3. Search Filtering with jQuery and JavaScript

Dynamic filtering of content is an essential capability for sites that provide access to large volumes of content. The jQuery library's extensive support for selectors includes the **:contains()** CSS pseudo-class selector. With the ability to specify elements by selectors that include reference to the elements' contents, it becomes possible to find elements based on a user-entered search filter term.

jQuery Methods in Use

Here we look at the use of the jQuery library's built-in **val()**, **children()**, **filter()** and **change()** and **css()** methods.

$('#myitem').val();	Returns the user-entered or selected value of form input element with *id="myitem"*.
$('#myitem').children();	Return an array containing references to all of the children of element with *id="myitem"*.
$('.mygroup').filter('li');	Returns only the items in selection that match filter selector for elements with *class="mygroup"*.
$('#myitem').change(function() {} **);**	The **change()** event handler runs code in anonymous function code block when value of form element is changed by user.
$('.mygroup').css('property-name', 'property-value'**);**	Sets the CSS property named in first parameter to property value in second parameter for elements with *class="mygroup"*.

Example

Our example takes a parent element, in this case, an unordered list, and paginates the child list item elements. The code manages the display of ten list items at a time by selectively hiding and showing list items. The code generates navigation to allow user to trigger events that call functions to switch set of list items shown.

Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>jQuery Example: Search Filtering</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">
        <h2>Filtering list with search text</h2>
        <!-- form input for user-entered filter text -->
        <form>
          <input id="filtertext">
```

```
        </form>
        <!-- list containing searched items -->
        <ul id="itemlist">
          <li>Apple</li>
          <li>Banana</li>
          <li>Carrot</li>
          <li>Date</li>
          <li>Endive</li>
          <li>Fig</li>
          <li>Grape</li>
          <li>Honeydew melon</li>
          <li>Iceberg lettuce</li>
          <li>Jerusalem artichoke</li>
          <li>Kiwi</li>
          <li>Lemon</li>
          <li>Mango</li>
          <li>Nectarine</li>
          <li>Orange</li>
          <li>Pear</li>
          <li>Quince</li>
          <li>Radish</li>
          <li>Strawberry</li>
          <li>Tomato</li>
          <li>Uglifruit</li>
          <li>Victoria plum</li>
          <li>Watermelon</li>
          <li>Yam</li>
          <li>Zucchini</li>
        </ul>
      </div>
    </div>
  </div>
  <!-- custom javascript using jquery to handle dynamic search filtering -->
  <script>
    // create event handler for user changes filter text input value
    $("#filtertext").change(
      function() {
        // save jquery selector text for finding matches for X
        var t = "li:contains('X')";
        // use user-entered filter text instead of X
        t = t.replace( "X", $("#filtertext").val() );
        // hide all child list items
        $('#itemlist').children().css('display','none');
        // show children list items with content matching filter text
        $('#itemlist').children().filter( t ).css('display','block');
      }
    );
  </script>
</body>
</html>
```

Try out some search terms in the code you create. Notice that the jQuery **filter()** method is case-sensitive when matching the **contains()** selector.