submit a form using JavaScript?**

**ANS:** Use document.forms[0].submit();

**Q. What is Negative infinity JavaScript?**

**ANS:** Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

**Q. What is JQUERY?**

**ANS:**

JQuery is a cross-browser lightweight JavaScript library. In simple words jQuery has been designed to make navigation to any element easier or adding/invoking event handlers on your HTML page and also simplify the way you access the elements in your web pages, provide help in working with client-side events, enable visual effects like animation, and make it easier to use Ajax in your applications. You can download the latest version of jQuery from the [official web site](http://jquery.com/).

A quick look at what is available in jQuery:

* Cross-browser support and detection.
* AJAX functions
* CSS functions
* DOM manipulations
* DOM transversal
* Attribute manipulation
* Event detection and handling.
* JavaScript animation
* Hundreds of plugins for pre-built user interfaces, advanced animations, form validation, etc.
* Expandable functionality using custom plugins.

**Q. What is difference between JavaScript and JQuery?**

**ANS:**

|  |  |  |
| --- | --- | --- |
| SR No | JavaScript | JQuery |
| 1 | JavaScript is a language. | JQuery is a framework built with JavaScript to help JavaScript programmers who are doing common web tasks. |
| 2 | In case of using JavaScript you are required to writer your own script that can be time consuming. | In case of using jQuery you are not required to write much scripting that already exists in libraries. |
| 3 | Developers need to handle by writing their own JavaScript code. | JQuery is a multi-browser JavaScript library designed to simplify the client-side scripting of HTML. |
| 4 | JavaScript is a scripting language that works with all web browsers | JQuery is only a framework that is a fast and concise JavaScript library that simplifies the HTML document. |
| 5 | Animations are not possible using JavaScript | Animations are possible using JQuery |
| 6 | JavaScript is an Object Oriented Programming (OOP) language | JQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. |

**Q. What is Jquery.noConflict?**

**ANS:**

JQuery no-conflict is an option given by jQuery to overcome the conflicts between the different js frameworks or libraries. When we use jQuery no-conflict mode, we are replacing the $ to a new variable and assigning to jQuery some other JavaScript libraries. Also use the $ (Which is the default reference of jQuery) as a function or variable name what jQuery has. And in our development life, we are not at all strict to only jQuery.

**Q. What is CDN in JQuery?**

**ANS:**

**CDN** stand for Content Delivery Network.

A [Content Delivery Network](http://en.wikipedia.org/wiki/Content_delivery_network) (CDN) is a system of computers that exist all over the world and cache files for users to access. CDNs can greatly reduce the load time of a page by offering files at a higher bandwidth from a server that is physically closer to your visitor than your web server might be. Web developers can leverage the use of CDNs (and the good graces of Google and Microsoft) to host their jQuery Library for faster access and improved performance.

**Q. Why should I use a CDN to host jQuery??**

**ANS:**

* The jQuery library download time will be reduced. Your users in Europe will hit the CDN in Europe and your users in the US should hit the US CDN. As a result, this will reduce your overall page load time.
* The jQuery library will already be cached in the visitor's browser if they visited another website that references the same jQuery library. In this case, the visitor won't even have to download the jQuery library.
* It's free!

**Example –**

head>

<script type="text/javascript" src="http://ajax.microsoft.com/ajax/jquery/jquery-1.6.1.min.js"></script>

</head>

**Q. What is Selector in JQuery? What are types of Selectors in JQuery?**

**ANS:**

Selectors allow page elements to be selected. Single or multiple elements are supported.

A Selector identifies an HTML element/tag that you will manipulate with jQuery code.

**Types of Selectors in JQuery –**

## 1. Select Elements by ID in jQuery

## The fastest type of Selector which we can use with jQuery is Id Selector. Normally, in JavaScript, we use document.getElementByID to find a specific Id. In jQuery, it is more compact than that.

## Use # character to select elements by ID.

## Example: $(‘#myID’).

The above Selector will select <p id=”myID”> element.

## 2. Select Elements by Class in jQuery

Use dot(.) character to select elements by class name.

Example: $(‘.myClass’)

The above selector will select <p class=”myClass”> element.

## 3. Select Elements by Attribute Value in jQuery

Use brackets [attribute] to select based on attribute name and/or attribute value.

Example: $(‘a[title]‘).

The above selector will select all <a> elements that have a title attribute.

$(‘a[title="Customer Info"]‘) will select all anchor elements that have a “Customer Info” title attribute.

## 4. Select Input Nodes in jQuery

A new character syntax $(‘:input’) used for this.

The above syntax will select all the input elements including Input, Select, TextArea, Button, Image, Radio and more. It will return a collection of input elements.

$(‘:input[type="radio"]‘) will specifically target RadioButtons on the web page.

**Q. What are the methods used to provide effects?**

**ANS:**

JQuery provides many amazing effects, we can apply these effects quickly and with simple configuration. The effect may be hiding, showing, toggling, fadeout, fadein, fadeto and so on toggle (), Show() and hide() methods. Similarly we can use other methods as in the following:

* **animate( params, [duration, easing, callback] )** This function makes custom animations for your HTML elements.
* **fadeIn( speed, [callback] )** This function fades in all the matched elements by adjusting their opacity and firing an optional callback after completion.
* **fadeOut( speed, [callback] )** This function is used to fade out all the matched elements by adjusting their opacity to 0, then setting the display to "none" and firing an optional callback after completion.
* **fadeTo( speed, opacity, callback )** This function fade the opacity of all the matched elements to a specified opacity and firing an optional callback after completion.
* **stop( [clearQueue, gotoEnd ])** This function stops all the currently running animations.

