Q1. What is jQuery?

Ans: jQuery is **fast, lightweight and feature-rich** client side JavaScript Library/Framework which helps in to traverse HTML DOM, make animations, add Ajax interaction, manipulate the page content, change the style and provide cool UI effect. It is one of the most popular client side library and as per a survey it runs on every second website.

Q2. Why do we use jQuery?

Ans: Due to following advantages.

- Easy to use and learn.
- Easily expandable.
- Cross-browser support (IE 6.0+, FF 1.5+, Safari 2.0+, Opera 9.0+)
- Easy to use for DOM manipulation and traversal.
- Large pool of built in methods.
- AJAX Capabilities.
- Methods for changing or applying CSS, creating animations.
- Event detection and handling.
- Tons of plug-ins for all kind of needs.

Q3. How JavaScript and jQuery are different?

Ans: JavaScript is a language While jQuery is a library built in the JavaScript language that helps to use the JavaScript language.

Q4. Is jQuery replacement of Java Script?

Ans: **No.** jQuery is not a replacement of JavaScript. jQuery is a different library which is written on top of JavaScript. jQuery is a lightweight JavaScript library that emphasizes interaction between JavaScript and HTML.

Q5. Is jQuery a library for client scripting or server scripting?

Ans. Client side scripting.

Q6. Is jQuery a W3C standard?

Ans: No. jQuery is not a W3C standard.

Q7. What is the basic need to start with jQuery?

Ans: To start with jQuery, one need to make reference of it's library. The latest version of jQuery can be downloaded from jQuery.com.

Q8. Which is the starting point of code execution in jQuery?

Ans: The starting point of jQuery code execution is $(document).ready() function which is executed when DOM is loaded.

Q9. 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.
⊟Collapse | Copy Code

```
$(document).ready(function(){
});
```

Over here $ sign can be replaced with "jQuery" keyword.
⊟Collapse | Copy Code

```
jQuery(document).ready(function(){
});
```

Q10. 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.

Q11. Can we use our own specific character in the place of $ sign in jQuery?

Ans: **Yes.** It is possible using jQuery.noConflict().

Q12. Is it possible to use other client side libraries like MooTools, Prototype along with jQuery?

Ans: **Yes.**

Q13. What is jQuery.noConflict?

Ans: As other client side libraries like MooTools, Prototype can be used with jQuery and they also use $() as their global function and to define variables. This situation creates conflict as `$()` is used by jQuery and other library as their global function. To overcome from such situations, jQuery has introduced `jQuery.noConflict()`.
⊟Collapse | Copy Code

```
jQuery.noConflict();
// Use jQuery via jQuery(...)
jQuery(document).ready(function(){
   jQuery("div").hide();
});
```

You can also use your own specific character in the place of $ sign in jQuery.
⊟Collapse | Copy Code

```
var $j = jQuery.noConflict();
```

```
// Use jQuery via jQuery(...)
$j(document).ready(function(){
   $j("div").hide();
});
```

Q14. Is there any difference between body onload() and document.ready() function?

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

1. We can have more than one `document.ready()` function in a page where we can have only one body `onload` function.
2. `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.

Q15. What is the difference between .js and .min.js?

Ans: jQuery library comes in 2 different versions Production and 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.

Q16. Why there are two different version of jQuery library?

Ans: jQuery library comes in 2 different versions.

1. Production
2. Deployment

The production 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 production 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

⊟Collapse | 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

⊟Collapse | 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)

⊟Collapse | 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.

⊟Collapse | Copy Code

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></scrip
t>
<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,

⊟Collapse | 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".

⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | Copy Code

```
$(document).ready(function(){
    if ($('#element').is(':empty')){
       //Element is empty
  }
});
```

And the second way is using the "$.trim()" method.
⊟Collapse | 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.
⊟Collapse | 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**.

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:
⊟Collapse | 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,
⊟Collapse | 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**

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.

`.bind()`: 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.

`.live()`: 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.

`.delegate()`: 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**

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,
⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | 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.

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.

⊟Collapse | 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 "`isNumeric()`" function which was introduced with jQuery 1.7.

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

Ans: Using $.type(Object) 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.

⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | 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.

⊟Collapse | 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**.

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.

⊟Collapse | 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,

⊟Collapse | 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**.

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**.

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**.

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**.

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**.

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**.

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**.

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.

⊟Collapse | 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.

⊟Collapse | 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.
⊟Collapse | 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**.

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.