

## 1. What is jQuery?

- jQuery is an open source and most popular library for simplifying interactions between DOM and JavaScript.

## 2. What are the advantages of jQuery?

- Most popular and open source
- Very fast and easily extensible
- Used to develop cross browser compatible web applications as jQuery works in almost the same manner for different kinds of browsers.
- Improves the performance of an application when using the minimised version of the jQuery library. The size of the minimized js file is almost 50% less than the normal js file. Reduction in the file size makes the web page load and work faster.
- Commonly implemented UI related functionalities are written using minimal lines of codes

## 3. Is jQuery a JavaScript or JSON library file?

- jQuery is said to be a library of single JavaScript file which consists of DOM/CSS manipulations, event effects or animations, AJAX functions and various commonly used plugins.

## 4. Does jQuery work for both HTML and XML documents?

- No. jQuery works only for HTML documents.

## 5. What are the jQuery functions used to provide effects?

Some of the methods are listed below which provide the effect:

### 1. **toggle()** :

- This function is used to check the visibility of selected elements to **toggle** between `hide()` and `show()` for the selected elements where:
  - `show()` is run when the element is hidden.
  - `hide()` is run when the element is visible.
- **Syntax:** `$(selector).toggle(speed, easing, callback)`

### 2. **slideDown()** :

- This function is used to either check the visibility of selected elements or to show the hidden elements. We can use this function on the following types of hidden elements:
  - Elements that are hidden using jQuery methods.
  - Elements that are hidden using `display: none` in the element's CSS properties.
- **Syntax:** `$(selector).slideDown( speed, easing, callback )`

### 3. **fadeOut()**:

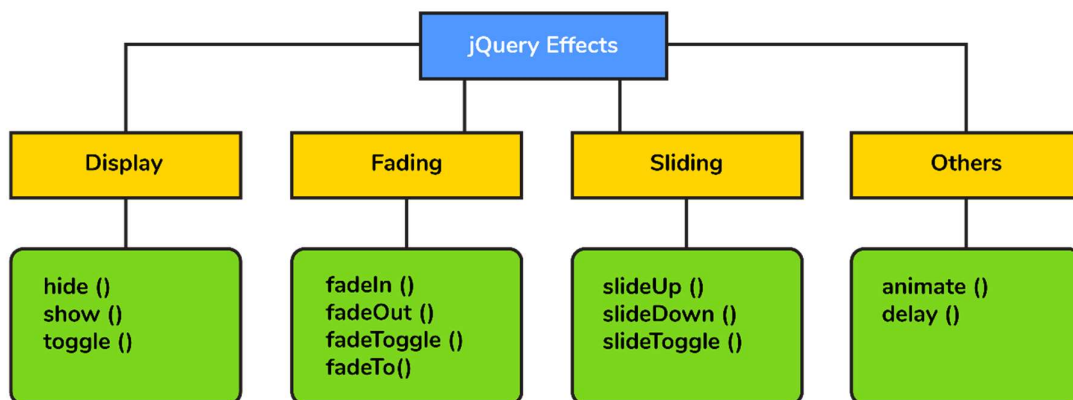
- This function is used to **change the level of opacity** for element of choice from visible to hidden. When used, the faded element will not occupy any space in DOM.
- **Syntax:** `$(selector).fadeOut( speed, easing, callback )`

#### 4. **fadeToggle():**

- This is used for toggling between the `fadeIn()` and `fadeOut()` methods.
  - If elements are faded in state, `fadeToggle()` will fade out those elements.
  - If elements are faded out, `fadeToggle()` will fade in those elements.
- **Syntax:** `$(selector).fadeToggle(speed, easing, callback)`

#### 5. **animate():**

- The method performs custom animation of a set of CSS properties. 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:** `(selector).animate({styles},speed,easing,callback)` where "styles" is a required field that specifies one or more CSS properties/values to animate. The properties needs to be mentioned in camel casing style.
- The parameters "speed", "easing" and "callback" in the syntaxes of the above methods represent:
  - **speed:** Optional parameter and used for specifying the speed of the effect. The default value is 400 millisecond. The possible value of speed are "slow", "fast" or some number in milliseconds.
  - **easing:** Again optional parameter which is used for specifying the speed of element to different types of animation. The default value is "swing". The possible value of easing are "swing" and "linear".
  - **callback:** Optional parameter. The callback function specified here is executed **after** the effect method is completed.



#### 6. What is the use of `css()` method in jQuery?

- The `css()` method is used to **change style property** of the selected element.

#### 7. What are events in jQuery?

- User actions on a webpage are called events and handling responses to those is called event handling. jQuery provides simple methods for attaching event handlers to selected elements. When an event occurs, the provided function is executed.

#### 8. What is the significance of jQuery.length?

- `jQuery.length` property is used to **count number of the elements** of the jQuery object.