**Q. What is the difference between $(window).load and $(document).ready function in jQuery?**

**ANS:**

**$(document).ready** is jQuery event that is fired when DOM is loaded, so it’s fired when the document structure is ready.

**$(window).load** event is fired after whole content is loaded like page contain images, css etc.

The Difference between **$(document).ready()** and **$(window).load()** functions is that the code included inside **$(window).load()** will run once the entire page(images, iframes, stylesheets,etc) are loaded whereas the document ready event fires before all images,iframes etc. are loaded, but after the whole DOM itself is ready.

<script type="text/javascript">

$(document).ready(function () {

// executes when HTML-Document is loaded and DOM is ready

alert("document is ready");

});

$(window).load(function () {

// executes when complete page is fully loaded, including all frames, objects and images

alert("window is loaded");

});

</script>

**Q. When to use $(window).load instead of $(document).ready?**

**ANS:**

In most cases, the script can be executed as soon as the DOM is fully loaded, so ready() is usually the best place to write your JavaScript code. But there could be some scenario where you might need to write scripts in the load() function. For example, to get the actual width and height of an image.  
  
As we know the $(window).load event is fired once the DOM and all the CSS, images and frames are fully loaded. So, it is the best place to write the jQuery code to get the actual image size or to get the details of anything that is loaded just before the load event is raised.

**Q. How to handle Controls attribute using jQuery?**

**ANS:**

For handle Controls attribute using jQuery we used **.addClass(),** **.removeClass(),** **.css(), .toggleClass(),**etc. to manage all css and html attributes of any html control.

**Q. What is chaining in jQuery?**

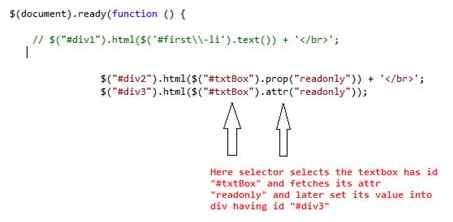
**ANS:**

Chaining is a powerful feature of jQuery. Chaining means specifying multiple functions and/or selectors to an element. Chaining reduces the code segment and keeps it very clean and easy to understand. Generally chaining uses the jQuery built in a function that makes compilation a bit faster.

By using chaining we can write the above code as follows:

1. $(document).ready(function() {
2. $("#div2").html($("#txtBox").prop("readonly")) + '</br>';
3. $("#div3").html($("#txtBox").attr("readonly"));
4. });

The code segment above is described by the following image:



**Q. Explain Ajax in jQuery.**

**ANS:**

AJAX stands for “Asynchronous JavaScript and XML”. AJAX is about exchanging data with a server, without reloading the whole page. It is a technique for creating fast and dynamic web pages.  
  
In .NET, we can call server side code using two ways:

1. ASP .NET AJAX
2. jQuery AJAX

In this article we will focus on jQuery Ajax.  
  
**$.ajax () Method**  
  
JQuery’s core method for creating Ajax requests. Here are some jQuery AJAX methods:

* **$.ajax()** -Performs an async AJAX request.
* **$.get()** -Loads data from a server using an AJAX HTTP GET request.
* **$.post()** - Loads data from a server using an AJAX HTTP POST request.

To know more [click](https://learn.jquery.com/ajax/jquery-ajax-methods/).  
  
**$.ajax () Method Configuration option**  
  
Options that we use:

* async
* type
* url
* data
* datatype
* success
* error

**async**  
Set to false if the request should be sent synchronously. Defaults to true.   
  
Note that if you set this option to false, your request will block execution of other code until the response is received.

Example:

1. async: false,

**Type**  
This is type of HTTP Request and accepts a valid HTTP verb.  
  
The type of the request, "POST" or "GET" defaults to "GET". Other request types, such as "PUT" and "DELETE" can be used, but they may not be supported by all the web browsers.  
  
Example:

1. type: "POST",

**url**  
The URL for the request.  
  
Example:

1. url: "/Home/JqAJAX",

**data**  
The data to be sent to the server. This can either be an object or a query string.   
  
Example:

1. data: JSON.stringify(model\_data),

**dataType**  
The type of data you expect back from the server. By default, jQuery will look at the MIME type of the response if no dataType is specified.   
  
Accepted values are text, xml, json, script, html jsonp.  
  
Example:

1. dataType: "json",

**contentType**  
This is the content type of the request you are making. The default is 'application/x-www-form-urlencoded'.  
  
Example:

1. contentType: 'application/json; charset=utf-8',

**success**  
A callback function to run if the request succeeds. The function receives the response data (converted to a JavaScript object if the DataType was JSON), as well as the text status of the request and the raw request object.

1. Success: function (result) {
2. $('#result').html(result);
3. }

**error**  
A callback function to run if the request results in an error. The function receives the raw request object and the text status of the request.

