2. jQuery Basics for Modifying HTML

Using jQuery's Built-in Methods to Modify Bootstrap HTML

With the jQuery JavaScript library alone, you can do some amazing things within the HTML files you create by adding small scripts, enabling interactivity with the user as well as CSS and HTML modifications.

Responding to user-generated events, like clicking on certain elements, is an important capability of simple JavaScript scripts built using jQuery.

In the examples here, we will look at nine different built-in methods of the jQuery JavaScript library:

1. **click()**
   The jQuery **click()** method gives us the chance to provide a block of code that runs when a click event takes place because the user clicked on an HTML element. Many other event handler methods besides **click()** are built into jQuery. See the **jQuery API documentation (http://api.jquery.com/category/events/)** for full details on all event handlers.
   $('button').click(function() { });

2. **addClass()**, **removeClass()**, and **toggleClass()**
   The jQuery **toggleClass()** method adds or removes (if it already exists) a class attribute value on any HTML elements targeted by the selector you provide jQuery when calling the method. **addClass()**, and **removeClass()** add or remove a class value.
   $('.item1').toggleClass('active');

3. **append()** and **html()**
   The jQuery **append()** method adds specified text content into the very end of any HTML elements targeted by the selector you provide jQuery when calling the method. **html()** is like append(), but replaces all existing content.
   $('.target1').append('New content');

4. **show(), hide()**, and **toggle()**
   The jQuery toggle() method shows (if hidden) or hides any HTML elements targeted by the selector you provide jQuery when calling the method. **show()** and **hide()** show or hide elements.
   $('p').toggle();

Using jQuery to Control Element Classes

With the **addClass()**, **removeClass()**, and **toggleClass()** methods we can easily change HTML elements' class attribute values in response to user interaction. This example allows a user to toggle (add or remove) a class value, "active". The code runs only when a user clicks on one of the elements matching the selector ".item1, .item2" which – just as in CSS – matches elements with class="item1" or class="item2".

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>jQuery Basics: addClass(), removeClass(), and toggleClass()</title>
  <!-- load bootstrap css via cdn -->
  <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <!-- load jquery js via cdn -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <!-- load bootstrap js via cdn -->
  <script src="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
  <style>
   .item1 {
     background:orange;
   }
   .item2 {
     background:yellow;
   }
```

```
    .active {
      background:blue;
      color:white;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-12">
        <h2>Click to toggle class</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6 item1">
        <p>Item 1</p>
      </div>
      <div class="col-md-6 item2">
        <p>Item 2</p>
      </div>
    </div>
  </div>
  <!-- custom javascript using jquery to handle click events -->
  <script>
    // quoted text in parens after $ call contains a selector, like css
    $(".item1, .item2").click(
      function() {
        // code inside function's {} runs when element matching selector clicked
        $(this).toggleClass("active");
        // toggleClass() adds the requested class(es) if missing, removes if there
        // addClass() always adds the requested class(es)
        // removeClass() always removes the requested class(es)
      }
    );
  </script>
</body>
</html>
```

Using jQuery to Modify the Contents of Elements

Using the **append()** method, we can insert content at the end of an element (before its closing tag) or with the **html()** method, you can replace the existing contents entirely with new content. This example attaches code to the click event on any button in the page with **$("button").click()**. Try switching the **append()** method out for the **html()** method to observe the difference in the two methods' behavior.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>jQuery Basics: append() and html()</title>
  <!-- load bootstrap css via cdn -->
  <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <!-- load jquery js via cdn -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <!-- load bootstrap js via cdn -->
  <script src="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col-md-12 target1">
        <h2>Click below to append content</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <button>Click me</button>
      </div>
      <div class="col-md-6">
        <button>Click me</button>
      </div>
    </div>
  </div>
  <!-- custom javascript using jquery to handle click events -->
  <script>
    // quoted text in parens after $ call contains a selector, like css
    $("button").click(
      function() {
        // code inside function's {} runs when element matching selector clicked
        $(".target1").append("<p>New content</p>");
        // .append() adds content into element right before closing tag
        // .html() replaces all content between opening and closing tags
      }
    );
  </script>
</body>
</html>
```

Showing and Hiding Elements with jQuery

The jQuery methods **show()**, **hide()** and **toggle()** give you an easy way to conceal or display one or more elements. The manner in which **show()**, **hide()** and **toggle()** work is all the same – manipulation of the targeted elements' style attribute, using the CSS **display** property. The **hide()** method sets affected elements' style attribute to "display:none" while the **show()** method sets the style attribute to "display:block".

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>jQuery Basics: show(), hide(), and toggle()</title>
  <!-- load bootstrap css via cdn -->
  <link rel="stylesheet" href="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css">
  <!-- load jquery js via cdn -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
  <!-- load bootstrap js via cdn -->
  <script src="https://netdna.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>
</head>
<body>
  <div class="container">
```

```
    <div class="row">
      <div class="col-md-12 target1">
        <h2>Click below to show or hide me</h2>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <button class="item1">Click to hide</button>
      </div>
      <div class="col-md-6">
        <button class="item2">Click to show</button>
      </div>
    </div>
  </div>
  <!-- custom javascript using jquery to handle click events -->
  <script>
    // quoted text in parens after $ call contains a selector, like css
    $(".item1").click(
      function() {
        // code inside function's {} runs when element matching selector clicked
        $(".target1").hide();
        // hide() always sets style attribute to display:none
      }
    );
    // quoted text in parens after $ call contains a selector, like css
    $(".item2").click(
      function() {
        // code inside function's {} runs when element matching selector clicked
        $(".target1").show();
        // hide() always sets style attribute to display:none
      }
    );
  </script>
</body>
</html>
```

Explore the other built-in methods of the jQuery JavaScript library by visiting the jQuery Learning Center at **http://learn.jquery.com/ (http://learn.jquery.com/)** .