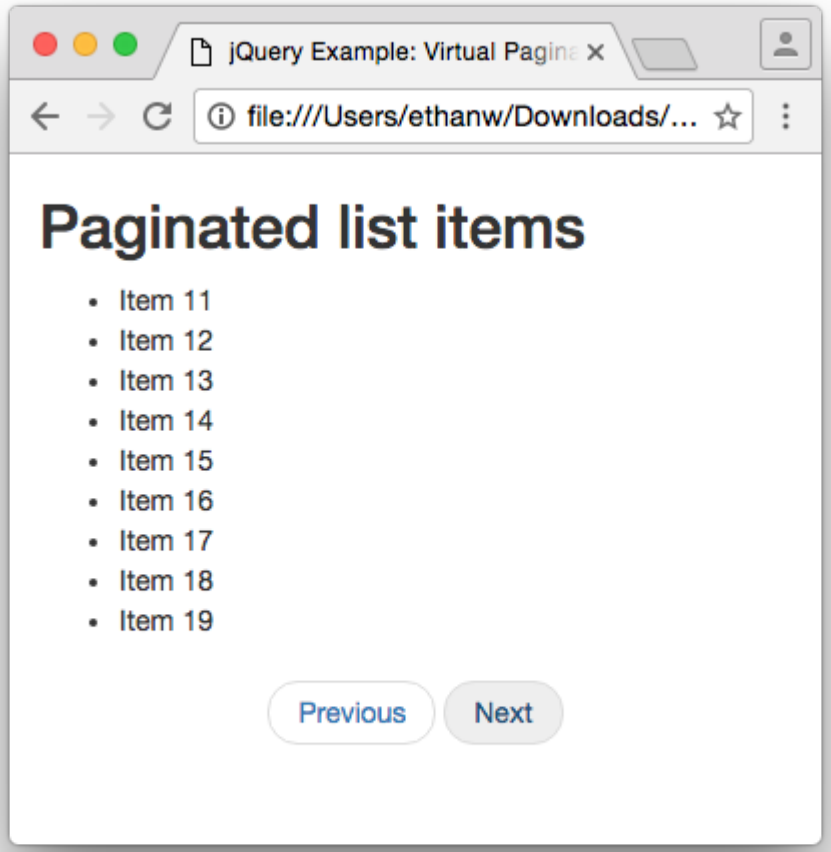
2. Virtual Pagination with jQuery and JavaScript

In this week's explorations of using JavaScript with the jQuery library, we will work with function-based JavaScript code blocks that allow us to manage content display and respond to user-generated events.

In the first experiment, we face the real-world problem of too much content on a page. The "wall of text" syndrome is a well-known experience. Come upon a Web page containing too much text – especially without a strong visual hierarchy – and one's comprehension is immediately overwhelmed. Good content design should utilize pagination.

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.

jQuery Methods in Use

Here we look at the use of the jQuery library's built-in **children()**, **click()**, **show()** and **hide()** methods.

$('#myitem').children();	Return an array containing references to all of the children of element with *id="myitem"*.
$('#myitem').click(function() { } **);**	The **click()** event handler runs the code block contained in the anonymous function when user clicks element with *id="myitem".*
$('.mygroup').show();	Sets the CSS display **property** of elements with *class="mygroup"* to **block**.
$('.mygroup').hide();	Sets the CSS display **property** of elements with *class="mygroup"* to **none**.

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: Virtual Pagination</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>Paginated list items</h2>
        <!-- list containing dynamically-paginated items -->
        <ul id="itemlist">
          <li>Item 1</li>
          <li>Item 2</li>
          <li>Item 3</li>
          <li>Item 4</li>
          <li>Item 5</li>
          <li>Item 6</li>
          <li>Item 7</li>
          <li>Item 8</li>
          <li>Item 9</li>
          <li>Item 10</li>
          <li>Item 11</li>
          <li>Item 12</li>
          <li>Item 13</li>
          <li>Item 14</li>
          <li>Item 15</li>
          <li>Item 16</li>
          <li>Item 17</li>
          <li>Item 18</li>
          <li>Item 19</li>
          <li>Item 20</li>
          <li>Item 21</li>
          <li>Item 22</li>
          <li>Item 23</li>
          <li>Item 24</li>
          <li>Item 25</li>
        </ul>
        <!-- bootstrap pager -->
        <ul class="pager">
          <li><a href="#" id="page-prev">Previous</a></li>
          <li><a href="#" id="page-next">Next</a></li>
        </ul>
      </div>
    </div>
```