1. error: function (result) {
2. alert('Error occured!!');
3. },

**Q. Explain posting Ajax in jQuery.**

**ANS:**

**Let’s Post Values using JQuey,Ajax:**  
We often use the jQuery Ajax method in ASP.NET Razor Web Pages. Here is a sample code:

1. <input type="submit" id="btnSubmit" value="Add Student" />
2. <script type="text/javascript">
3. $(document).ready(function() {
4. $(function() {
5. $('#btnSubmit').click(function(event) {
6. event.preventDefault();
7. var Student = {
8. ID: '10001',
9. Name: 'Shashangka',
10. Age: 31
11. };
12. $.ajax({
13. type: "POST",
14. url: "/Home/JqAJAX",
15. data: JSON.stringify(Student),
16. dataType: "json"
17. contentType: 'application/json; charset=utf-8',
18. success: function(data) {
19. alert(data.msg);
20. },
21. error: function() {
22. alert("Error occured!!")
23. }
24. });
25. });
26. });
27. });
28. </script>

**Controller Action:**

1. // GET: Home/JqAJAX
2. [HttpPost]
3. public ActionResult JqAJAX(Student st) {
4. try {
5. return Json(new {
6. msg = "Successfully added " + st.Name
7. });
8. } catch (Exception ex) {
9. throw ex;
10. }
11. }

**Posting JSON**  
JSON is a data interchange format where values are converted to a string. The recommended way to create JSON is include the JSON.stringify method. In this case we have defined:

1. JSON.stringify(Student)

And the data type set to:

1. datatype: "json"

And the content type set to application/json

1. contentType: 'application/json; charset=utf-8'

Syntax:   
  
JSON.stringify(value[, replacer[, space]])  
  
Post Script:

1. var Student = {
2. ID: '10001',
3. Name: 'Shashangka',
4. Age: 31
5. };
6. $.ajax({
7. type: "POST",
8. url: "/Home/JqAJAX",
9. data: JSON.stringify(Student),
10. contentType: 'application/json; charset=utf-8',
11. success: function(data) {
12. alert(data.msg);
13. },
14. error: function() {
15. alert("Error occured!!")
16. }
17. });

**Controller Action:**

1. // GET: Home/JqAJAX
2. [HttpPost]
3. public ActionResult JqAJAX(Student st) {
4. try {
5. return Json(new {
6. msg = "Successfully added " + st.Name
7. });
8. } catch (Exception ex) {
9. throw ex;
10. }
11. }

**JSON Response Result:**  
Sent data format:  
{"ID":"10001","Name":"Shashangka","Age":31}  
Received Data format:  
{"msg":"Successfully added Shashangka"}  
  
Let’s Post JavaScript Objects:  
  
To send JavaScript Objects we need to omit the JSON.stringify(Student) method and we need to pass the plain object to the data option. In this case we have defined:

1. data: Student

And the datatype set to:

1. datatype: "html"

And the content type set to default.

1. contentType: 'application/x-www-form-urlencoded'

Post Script:

1. var Student = {
2. ID: '10001',
3. Name: 'Shashangka',
4. Age: 31
5. };
6. $.ajax({
7. type: "POST",
8. url: "/Home/JqAJAX",
9. data: Student,
10. contentType: 'application/x-www-form-urlencoded',
11. datatype: "html",
12. success: function(data) {
13. alert(data.msg);
14. },
15. error: function() {
16. alert("Error occured!!")
17. }
18. });

Controller Action:

1. // GET: Home/JqAJAX
2. [HttpPost]
3. public ActionResult JqAJAX(Student st) {
4. try {
5. return Json(new {
6. msg = "Successfully added " + st.Name
7. });
8. } catch (Exception ex) {
9. throw ex;
10. }
11. }

JavaScript Objects Response Result:  
  
Sent data format:  
ID=10001&Name=Shashangka&Age=31   
Received Data format:  
{"msg":"Successfully added Shashangka"}  
  
Let’s Post JavaScript Arrays:  
  
To send Array we need to omit the JSON.stringify(Student) method and we need to pass the plain object to the data option. In this case we have defined:

1. data: Student

And the datatype set to:

1. datatype: "html"

And the content type set to default

1. contentType: 'application/x-www-form-urlencoded'

Post Script:

1. var ID = ["Shashangka", "Shekhar", "Mandal"];
3. $.ajax({
4. type: "POST",
5. url: "/Home/JqAJAX",
6. data: {
7. values: ID
8. },
9. datatype: "html",
10. contentType: 'application/x-www-form-urlencoded',
11. success: function(data) {
12. alert(data.msg);
13. },
14. error: function() {
15. alert("Error occured!!")
16. }
17. });

**Controller Action:**

1. // GET: Home/JqAJAX
2. [HttpPost]
3. public ActionResult JqAJAX(string[] values) {
4. try {
5. return Json(new {
6. msg = String.Format("Fist Name: {0}", values[0])
7. });
8. } catch (Exception ex) {
9. throw ex;
10. }
11. }

Array Response Result:  
  
Sent data format:  
values[]=Shashangka&values[]=Shekhar&values[]=Mandal  
Received Data format:  
{"msg":"Fist Name: Shashangka"}

### Q. What does dollar sign ($) means in jQuery?

**ANS:**

Dollar Sign is nothing but it's an alias for JQuery. Take a look at below jQuery code.

