**#Jquery interview Questions**

**1)** **Explain what the following code will do**:

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

This code performs a query to retrieve any <div> element with the id first, plus all <div> elements with the class first, plusall elements which are children of the <ol id="items"> element and whose name attribute ends with the string "first". This is an example of using multiple selectors at once. The function will return a jQuery object containing the results of the query.

**2)** **What is the difference between jQuery.get() and jQuery.ajax()?**

jQuery.ajax() is the all-encompassing Ajax request method provided by jQuery. It allows for the creation of highly-customized Ajax requests, with options for how long to wait for a response, how to handle a failure, whether the request is blocking (synchronous) or non-blocking (asynchronous), what format to request for the response, and many more options.

jQuery.get() is a shortcut method that uses jQuery.ajax() under the hood, to create an Ajax request that is typical for simple retrieval of information. Other pre-built Ajax requests are provided by jQuery, such as jQuery.post(), jQuery.getScript(), and jQuery.getJSON().

**3)Which of two lines of code is more efficient an &explain?**

**document.getElementById(“logo”); or**

**$(“#logo”);**

The first line of code, which is pure JavaScript without jQuery, is more efficient and faster. Executing the second line of code, which is jQuery, will trigger a call to the JavaScript version.

jQuery is built on top of JavaScript and uses its methods under the hood to make DOM manipulation easier, at the cost of some performance overhead. It is a good idea to remember that jQuery is not always better than plain old JavaScript. Always consider whether using jQuery really provides a useful advantage for your project.

**4) Explain the difference between the .detach() and .remove() methods in jQuery.**

The [.detach()](https://api.jquery.com/detach/) and [.remove()](https://api.jquery.com/remove/) methods are the same, except that .detach() retains all jQuery data associated with the removed elements and .remove() does not. .detach() is therefore useful when removed elements may need to be reinserted into the DOM at a later time.

## 5) Explain width() vs css(‘width’).

1. In jQuery, there are two way to change the width of an element.  
2. One way is using .css(‘width’) and other way is using .width().

**For example**

$(‘#mydiv’).css(‘width’,’300px’);  
$(‘#mydiv’).width(100);

## 6)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.

## 7) 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

## 8) 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.

## 9)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.

## 10) 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.