# Module (JQuery Basic, Effects & Advanced) – 5

**Que.1 What is jQuery?**

**Ans**. jQuery is a fast, small, and feature-rich JavaScript library. It simplifies HTML document traversal, event handling, animating, and Ajax interactions for rapid web development. jQuery is designed to make it easier to navigate and manipulate HTML documents, handle events, create animations, and perform asynchronous requests to the server. It is widely used by developers to enhance the functionality and interactivity of websites.

**Que.2 How to Apply CSS Using JQuery, How to Add Class and Remove Class in Jquery , JQuery Animation?**

**Ans**. To apply CSS using jQuery, you can use the `.css()` method. This method allows you to modify the CSS properties of an element. Here's an example:

**javascript**

**// Applying CSS using jQuery**

**$(selector).css(property, value);**

In the above code, `selector` represents the element(s) you want to apply the CSS to. `property` is the CSS property you want to modify, and `value` is the new value for that property.  
  
To add a class to an element using jQuery, you can use the `.addClass()` method. Here's an example:

**javascript**

**// Adding a class using jQuery**

**$(selector).addClass(className);**

In the above code, `selector` represents the element(s) you want to add the class to, and `className` is the name of the class you want to add.  
  
To remove a class from an element using jQuery, you can use the `.removeClass()` method. Here's an example:

**javascript**

**// Removing a class using jQuery**

**$(selector).removeClass(className);**

In the above code, `selector` represents the element(s) you want to remove the class from, and `className` is the name of the class you want to remove.  
  
Regarding jQuery animation, jQuery provides various methods to create animations. The most commonly used method is `.animate()`, which allows you to animate CSS properties. Here's an example:

**javascript**

**// jQuery animation using .animate()**

**$(selector).animate({ property: value }, duration, easing, complete);**

In the above code, `selector` represents the element(s) you want to animate, `property` is the CSS property you want to animate, `value` is the target value for that property, `duration` is the duration of the animation in milliseconds, `easing` is the easing function to control the animation's acceleration, and `complete` is a callback function that executes after the animation is complete.  
  
These are some basic examples of how to apply CSS, add classes, remove classes, and perform animations using jQuery.

**Que.3 How to create slider with animation?**

**Ans**. To create a slider with animation using jQuery, you can follow these general steps:  
  
**1. HTML Structure:** Create the HTML structure for your slider. Typically, this involves a container element that holds the slide items.

**html**

**<div class="slider">**

**<div class="slide">Slide 1</div>**

**<div class="slide">Slide 2</div>**

**<div class="slide">Slide 3</div>**

**</div>**

**2.CSS Styling:** apply css to position and style the slider and slide items as desired.

**css**

**.slider {**

**width: 100%;**

**height: 300px;**

**overflow: hidden;**

**}**

**.slide {**

**width: 100%;**

**height: 100%;**

**display: none;**

**}**

**3. jQuery animation:** use jquery to animate the slide transitions. Here an example using the .animate() method.

**javascript**

**$(document).ready(function() {**

**// Set initial slide**

**$('.slide:first').addClass('active').show();**

**// Slide animation function**

**function slideAnimation() {**

**var currentSlide = $('.active');**

**var nextSlide = currentSlide.next('.slide');**

**// If there is no next slide, loop back to the first slide**

**if (nextSlide.length === 0) {**

**nextSlide = $('.slide:first');**

**}**

**// Animate the slide transition**

**currentSlide.removeClass('active').fadeOut();**

**nextSlide.addClass('active').fadeIn();**

**}**

**// Set interval for automatic slide transition**

**setInterval(slideAnimation, 3000);**

**});**

In the above code, the `slideAnimation()` function is responsible for transitioning between slides. It selects the current active slide, finds the next slide, and animates the transition by fading out the current slide and fading in the next slide. The `setInterval()` function is used to repeatedly call the `slideAnimation()` function at a specified interval (in this case, every 3 seconds).  
  
Remember to adjust the CSS and jQuery code according to your specific slider design and requirements.

**Que.4 Event bubbling tickling example**

**Ans**. Event bubbling is a concept in JavaScript where an event triggered on a nested element will also trigger the same event on its parent elements, propagating up the DOM tree. To illustrate event bubbling, let's consider a tickling example:

**html**

**<div id="grandparent">**

**<div id="parent">**

**<div id="child">**

**Tickle me!**

**</div>**

**</div>**

**</div>**

**Java script using jquery.**

**javascript**

**// Event bubbling tickling example**

**$("#grandparent").on("click", function() {**

**console.log("Grandparent tickled!");**

**});**

**$("#parent").on("click", function() {**

**console.log("Parent tickled!");**

**});**

**$("#child").on("click", function() {**

**console.log("Child tickled!");**

**});**

In the above code, we have three nested elements: grandparent, parent, and child. Each element has a click event handler attached to it. When you click on the "Tickle me!" text inside the child element, the click event will bubble up through the parent element and eventually reach the grandparent element.  
  
As a result, the console will display the following output:

**Child tickled!**

**Parent tickled!**

**Grandparent tickled!**

The demonstrates how the click event bubbles up from the child element to its parent elements due to event bubbling.