Hide   Copy Code

$(document).ready(function(){

});

Over here $ sign can be replaced with "jQuery" keyword.

Hide   Copy Code

jQuery(document).ready(function(){

});

### Q. Can we have multiple document. Ready () function on the same page?

**ANS:**

**YES**. We can have any number of document.ready() function on the same page.

### Q. Is there any difference between body onload() and document.ready() function ?

**ANS:**

document.ready() function is different from body onload() function for 2 reasons.

* We can have more than one document.ready() function in a page where we can have only one body onload function.
* document.ready() function is called as soon as DOM is loaded where body.onload() function is called when everything gets loaded on the page that includes DOM, images and all associated resources of the page.

### Q. What is the difference between .js and .min.js?

**ANS:**

JQuery library comes in 2 different versions Development and Production/Deployment. The deployment version is also known as minified version. So .min.js is basically the minified version of jQuery library file. Both the files are same as far as functionality is concerned. But .min.js is quite small in size so it loads quickly and saves bandwidth.

### Q.Why there is two different version of jQuery library?

Ans: jQuery library comes in 2 different versions.

1. Development
2. Production/Deployment

The development version is quite useful at development time as jQuery is open source and if you want to change something then you can make those changes in development version. But the deployment version is minified version or compressed version so it is impossible to make changes in it. Because it is compressed, so its size is very less than the production version which affects the page load time.

### Q17. What is a CDN?

Ans: A content delivery network or content distribution network (CDN) is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a CDN is to serve content to end-users with high availability and high performance.

### Q18. Which are the popular jQuery CDN? and what is the advantage of using CDN?

Ans: There are 3 popular jQuery CDNs.

1. 1. Google.
2. 2. Microsoft
3. 3. jQuery.

Advantage of using CDN.

* It reduces the load from your server.
* It saves bandwidth. jQuery framework will load faster from these CDN.
* The most important benefit is it will be cached, if the user has visited any site which is using jQuery framework from any of these CDN

### Q19. How to load jQuery from CDN?

Ans: Below is the code to load jQuery from all 3 CDNs.  
**Code to load jQuery Framework from Google CDN**

Hide   Copy Code

<script type="text/javascript"

src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js">

</script>

**Code to load jQuery Framework from Microsoft CDN**

Hide   Copy Code

<script type="text/javascript"

src="http://ajax.microsoft.com/ajax/jquery/jquery-1.9.1.min.js">

</script>

**Code to load jQuery Framework from jQuery Site(EdgeCast CDN)**

Hide   Copy Code

<script type="text/javascript"

src="http://code.jquery.com/jquery-1.9.1.min.js">

</script>

### Q20. How to load jQuery locally when CDN fails?

Ans: It is a good approach to always use CDN but sometimes what if the CDN is down (rare possibility though) but you never know in this world as anything can happen.  
  
Below given jQuery code checks whether jQuery is loaded from Google CDN or not, if not then it references the jQuery.js file from your folder.

Hide   Copy Code

<script type="text/javascript">

if (typeof jQuery == 'undefined')

{

document.write(unescape("%3Cscript src='Scripts/jquery.1.9.1.min.js' type='text/javascript'%3E%3C/script%3E"));

}

</script>

It first loads the jQuery from Google CDN and then check the jQuery object. If jQuery is not loaded successfully then it will references the jQuery.js file from hard drive location. In this example, the jQuery.js is loaded from Scripts folder.

### Q21. What are selectors in jQuery and how many types of selectors are there?

Ans: To work with an element on the web page, first we need to find them. To find the html element in jQuery we use selectors. There are many types of selectors but basic selectors are:

* Name: Selects all elements which match with the given element Name.
* #ID: Selects a single element which matches with the given ID
* .Class: Selects all elements which match with the given Class.
* Universal (\*): Selects all elements available in a DOM.
* Multiple Elements E, F, G: Selects the combined results of all the specified selectors E, F or G.
* Attribute Selector: Select elements based on its attribute value.

### Q22. How do you select element by ID in jQuery?

Ans: To select element use ID selector. We need to prefix the id with "#" (hash symbol). For example, to select element with ID "txtName", then syntax would be,

Hide   Copy Code

$('#txtName')

### Q23. What does $("div") will select?

Ans: This will select all the div elements on page.

### Q24. How to select element having a particular class (".selected")?

Ans: $('.selected'). This selector is known as class selector. We need to prefix the class name with "." (dot).

### Q25. What does $("div.parent") will select?

Ans: All the div element with parent class.

### Q26. What are the fastest selectors in jQuery?

Ans: ID and element selectors are the fastest selectors in jQuery.

### Q27. What are the slow selectors in jQuery?

Ans: class selectors are the slow compare to ID and element.

### Q28. How jQuery selectors are executed?

Ans: Your last selectors is always executed first. For example, in below jQuery code, jQuery will first find all the elements with class ".myCssClass" and after that it will reject all the other elements which are not in "p#elmID".

