

Module– 5

JQuery Basic, Effects & Advanced

1. What is jQuery?

- jQuery greatly simplifies JavaScript programming.
- jQuery is easy to learn.
- jQuery is a lightweight, "write less, do more", 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.

Here are some key features of jQuery:

- **DOM Manipulation:** jQuery makes it easy to select elements in a document and manipulate them. You can change the content, structure, and style of HTML elements.
- **Event Handling:** jQuery provides a simple way to capture a wide variety of events, such as clicks, mouse movements, and keyboard input, and respond to them with custom functions.
- **AJAX Support:** jQuery includes functions to simplify making asynchronous HTTP requests, allowing you to load or send data to a web server without refreshing the page.
- **Animations and Effects:** jQuery includes a range of built-in animations and effects, such as fading elements in and out, sliding elements up and down, and creating custom animations.
- **Cross-Browser Compatibility:** jQuery handles many of the cross-browser issues that developers used to face, providing a more consistent experience across different web browsers.
- **Plugins:** jQuery's plugin architecture allows developers to create reusable components and extend the functionality of the core library.

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

2.1] How to Apply CSS Using JQuery.

- To apply CSS using jQuery, you can use the **.css()** method. This method allows you to get or set style properties of the selected elements.

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery CSS Example</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script
>
  <script>
    $(document).ready(function(){
      $("button").click(function(){
        $("p").css("color", "red");
      });
    });
  </script>
</head>
<body>

<button>Change Text Color</button>
<p>This is a paragraph.</p>

</body>
</html>
```

- In this example, when the button is clicked, the color of all paragraphs will change to red.

2.2] How to Add Class and Remove Class in JQuery

- Using the .addClass() and .removeClass() methods, you can apply and remove CSS classes.
- In below example, clicking "Add Class" will apply the highlight class to all paragraphs, while clicking "Remove Class" will remove it.

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery CSS Example</title>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style>
    .highlight {
      color: white;
      background-color: green;
      font-weight: bold;
    }
  </style>
  <script>
    $(document).ready(function(){
      $("#add").click(function(){
        $("p").addClass("highlight");
      });
      $("#remove").click(function(){
        $("p").removeClass("highlight");
      });
    });
  </script>
</head>
<body>

<button id="add">Add Class</button>
<button id="remove">Remove Class</button>
<p>This is a paragraph.</p>

</body>
</html>
```

2.3] JQuery Animation

To create Animations on a web page, use the jQuery animate() method. To move an element on a web page, you need to also use the position property. Set the value to relative, fixed, or absolute for the HTML element.

Let us see how to:

Animate a div with animate()

Animate a div with animate() – Use multiple properties

Animate a div

To animate a div, use the animate() method. In the below example, we have animated a div and moved it to the right. The left property is set to 300px and this is the limit till when the div will animate:

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({left: '300px'});
    });
});
</script>
<style>
div {
    background:red;
    border: 2px solid blue;
    height:200px;
    width:200px;
    color: white;
    text-align: center;
    font-size: 25px;
    position:absolute;
}
</style>
</head>
<body>
<h1> jQuery Animation</h1>
<button> Animate</button><br>
<p>Click the above button to animate the below box:</p>
<div>This box will get animated</div>

</body>
</html>
```

Animate a Div – Use multiple properties)

In this example, we will animate a Div and update the height, width, and font size as well. With jQuery, we can easily animate a div using the **animate()** method and update the div properties alongside. That means we can change multiple properties of that specific element.

Let us see an example wherein we will animate a div and update the height, width, and font size properties as well. For the `animate()` method, the **font-size** property is to be written in camel-case i.e. `fontSize`. This works for other properties as well:

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            left: '200px',
            height: '100px',
            width: '100px',
            fontSize: '15px'
        });
    });
});
</script>
<style>
div {
    background:red;
    border: 2px solid blue;
    height:200px;
    width:200px;
    color: white;
    text-align: center;
    font-size: 22px;
    position:absolute;
}
</style>
</head>
<body>
<h1> jQuery Animation</h1>
<button> Animate</button><br>
<p>Click the above button to animate the below box:</p>
<div>This box will get animated and the properties will also change</div>

</body>
</html>
```

3. How to create slider with animation?

- Creating an image slider with animation using jQuery involves setting up the HTML structure, styling it with CSS, and then writing the jQuery code to handle the slider logic and animations.

Step 1: HTML Structure

Create the HTML structure for the slider. This includes a container for the slider and the images within it.

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Slider Example</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="scripts.js"></script>
</head>
<body>

<div class="slider">
  <div class="slides">
    <div class="slide"></div>
    <div class="slide"></div>
    <div class="slide"></div>
  </div>
</div>

</body>
</html>
```

Step 2: CSS

- Style the slider and its contents using CSS.

```
/* styles.css */
.slider {
  width: 600px;
  height: 400px;
  overflow: hidden;
  position: relative;
}

.slides {
  display: flex;
  width: 300%; /* 100% for each slide */
  transition: transform 1s ease;
}

.slide {
  width: 100%;
  flex-shrink: 0;
}

.slide img {
  width: 100%;
  height: auto;
}
```

Step 3: jQuery

- Write the jQuery code to handle the slider animation

```
// scripts.js
$(document).ready(function() {
  let currentIndex = 0;
  const slides = $(".slide");
  const totalSlides = slides.length;

  function showNextSlide() {
    currentIndex = (currentIndex + 1) % totalSlides;
    $(".slides").css("transform", `translateX(-${currentIndex * 100}%)`);
  }

  setInterval(showNextSlide, 3000); // Change slide every 3 seconds
});
```


Explanation

1. HTML Structure:

- The .slider container holds the .slides div, which contains individual .slide divs for each image.

2. CSS:

- The .slider container has a fixed width and height, with overflow: hidden to hide the overflow from the images.
- The .slides div uses display: flex to position the slides horizontally. The width is set to 300% to accommodate three slides side by side.
- The .slide divs take up 100% of the width of the .slider container.
- The transition property on .slides ensures smooth animation when the transform property changes.

3. jQuery:

- When the document is ready, the script initializes the currentIndex to 0.
- The showNextSlide function updates the currentIndex to point to the next slide, wrapping back to the first slide after the last one.
- The transform CSS property is updated to shift the .slides container, creating a sliding effect.
- The setInterval function calls showNextSlide every 3 seconds to automatically transition between slides.

This basic slider can be extended with additional features such as navigation buttons, indicators, and pause on hover functionality.