1. Why JQuery?

Challenges in Traditional JavaScript

Cross-browser Compatibility: Traditional JavaScript code may behave differently across various browsers, leading to compatibility issues.

Lengthy Code: Writing JavaScript code for common tasks, such as DOM manipulation, can be verbose and time-consuming.

Complexity: Handling asynchronous tasks and events can become complex and difficult to manage

Benefits of Using JQuery

Simplified Code: JQuery offers a concise syntax for common tasks, reducing the amount of code needed compared to traditional JavaScript.

Cross-browser Compatibility: JQuery abstracts away browser-specific implementations, ensuring consistent behavior across different browsers

Ease of Use: With its intuitive API, JQuery is easy to learn and use, making it accessible for developers at various skill levels.

Ajax Simplification: JQuery simplifies the implementation of AJAX, making asynchronous communication with the server straightforward

2. What is jQuery?

JQuery is a lightweight, "write less, do more", and JavaScript library.

The purpose of jQuery is to make it much easier to use JavaScript on your website.

JQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

JQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities

2.1 Advantages of jQuery

- **Easy to learn:** jQuery is easy to learn because it supports same JavaScript style coding.
- **Write less do more:** jQuery provides a rich set of features that increase developers' productivity by writing less and readable code.
- **Excellent API Documentation:** jQuery provides excellent online API documentation.
- **Cross-browser support:** jQuery provides excellent cross-browser support without writing extra code.

3. How to use jQuery?

There are two ways to use jQuery.

- **Local Installation** − you can download jQuery library on your local machine and include it in your HTML code.
- **CDN Based Version** − you can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

4. JQuery Important Features

- **DOM Selection:** jQuery provides Selectors to retrieve DOM element based on different criteria like tag name, id, css class name, attribute name, value, nth child in hierarchy etc.
- **DOM Manipulation:** You can manipulate DOM elements using various built-in jQuery functions. For example, adding or removing elements, modifying html content, css class etc.
- **Special Effects:** You can apply special effects to DOM elements like show or hide elements, fade-in or fade-out of visibility, sliding effect, animation etc.
- **Events:** jQuery library includes functions which are equivalent to DOM events like click, dblclick, mouseenter, mouseleave, blur, keyup, keydown etc. These functions automatically handle cross-browser issues.
- **Ajax**: jQuery also includes easy to use AJAX functions to load data from servers without reloading whole page.
- **Cross-browser support:** jQuery library automatically handles cross-browser issues, so the user does not have to worry about it. jQuery supports IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+.

5. JQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is: **$(*selector*).*action* ()**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

```
$(document).ready(function(){
  // jQuery methods go here...
});
```

This is to prevent any jQuery code from running before the document is finished loading (is ready).

It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.

Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){
  // jQuery methods go here...
});
```

DOM Manipulation with JQuery

jQuery simplifies DOM manipulation and event handling in JavaScript.

To use jQuery, you need to include the jQuery library in your HTML file. You can either download it and host it locally or use a CDN.

Selecting Elements:

```
// Selecting elements by tag name
 $('p').text('This is a paragraph.');
// Selecting elements by class
 $('.myClass').css('color', 'blue');
// Selecting elements by ID
 $('#myId').html('Content changed.');
```

Modifying Elements:

```
// Changing text content
$('p').text('New text content.');
// Appending content
$('#myDiv').append('<p>Appended paragraph.</p>');
// Removing an element
$('.removeMe').remove();
```

Event Handling:

```
// Click event

$('#myButton').click(function() {
    alert('Button clicked!');
});
```

```
// Hover event

<script>
    $(document).ready(function() {
        $('.hover-element').hover(
            function() {
                // Mouse over - change background color
                $(this).css('background-color', 'red');
            },
            function() {
             // Mouse out - revert to initial background color
                $(this).css('background-color', 'initial');
            }
        );
    });
</script>
```

Animations:

```
// Hide and show with animation

$('#myElement').hide(1000).show(1000);

// Fade in and out with animation

$('#myElement').fadeOut(1000).fadeIn(1000);
```

Increase image's visibility with Fade In

```
// Slide up and down with animation

$('#myElement').slideUp(1000).slideDown(1000);
```

In android, Slide Up and Slide Down animations are used to change the appearance and behavior of the objects over a particular interval of time.