Hide   Copy Code

$("p#elmID .myCssClass");

### Q29. Which is fast document.getElementByID('txtName') or $('#txtName').?

Ans: Native JavaScipt is always fast. jQuery method to select txtName "$('#txtName')" will internally makes a call to document.getElementByID('txtName'). As jQuery is written on top of JavaScript and it internally uses JavaScript only So JavaScript is always fast.

### Q30. Difference between $(this) and 'this' in jQuery?

Ans: this and $(this) refers to the same element. The only difference is the way they are used. 'this' is used in traditional sense, when 'this' is wrapped in $() then it becomes a jQuery object and you are able to use the power of jQuery.

Hide   Copy Code

$(document).ready(function(){

$('#spnValue').mouseover(function(){

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

});

});

In below example, this is an object but since it is not wrapped in $(), we can't use jQuery method and use the native JavaScript to get the value of span element.

Hide   Copy Code

$(document).ready(function(){

$('#spnValue').mouseover(function(){

alert(this.innerText);

});

});

### Q31. How do you check if an element is empty?

Ans: There are 2 ways to check if element is empty or not. We can check using ":empty" selector.

Hide   Copy Code

$(document).ready(function(){

if ($('#element').is(':empty')){

*//Element is empty*

}

});

And the second way is using the "$.trim()" method.

Hide   Copy Code

$(document).ready(function(){

if($.trim($('#element').html())=='') {

*//Element is empty*

}

});

### Q32. How do you check if an element exists or not in jQuery?

Ans: Using jQuery length property, we can ensure whether element exists or not.

Hide   Copy Code

$(document).ready(function(){

if ($('#element').length > 0){

*//Element exists*

}

});

### Q33. What is the use of jquery .each() function?

Ans: The $.each() function is used to iterate over a jQuery object. The $.each() function can be used to iterate over any collection, whether it is an object or an array.

### Q34. What is the difference between jquery.size() and jquery.length?

Ans: jQuery .size() method returns number of element in the object. But it is not preferred to use the size()method as jQuery provide .length property and which does the same thing. But the .length property is preferred because it does not have the overhead of a function call.

### Q35. What is the difference between $('div') and $('<div/>') in jQuery?

Ans: $('<div/>') : This creates a new div element. However this is not added to DOM tree unless you don't append it to any DOM element.  
  
$('div') : This selects all the div element present on the page.

### Q36. What is the difference between parent() and parents() methods in jQuery?

Ans: The basic difference is the parent() function travels only one level in the DOM tree, where parents() function search through the whole DOM tree.

### Q37. What is the difference between eq() and get() methods in jQuery?

Ans: eq() returns the element as a jQuery object. This method constructs a new jQuery object from one element within that set and returns it. That means that you can use jQuery functions on it.  
  
get() return a DOM element. The method retrieve the DOM elements matched by the jQuery object. But as it is a DOM element and it is not a jQuery-wrapped object. So jQuery functions can't be used. Find out more [**here**](http://jquerybyexample.blogspot.com/2013/04/jquery-difference-between-eq-and-get-method.html).

### Q38. How do you implement animation functionality?

Ans: The .animate() method allows us to create animation effects on any numeric CSS property. This method changes an element from one state to another with CSS styles. The CSS property value is changed gradually, to create an animated effect.  
  
Syntax is:

Hide   Copy Code

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

* styles: Specifies one or more CSS properties/values to animate.
* duration: Optional. Specifies the speed of the animation.
* easing: Optional. Specifies the speed of the element in different points of the animation. Default value is "swing".
* callback: Optional. A function to be executed after the animation completes.

Simple use of animate function is,

Hide   Copy Code

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

$("#dvBox").animate({height:"100px"});

});

### Q39. How to disable jQuery animation?

Ans: Using jQuery property "**jQuery.fx.off**", which when set to true, disables all the jQuery animation. When this is done, all animation methods will immediately set elements to their final state when called, rather than displaying an effect.

### Q40. How do you stop the currently-running animation?

Ans: Using jQuery ".stop()" method.

### Q41. What is the difference between .empty(), .remove() and .detach() methods in jQuery?

Ans: All these methods .empty(), .remove() and .detach() are used for removing elements from DOM but they all are different.  
  
.empty(): This method removes all the child element of the matched element where remove() method removes set of matched elements from DOM.  
  
.remove(): Use .remove() when you want to remove the element itself, as well as everything inside it. In addition to the elements themselves, all bound events and jQuery data associated with the elements are removed.  
  
.detach(): This method is the same as .remove(), except that .detach() keeps all jQuery data associated with the removed elements. This method is useful when removed elements are to be reinserted into the DOM at a later time.  
  
Find out more [**here**](http://jquerybyexample.blogspot.com/2012/05/empty-vs-remove-vs-detach-jquery.html)

### Q42. Explain .bind() vs .live() vs .delegate() vs .on()

Ans: All these 4 jQuery methods are used for attaching events to selectors or elements. But they all are different from each other.  
  
<a href="http://jquerybyexample.blogspot.com/2010/06/jquery-bind-function-exampledemo.html">.bind()</a>: This is the easiest and quick method to bind events. But the issue with bind() is that it doesn't work for elements added dynamically that matches the same selector. bind() only attach events to the current elements not future element. Above that it also has performance issues when dealing with a large selection.  
  