#### 9. What is jQuery click event?

- jQuery click event happens when we click on an HTML element.
- jQuery provides a method `click()` method that aids to trigger the click event.
- For example `$(“p”).click()` will trigger the click event whenever the elements with paragraph tag is clicked on a browser page.

- **Syntax:**

```
$(selector).click(function(){
    //code that runs when the click event is triggered
});
```

#### 10. Can you tell something about jQuery each() method?

- The `each()` method in jQuery allows us to loop through different datasets such as arrays or objects (even DOM objects).
- It can be used to loop through a number of DOM objects from the same selectors.
- For example, if you want to add a `width="600"` to all the images in a page then we select all images and loop through each of them and add `width = "600"` to each tag. We can write the code as below:

```
$(“img”).each(function(im){
    $(this).attr(“width”, “600”)
});
```

- `$` is a jQuery object definer. In the above syntax, “this” is a DOM object and we can apply jQuery functions to only jQuery objects which is why we convert the DOM object to jQuery object by wrapping it inside the `$` definer.
- We can also use `each()` to loop through the arrays of data and get the index and the value of the position of data inside the array.
- For example,

```
var list = ["InterviewBit", "jQuery", "Questions"];
$.each(list, function(index, value){
    console.log(index + " " + value);
})
```

- The above code prints

```
0 InterviewBit
1 jQuery
2 Questions
```

- You can also use `each()` to loop through objects.

- For example:

```
var obj = {"name":"InterviewBit","type": "jQuery"};
$.each(obj, function(key,value){
    console.log(key + " - " + value);
})
```

- The above code prints:

```
name - InterviewBit
type - jQuery
```

## 11. What is the difference between javascript and jquery?

- JavaScript is an interpreted language written in C and is combination of ECMAScript and DOM where jQuery is a JavaScript library developed to run things faster and make things simplified for JavaScript. jQuery doesn't have the ECMAScript.
- JavaScript requires long lines of code to code a functionality where in case of jQuery, just import the library and call the functions which would reduce the programmer's effort of coding.
- JavaScript doesn't have the cross browser compatible functionality which is why a developer has to write code manually to implement the functionality. Whereas the cross browser code compatibility is inbuilt in jQuery.

## 12. What are the selectors in jQuery? How many types of selectors in jQuery?

In order to work with any element on the web page, we would first need to find it. Selectors find the HTML elements in jQuery.

Some of the most commonly used and basic selectors are:

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

## 13. Explain how CSS classes can be manipulated in HTML using jQuery.

- Query provides several methods to manipulate the CSS classes assigned to HTML elements. The most important methods are `addClass()`, `removeClass()` and `toggleClass()`.
- **addClass():** This method adds one or more classes to the selected elements.
  - Syntax: `$(selector).addClass(className);`
  - You can also add multiple classes to the selector. Syntax: `$(selector).addClass(class1, class2);`
- **removeClass():** Similar to adding class, you can also remove the classes from the elements by using this method.
  - The `removeClass()` method can remove a single class, multiple classes, or all classes at once from the selected elements.

- Syntax:
  - For removing one class: `$(selector).removeClass(class1);`
  - For removing multiple class: `$(selector).removeClass(class1, class2, class 3);`
  - For removing all classes at once: `$(selector).removeClass();`
- **toggleClass():** This method is used for adding or removing one or more classes from the selected elements in such a way that if the selected element already has the class, then it is removed. Else if an element does not have the specified class, then it is added i.e. it toggles the application of classes.
  - Syntax: `$(selector).toggleClass(className);`

#### Jquery Coding Problems:

#### 14. How to perform jQuery AJAX requests?

- jQuery provides the `ajax()` method to perform an AJAX (asynchronous HTTP) request.
- Syntax: `$.ajax({name:value, name:value, ... })`. The parameters specify one or more value of name-value pairs.
  - `url` : this name specifies the URL to send the request to. Default is the current page.
  - `success(result,status,xhr)` : success callback function which runs when the request succeeds
  - `error(xhr,status,error)` : A function to run if the request fails.
  - `async` : Boolean value that indicates whether the request should be handled asynchronous or not. Default value is true.
  - `complete(xhr,status)` : A function to run when the request is completed (after success and error functions are handled)
  - `xhr` : A function used for creating the `XMLHttpRequest` object
- Example:

```
$.ajax({
  url: "resourceURL",
  async: false,
  success: function(result){
    $("div").html(result);
  },
  error: function(xhr){
    alert("An error occurred: " + xhr.status + " " + xhr.statusText);
  }
});
```

#### 15. What does the following code do?

```
$( "div#firstDiv, div.firstDiv, ol#items > [name$='firstDiv']" )
```

- The given code is an example of getting elements that satisfy multiple selectors at once. The function returns a jQuery object having the results of the query.

