

Module (JQuery Basic, Effects & Advanced) – 5

VIJAY RATNOTAR



June 28, 2023

TOPS TECHNOLOGIES

MODULE (JQUERY BASIC, EFFECTS & ADVANCED) – 5 ( VijayRatnotar)

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

**Name:- Vijay Ratnotar Mo.No.9429099686**

1. **What is jQuery?**

jQuery is a lightweight, feature-rich JavaScript library. It makes HTML document navigation, event handling, animation, and interaction with the Document Object Model (DOM) easier. jQuery was created to make web page navigation and manipulation simpler by offering a succinct and efficient syntax.

Here are some of jQuery's primary features and functionalities:

1. DOM Manipulation: Using CSS-style selectors, you can easily manipulate the DOM by choosing items and then performing different operations or adjustments to them. You may add, delete, or alter items, as well as change their styles and characteristics.

1. Event Handling: jQuery simplifies and cross-browser-compatible event handling on web sites. Event handlers may be simply attached to elements and defined to conduct actions when events such as clicks, mouse movements, or key presses occur.
2. Ajax Support: jQuery makes it easier to make asynchronous JavaScript and XML (Ajax) queries. It includes methods for performing Ajax calls, transferring data to a server, and managing replies. This enables you to construct dynamic web apps that can update sections of a page without having to reload the entire page.

4. Animation and Effects: Built-in animation and effect methods in jQuery make it simple to construct dynamic and aesthetically appealing web pages. With just a few lines of code, you can animate components, create transitions, fade effects, and much more.

1. Cross-Browser Compatibility: jQuery is designed to work consistently across different web browsers. It takes care of browser inconsistencies and provides a unified interface for interacting with the DOM, making it easier to develop cross-browser compatible code.

Overall, jQuery simplifies many basic JavaScript operations while also providing a robust and expressive interface for interacting with web pages. However, with developments in current JavaScript and the advent of new web standards, the necessity for jQuery has lessened to some extent. Many of the jQuery functionalities are now natively available in newer browsers and frameworks like as React, Angular, and Vue.js provides more complete tools for developing complicated web apps.

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

To apply CSS using jQuery, you can use the `css()` method. It allows you to modify the CSS properties of selected elements. Here's an example:

```javascript

// Apply CSS using jQuery

$(selector).css(propertyName, value);

// Example: Change the color of all paragraphs to red

$('p').css('color', 'red');

```

In the example above, the `$()` function is used to select all `<p>` elements on the page, and the `css()` method is called to change the color property of those elements to red.

To add a class to an element using jQuery, you can use the `addClass()` method. It adds one or more class names to the selected elements. Here's an example:

```javascript

// Add class using jQuery

$(selector).addClass(className);

// Example: Add the "highlight" class to all paragraphs

$('p').addClass('highlight');

```

In this example, the `addClass()` method is called on all `<p>` elements, adding the "highlight" class to them.

To remove a class from an element using jQuery, you can use the `removeClass()` method. It removes one or more class names from the selected elements. Here's an example:

```javascript

// Remove class using jQuery

$(selector).removeClass(className);

// Example: Remove the "highlight" class from all paragraphs

$('p').removeClass('highlight');

```

In this example, the `removeClass()` method is called on all `<p>` elements, removing the "highlight" class from them.

Regarding jQuery animation, jQuery provides a range of animation effects that you can apply to elements. The most commonly used method is `animate()`. It allows you to animate CSS properties over a specified duration. Here's an example:

```javascript

// jQuery animation

$(selector).animate({ properties }, duration, easing, complete);

// Example: Animate the width of a div to 300 pixels over 1 second

$('div').animate({ width: '300px' }, 1000);

```

In the example above, the `animate()` method is called on all `<div>` elements. It animates the width property of the elements to 300 pixels over a duration of 1 second.

You can specify multiple properties within the object passed to `animate()` to animate multiple CSS properties simultaneously. The `duration` parameter defines the duration of the animation in milliseconds. The `easing` parameter specifies the easing function to control the animation's pace, and the `complete` parameter is a callback function that runs after the animation is complete.

These are just some basic examples of how to apply CSS, add/remove classes, and use animation effects with jQuery. jQuery provides a wide range of methods and options for handling these tasks, allowing you to create dynamic and visually appealing web pages.

1. **How to create slider with animation?**

To create a slider with animation using jQuery, you can follow these steps:

1. HTML Structure: Start by creating the HTML structure for your slider. Typically, it involves a container element that holds the slides and navigation controls. For example:

```html

<div class="slider">

<div class="slides">

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

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

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

</div>

<div class="controls">

<button class="prev">Previous</button>

<button class="next">Next</button>

</div>

</div>

```

2. CSS Styling: Apply CSS styles to position the slides and style the navigation controls. You can customize the styles based on your design requirements.

3. jQuery Animation: Use jQuery to handle the animation and navigation functionality. Here's an example implementation:

```javascript

$(document).ready(function() {

var $slider = $('.slider');

var $slides = $slider.find('.slide');

var slideCount = $slides.length;

var currentIndex = 0;

// Function to show the current slide

function showSlide(index) {

$slides.removeClass('active');

$slides.eq(index).addClass('active');

}

// Function to navigate to the next slide

function nextSlide() {

currentIndex++;

if (currentIndex >= slideCount) {

currentIndex = 0;

}

showSlide(currentIndex);

}

// Function to navigate to the previous slide

function prevSlide() {

currentIndex--;

if (currentIndex < 0) {

currentIndex = slideCount - 1;

}

showSlide(currentIndex);

}

// Attach click event handlers to the navigation buttons

$slider.find('.next').on('click', nextSlide);

$slider.find('.prev').on('click', prevSlide);

// Show the initial slide

showSlide(currentIndex);

});

```

In the example above, we define functions to show the current slide (`showSlide()`), navigate to the next slide (`nextSlide()`), and navigate to the previous slide (`prevSlide()`). The `showSlide()` function adds the "active" class to the current slide, while removing it from the others.

We attach click event handlers to the next and previous buttons using jQuery's `on()` method. When the buttons are clicked, the corresponding functions (`nextSlide()` and `prevSlide()`) are executed to update the current slide.

Finally, we call the `showSlide()` function initially to display the first slide.

Remember to include the jQuery library in your HTML file before the script code.

By customizing the CSS styles and adding transitions or animations, you can create visually appealing slider effects. Additionally, you may want to include additional features such as automatic sliding, slide pagination, or slide indicators to enhance the functionality of your slider.