<a href="http://jquerybyexample.blogspot.com/2010/06/jquery-live-function-exampledemo.html">.live()</a>: This method overcomes the disadvantage of bind(). It works for dynamically added elements or future elements. Because of its poor performance on large pages, this method is deprecated as of jQuery 1.7 and you should stop using it. Chaining is not properly supported using this method.  
  
<a href="http://jquerybyexample.blogspot.com/2010/08/jquery-delegate-function-exampledemo.html">.delegate()</a>: The .delegate() method behaves in a similar fashion to the .live() method, but instead of attaching the selector/event information to the document, you can choose where it is anchored and it also supports chaining.  
  
.on(): Since live was deprecated with 1.7, so new method was introduced named ".on()". This method provides all the goodness of previous 3 methods and it brings uniformity for attaching event handlers.  
  
Find out more [**here**](http://www.elijahmanor.com/2012/02/differences-between-jquery-bind-vs-live.html)

### Q43. What is wrong with this code line "$('#myid.3').text('blah blah!!!');"

Ans: The problem with above statement is that the selectors is having meta characters and to use any of the meta-characters ( such as !"#$%&'()\*+,./:;<=>?@[\]^`{|}~ ) as a literal part of a name, it must be escaped with with two backslashes: \\. For example, an element with id="foo.bar", can use the selector $("#foo\\.bar").  
So the correct syntax is,

Hide   Copy Code

$('#myid\\.3').text('blah blah!!!');

### Q44. How to create clone of any object using jQuery?

Ans: jQuery provides clone() method which performs a deep copy of the set of matched elements, meaning that it copies the matched elements as well as all of their descendant elements and text nodes.

Hide   Copy Code

$(document).ready(function(){

$('#btnClone').click(function(){

$('#dvText').clone().appendTo('body');

return false;

});

});

### Q45. Does events are also copied when you clone any element in jQuery?

Ans: As explained in previous question, using clone() method, we can create clone of any element but the default implementation of the clone() method doesn't copy events unless you tell the clone() method to copy the events. The clone() method takes a parameter, if you pass true then it will copy the events as well.

Hide   Copy Code

$(document).ready(function(){

$("#btnClone").bind('click', function(){

$('#dvClickme').clone(true).appendTo('body');

});

​

### Q46. What is difference between prop and attr?

Ans: attr(): Get the value of an attribute for the first element in the set of matched elements. Whereas,.prop(): (Introduced in jQuery 1.6) Get the value of a property for the first element in the set of matched elements.  
  
Attributes carry additional information about an HTML element and come in name="value" pairs. Where Property is a representation of an attribute in the HTML DOM tree. once the browser parse your HTML code ,corresponding DOM node will be created which is an object thus having properties.  
  
attr() gives you the value of element as it was defines in the html on page load. It is always recommended to use prop() to get values of elements which is modified via javascript/jquery , as it gives you the original value of an element's current state. Find out more [here](http://techbrij.com/jquery-attr-vs-prop-difference).

### Q47. What is event.PreventDefault?

Ans: The event.preventDefault() method stops the default action of an element from happening. For example, Prevents a link from following the URL.

### Q48. What is the difference between event.PreventDefault and event.stopPropagation?

Ans: event.preventDefault(): Stops the default action of an element from happening.  
event.stopPropagation(): Prevents the event from bubbling up the DOM tree, preventing any parent handlers from being notified of the event. For example, if there is a link with a click method attached inside of a DIV or FORM that also has a click method attached, it will prevent the DIV or FORM click method from firing.

### Q49. What is the difference between event.PreventDefault and "return false"?

Ans: e.preventDefault() will prevent the default event from occurring, e.stopPropagation() will prevent the event from bubbling up and return false will do both.

### Q50. What is the difference between event.stopPropagation and event.stopImmediatePropagation?

Ans: event.stopPropagation() allows other handlers on the same element to be executed, while event.stopImmediatePropagation() prevents every event from running. For example, see below jQuery code block.

Hide   Copy Code

$("p").click(function(event){

event.stopImmediatePropagation();

});

$("p").click(function(event){

*// This function won't be executed*

$(this).css("background-color", "#f00");

});

If event.stopPropagation was used in previous example, then the next click event on p element which changes the css will fire, but in case event.stopImmediatePropagation(), the next p click event will not fire.

### Q51. How to check if number is numeric while using jQuery 1.7+?

Ans: Using "<a href="http://jquerybyexample.blogspot.com/2011/11/jqueryisnumeric-in-jquery-17.html">isNumeric()</a>" function which was introduced with jQuery 1.7.

### Q52. How to check data type of any variable in jQuery?

Ans: Using [$.type(Object)](http://jquerybyexample.blogspot.in/2012/04/common-utility-methods-of-jquery.html) which returns the built-in JavaScript type for the object.

### Q53. How do you attach a event to element which should be executed only once?

Ans: Using jQuery one() method. This attaches a handler to an event for the element. The handler is executed at most once per element. In simple terms, the attached function will be called only once.

Hide   Copy Code

$(document).ready(function() {

$("#btnDummy").one("click", function() {

alert("This will be displayed only once.");

});

});​

### Q54. Can you include multiple version of jQuery? If yes, then how they are executed?

Ans: Yes. Multiple versions of jQuery can be included in same page.