- The given code does a query to retrieve those `<div>` element with the id value `firstDiv` along with all `<div>` elements that has the class value `firstDiv` and all elements that are children of the `<ol id="items">` element and whose name attribute ends with the string `firstDiv`.

16. Consider the following code that exists in following HTML with the CSS:

```
<
<div id="expand"></div>

<style>
div#expand{
width: 50px;
height: 50px;
background-color: gray;
}
</style>
```

- Write jQuery code to animate the `#expand` div, expanding it from 50 \* 50 pixels to 300 \* 300 pixels within five seconds.
- We can do this by using the `animate()` function. We first need to have access to the div element which has id value of `expand` and then apply `animate` function on the element as follows:

```
$("#expand").animate(
{
width: "300px",
height: "300px",
},
5000
);
```

17. What does the following code do?

```
$( "div" ).css( "width", "500px" )
.add( "p" )
.css( "background-color", "yellow" );
```

- The given code first selects all the `<div>` elements and applies width of 500px to them and adds all the `<p>` elements to the elements selection after which the code can finally change the background color to yellow for all the `<div>` and `<p>` elements
- The given code is an example of method chaining in jQuery which is used to accomplish a couple of things in one single instruction.

18. Can you explain the difference between `jQuery.get()` and `jQuery.ajax()`?

- `jQuery.ajax()` allows the creation of highly-customized AJAX requests, with options for how long to wait for a response, what to do once the request is successful, how to handle a failure scenarios, whether the request to be sent is blocking (synchronous) or non-blocking (asynchronous), what format to expect as the response, and many more customizable options.

- `jQuery.get()` is uses `jQuery.ajax()` underneath to create an AJAX request typically meant for simple retrieval of information.
  - There are various other pre-built AJAX requests given by jQuery such as:
    - `jQuery.post()` for performing post requests
    - `jQuery.getScript()` meant for loading and then executing a JavaScript file from the server using GET request.
    - `jQuerygetJSON()` for loading JSON-encoded data from the server using a GET HTTP request.

**19. Which of the two lines of code below is more efficient and why? `document.getElementById("interviewBit");` OR `$("#interviewBit");`**

- The code `document.getElementById("interviewBit");` is more efficient because its the pure JavaScript version.
- The reason being jQuery is built on top of JavaScript and internally uses its methods to make DOM manipulation easier. This introduces some performance overhead. We need to remember that jQuery is not always better than pure JavaScript and we need to use it only if it adds advantage to our project.

**20. Can you write a jQuery code selector that needs to be used for querying all elements whose ID ends with string "IB"?**

- We can use the following selector: `$("[id$='IB']")`

**21. Explain the `.promise()` method in jQuery.**

- The `.promise()` method returns a dynamically generated promise that is resolved when **all actions** of a certain type bound to the collection, queued or not, **have ended**.
- The method takes two optional arguments:
  - **type** - The default type is "fx" which indicates that the returned promise is resolved only when all animations of the selected elements have been completed.
  - **target** - If a target object is specified, `.promise()` will attach to promise to that specified object and then return it rather than creating a new one.

**22. Consider the below code snippet and assume that there are 5 <div> elements on the page. What is the difference between the start and end times displayed?**

```
function getMinsSeconds() {
    var dt = new Date();
    return dt.getMinutes()+":"+dt.getSeconds();
}

$( "input" ).on( "click", function() {
    $( "p" ).append( "Start: " + getMinsSeconds() + "<br />" );
    $( "div" ).each(function( i ) {
        $( this ).fadeOut( 1000 * ( i * 2 ) );
    });
    $( "div" ).promise().done(function() {
        $( "p" ).append( "End: " + getMinsSeconds() + "<br />" );
    });
});
```

- For the above code, the difference between the start and end times will be **10 seconds**.

- This is because `.promise()` will wait for all `<div>` animations (here, all `fadeOut()` calls) to complete, the last one will complete 10 seconds (i.e.  $5 * 2 = 10$  seconds) after the `fadeOut()` starts.

**23. Can you tell the difference between `prop()` and `attr()`s?**

- Both `prop()` and `attr()` can be used to get or set the value of the specified property of an element attribute.
- The `attr()` gives the **default value** of a property whereas `prop()` returns its **current value**.

### Practice jQuery Questions asked in Interviews

---

**Q1 - Which of the following is a single global function defined in the jQuery library?**

- ☒ `jQuery()`
- ☐ `$()`
- ☐ `Queryanalysis()`
- ☐ `global()`

**Explanation:** jQuery function is so frequently used that the library also defines the global symbol `$` as a shortcut for it. The `$` sign it's just an alias to `jQuery()`, then an alias to a function which is used as a selector element.

**Q2 - Which is code asks for the set of all div elements in a document?**

- ☐ `var divs = $(div);`
- ☒ `var divs = jQuery("div");`
- ☐ `var divs = $("div");`
- ☐ `var divs = #("div");`

**Explanation:** `$` is often used by other libraries or JavaScript constructs, so it could be overwritten and may not do what you expect. jQuery always refers to the jQuery library, so it will do what you expect. When there are no other libraries, jQuery and `$` mean the same thing.

In environments where you cannot guarantee that `$` will not be overwritten by something else, using `jQuery` or `jQuery.noConflict()` is safer than `$`.

**Q3 - Which method operates on the return value of `$()`?**

- ☐ `show()`
- ☐ `css()`
- ☒ `click()`
- ☐ `each()`

**Q4 - Which method is used for parsing JSON text?**



- ☐ jQuery.each()
- ☒ jQuery.parseJSON()
- ☐ jQuery.noConflict()
- ☐ jQuery.done()

**Q5 - Select the correct jQuery code to set the background color of all div elements to red?**

- ☒ \$("div").css("background-color","red");
- ☐ \$("div").style("background-color","red");
- ☐ \$("div").layout("background-color","red");
- ☐ \$("div").manipulate("background-color","red");

**Q6 - What does the following selector select? \$("span.first").**

- ☒ All span elements with class="first"
- ☐ The first span element with class="first"
- ☐ The first span element with id="first"
- ☐ All span elements with id="first"

**Explanation:** Only the class name is confusing, apart from that the selector is straight forward one.

**Q7 - Which method is used to perform an asynchronous HTTP request?**

- ☐ jQuery.asyncAjax()
- ☐ jQuery.ajaxSetup()
- ☐ jQuery.ajaxAsync()
- ☒ jQuery.ajax()

**Q8 - Select the correct jQuery code for making all div elements 100 px high?**

- ☐ \$("<div>").height(100)
- ☐ \$("div").yPos(100)
- ☐ \$("<div>").height="100"
- ☒ \$("div").height(100)

**Q9 - Which method is used to switch between adding/removing one or more classes (for CSS) from selected elements?**

- ☒ toggleClass()
- ☐ switch()
- ☐ switchClass()

☐ altClass()

Q10 - What does the following selector select \$("div p")?

- ☐ The first p element inside a div element
- ☐ All div elements with a p element
- ☒ All p elements inside a div element
- ☐ None of the above

Q11 - What does the following selector select \$(" :disabled")?

- ☐ All hidden elements
- ☐ All elements that does not contain the text "disabled"
- ☒ All disabled input elements
- ☐ All elements containing the text "disabled"

Explanation: :disabled represents the disabled state of any element.

Q12 - Can the animate() method be used to animate ANY CSS property?

- ☐ Yes
- ☐ No
- ☒ Only properties containing numeric values
- ☐ All properties except the shorthand properties

Q13 - \$.foobar() is equivalent to which of the following below??

- ☐ javascript.foobar()
- ☐ document.foobar()
- ☒ jQuery.foobar()
- ☐ None of the above

Q14 - What does the following lines of code do?

```
$(document).ready(function() {  
    // Some code.  
});
```

- ☐ Make sure no code is executed till the entire page is fully loaded
- ☒ Make sure no code is executed till the DOM is fully loaded
- ☐ Both A and B
- ☐ Neither A nor B

Q15 - Query.noConflict(true) is used to for?

- ☐ Freeing up the \$ symbol for use by other libraries
- ☐ Improving compatibility
- ☐ Removing all jQuery variables from the global scope
- ☒ All of the above

Q16 - What is the phenomenon used in the below code?

```
$('#mainList').find('li')  
    .width(1000).addClass('selectedElement');
```

☐ Toggling

☐ Event bubbling

☒ Chaining

☐ Animating

**Q17 - Select the correct options relating to the `:odd` and `:even` filters?**

☐ They allow you to determine if a number is odd or even.

☒ They allow you to determine if a specific element is in an odd or even position.

☐ They allow to remove the odd or even elements appropriately.

☐ None of the above

**Q18 - Identify the difference between `.width()` and `.outerWidth()`?**

☐ The methods are basically the same. The only difference is that `.width()` returns a number, whereas `outerWidth()` a string.

☐ No difference. `width()` is a shorthand alias for `outerWidth()`

☒ `width()` returns the computed width of the element, while `outerWidth()` returns the width plus all the margins and paddings.

☐ There is no such method called `outerWidth()`.

**Q19 - Which of the following code fetches the first span on the page, which has the class 'grey'?**

☐ `$('#span, .grey, :first')`

☐ `$('#first .grey span')`

☒ `$('#span.grey:first')`

☐ None of the above

**Q20 - What function is used to stop your jQuery for a few milliseconds?**

☐ `stop()`

☒ `delay()`

☐ `slowdown()`

☐ `pause()`