### Q55. In what situation you would use multiple version of jQuery and how would you include them?

Ans: Well, it is quite possible that the jQuery plugins which are used are dependent on older version but for your own jQuery code, you would like to use newer version. So because of this dependency, multiple version of jQuery may required sometimes on single page.  
  
Below code shows how to include multiple version of jQuery.

Hide   Copy Code

<script type='text/javascript' src='js/jquery\_1.9.1.min.js'></script>

<script type='text/javascript'>

var $jq = jQuery.noConflict();

</script>

<script type='text/javascript' src='js/jquery\_1.7.2.min.js'></script>

By this way, for your own jQuery code use "$jq", instead of "$" as "$jq" refers to jQuery 1.9.1, where "$" refers to 1.7.2.

### Q56. Is it possible to hold or delay document.ready execution for sometime?

Ans: Yes, its possible. With Release of jQuery 1.6, a new method "jQuery.holdReady(hold)" was introduced. This method allows to delay the execution of document.ready() event. document.ready() event is called as soon as your DOM is ready but sometimes there is a situation when you want to load additional JavaScript or some plugins which you have referenced.

Hide   Copy Code

​

$.holdReady(true);

$.getScript("myplugin.js", function() {

$.holdReady(false);

});

### Q57. What is chaining in jQuery?

Ans: Chaining is one of the most powerful feature of jQuery. In jQuery, Chaining means to connect multiple functions, events on selectors. It makes your code short and easy to manage and it gives better performance. The chain starts from left to right. So left most will be called first and so on.

Hide   Copy Code

​$(document).ready(function(){

$('#dvContent').addClass('dummy');

$('#dvContent').css('color', 'red');

$('#dvContent').fadeIn('slow');

});​

The above jQuery code sample can be re-written using chaining. See below.

Hide   Copy Code

​$(document).ready(function(){

$('#dvContent').addClass('dummy')

.css('color', 'red')

.fadeIn('slow');

});​

Not only functions or methods, chaining also works with events in jQuery. Find out more [**here**](http://jquerybyexample.blogspot.com/2012/06/what-is-chaining-in-jquery.html).

### Q58. How does caching helps and how to use caching in jQuery?

Ans: Caching is an area which can give you awesome performance, if used properly and at the right place. While using jQuery, you should also think about caching. For example, if you are using any element in jQuery more than one time, then you must cache it. See below code.

Hide   Copy Code

$("#myID").css("color", "red");

*//Doing some other stuff......*

$("#myID").text("Error occurred!");

​

Now in above jQuery code, the element with #myID is used twice but without caching. So both the times jQuery had to traverse through DOM and get the element. But if you have saved this in a variable then you just need to reference the variable. So the better way would be,

Hide   Copy Code

var $myElement = $("#myID").css("color", "red");

*//Doing some other stuff......*

$myElement.text("Error occurred!");

​

So now in this case, jQuery won't need to traverse through the whole DOM tree when it is used second time. So in jQuery, Caching is like saving the jQuery selector in a variable. And using the variable reference when required instead of searching through DOM again.

### Q59. You get "jquery is not defined" or "$ is not defined" error. What could be the reason?

Ans: There could be many reasons for this.

* You have forgot to include the reference of jQuery library and trying to access jQuery.
* You have include the reference of the jQuery file, but it is after your jQuery code.
* The order of the scripts is not correct. For example, if you are using any jQuery plugin and you have placed the reference of the plugin js before the jQuery library then you will face this error.

Find out more [**here**](http://jquerybyexample.blogspot.com/2013/01/jquery-is-not-defined-error-reasons.html).

### Q60. How to write browser specific code using jQuery?

Ans: Using jQuery.browser property, we can write browser specific code. This property contains flags for the useragent, read from navigator.userAgent. This property was removed in jQuery 1.9.

### Q61. Can we use jQuery to make ajax request?

Ans: Yes. jQuery can be used for making ajax request.

### Q62. What are various methods to make ajax request in jQuery?

Ans: Using below jQuery methods, you can make ajax calls.

* load() : Load a piece of html into a container DOM
* $.getJSON(): Load JSON with GET method.
* $.getScript(): Load a JavaScript file.
* $.get(): Use to make a GET call and play extensively with the response.
* $.post(): Use to make a POST call and don't want to load the response to some container DOM.
* $.ajax(): Use this to do something on XHR failures, or to specify ajax options (e.g. cache: true) on the fly.

Find out more [**here**](http://net.tutsplus.com/tutorials/javascript-ajax/5-ways-to-make-ajax-calls-with-jquery/).

### Q63. Is there any advantage of using $.ajax() for ajax call against $.get() or $.post()?

Ans: By using jQuery post()/ jQuery get(), you always trust the response from the server and you believe it is going to be successful all the time. Well, it is certainly not a good idea to trust the response. As there can be n number of reason which may lead to failure of response.  
  
Where jQuery.ajax() is jQuery's low-level AJAX implementation. $.get and $.post are higher-level abstractions that are often easier to understand and use, but don't offer as much functionality (such as error callbacks). Find out more [**here**](http://jquerybyexample.blogspot.com/2011/11/avoid-jquerypost-use-jqueryajax.html).

### Q64. What are deferred and promise object in jQuery?

Ans: Deferred and promise are part of jQuery since version 1.5 and they help in handling asynchronous functions like Ajax. Find out more [**here**](http://www.bitstorm.org/weblog/2012-1/Deferred_and_promise_in_jQuery.html).

### Q65. Can we execute/run multiple Ajax request simultaneously in jQuery? If yes, then how?

Ans: Yes, it is possible to execute multiple Ajax request simultaneously or in parallel. Instead of waiting for first ajax request to complete and then issue the second request is time consuming. The better approach to speed up things would be to execute multiple ajax request simultaneously.  
  
Using jQuery .when() method which provides a way to execute callback functions based on one or more objects, usually Deferred objects that represent asynchronous events. Find out more [**here**](http://jquerybyexample.blogspot.com/2013/05/jquery-execute-multiple-ajax-request-simultaneously-in-parallel.html).

### Q66. Can you call C# code-behind method using jQuery? If yes,then how?

Ans: Yes. We can call C# code-behind function via $.ajax. But for do that it is compulsory to mark the method as WebMethod.

### Q67. Which is the latest version of jQuery library?

Ans: The latest version (when this post is written) of jQuery is 1.10.2 or 2.0.3. jQuery 2.x has the same API as jQuery 1.x, but does not support Internet Explorer 6, 7, or 8.

### Q68. Does jQuery 2.0 supports IE?

Ans: No. jQuery 2.0 has no support for IE 6, IE 7 and IE 8.

### Q69. What are source maps in jQuery?

Ans: In case of jQuery, Source Map is nothing but mapping of minified version of jQuery against the un-minified version. Source map allows to debug minified version of jQuery library. Source map feature was release with jQuery 1.9. Find out more [**here**](http://jquerybyexample.blogspot.com/2013/01/all-you-need-to-know-about-jquery-source-maps.html).

### Q70. How to use migrate jQuery plugin?

Ans: with release of 1.9 version of jQuery, many deprecated methods were discarded and they are no longer available. But there are many sites in production which are still using these deprecated features and it's not possible to replace them overnight. So jQuery team provided with jQuery Migrate plugin that makes code written prior to 1.9 work with it.  
  
So to use old/deprecated features, all you need to do is to provide reference of jQuery Migrate Plugin. Find out more [**here**](http://jquerybyexample.blogspot.com/2013/01/migrate-older-jquery-code-to-jquery1-9.html).

### Q71. Is it possible to get value of multiple CSS properties in single statement?

Ans: Well, before jQuery 1.9 release it was not possible but one of the new feature of jQuery 1.9 was .css()multi-property getter.

Hide   Copy Code

var propCollection = $("#dvBox").css([ "width", "height", "backgroundColor" ]);

In this case, the propCollection will be an array and it will look something like this.

Hide   Copy Code

{

width: "100px",

height: "200px",

backgroundColor: "#FF00FF"

}

### Q72. How do you stop the currently-running animation, remove all queued animations, and complete all animations for the matched elements?

Ans: It can be done via calling .stop([clearQueue ] [, jumpToEnd ]) method and by passing both the parameters as true.

### Q73. What is finish method in jQuery?

Ans: The .finish() method stops all queued animations and places the element(s) in their final state. This method was introduced in jQuery 1.9.

### Q74. What is the difference between calling stop(true,true) and finish method?

Ans: The .finish() method is similar to .stop(true, true) in that it clears the queue and the current animation jumps to its end value. It differs, however, in that .finish() also causes the CSS property of all queued animations to jump to their end values, as well.

### Q75. Consider a scenario where things can be done easily with javascript, would you still prefer jQuery?

Ans: No. If things can be done easily via CSS or JavaScript then You should not think about jQuery. Remember, jQuery library always comes with xx kilobyte size and there is no point of wasting bandwidth.

### Q76. Can we use protocol less URL while referencing jQuery from CDNs?

Ans: Yes. Below code is completely valid.

Hide   Copy Code

<script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>

### Q77. What is the advantage of using protocol less URL while referencing jQuery from CDNs?

Ans: It is quite useful when you are moving from HTTP to HTTPS url. You need to make sure that correct protocol is used for referencing jQuery library as pages served via SSL should contain no references to content served through unencrypted connections.  
  
"protocol-less" URL is the best way to reference third party content that’s available via both HTTP and HTTPS. When a URL’s protocol is omitted, the browser uses the underlying document’s protocol instead. Find out more [**here**](http://jquerybyexample.blogspot.com/2013/06/use-protocol-less-url-referencing-jquery.html).

### Q78. What is jQuery plugin and what is the advantage of using plugin?

Ans: A plug-in is piece of code written in a standard JavaScript file. These files provide useful jQuery methods which can be used along with jQuery library methods. jQuery plugins are quite useful as its piece of code which is already written by someone and re-usable, which saves your development time.

### Q79. What is jQuery UI?

Ans: jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library that can be used to build interactive web applications.

### Q80. What is the difference between jQuery and jQuery UI?

Ans: jQuery is the core library. jQueryUI is built on top of it. If you use jQueryUI, you must also include jQuery.