```
  </div>
  <!-- custom javascript using jquery to handle dynamic pagination and pager -->
  <script>
    // keep track of page showing
    var page = 0;

    // function to show / hide sets of 10 list items base on page number
    function paginate() {
      // hide all child list items
      $('#itemlist').children().hide();
      // show range of children
      $('#itemlist').children().slice( page * 10, page * 10 + 9).show();
    }

    // call paginate() first time to hide items
    paginate();

    // quoted text in parens after $ call contains selector for first list item
    $("#page-prev").click(
      function() {
        // code inside function's {} runs when element matching selector clicked
        page = page - 1;
        if ( page < 0 ) page = 0;
        paginate();
      }
    );
    $("#page-next").click(
      function() {
        page = page + 1;
        if ( page * 10 > $('#itemlist').children().length ) page = page - 1;
        paginate();
      }
    );
  </script>
</body>
</html>
```

Play by Play: jQuery Chaining

The long last line of code in the **paginate()** function's code block above is dense with a lot going on.

```
$('#itemlist').children().slice( page * 10, page * 10 + 9).show();
```

That single line is using JavaScript dot notation and jQuery chaining (see [http://www.w3schools.com/jquery/jquery_chaining.asp (http://www.w3schools.com/jquery/jquery_chaining.asp)](http://www.w3schools.com/jquery/jquery_chaining.asp)) to issue multiple methods on the same set of objects found by the selector **$('#itemlist').**

The methods are executed in the order they appear in the statement, so here is the play-by-play:

1. **$('#itemlist')**
   Call jQuery to return a one-element array with object reference to HTML element with id="itemlist"

2. **.children()**
   *Call jQuery **children()** method to return an array of object references for the children of **<ul id="itemlist">** – all the list items.*

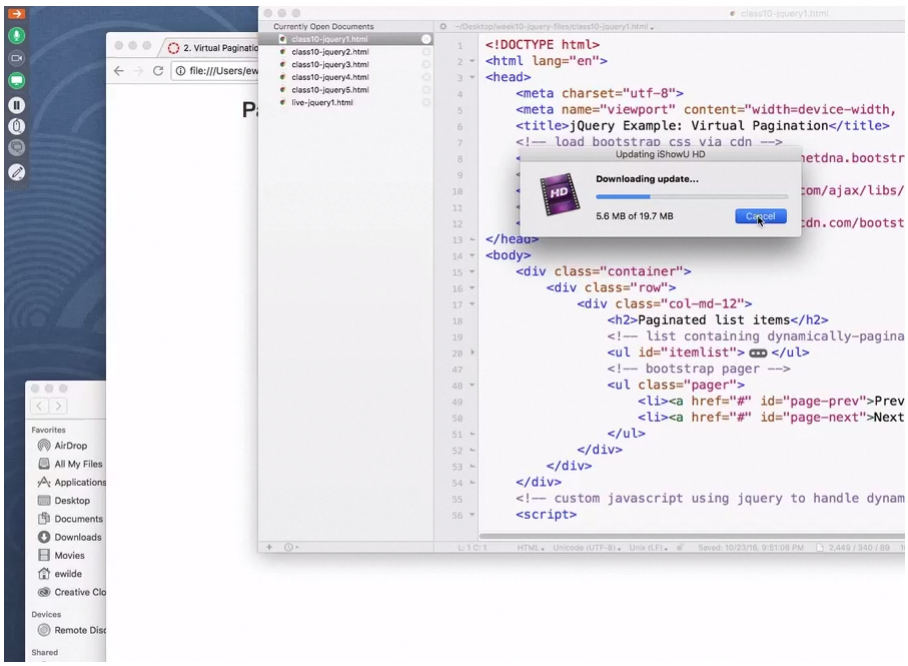
3. **.slice(page * 10, page * 10 + 9)**
   *Use jquery **slice()** method to retrieve a range of elements in an array – here just the ten list items that match the current page number's range. The formula **page * 10** uses the value in the variable **page** to multiply by ten for the starting item to show. The second parameter's formula adds 9 to get the last list item to show.*

the last list item to show.

4. **.show()**
   Call jQuery **show()** method to set the HTML style attribute of the remaining HTML element references in the array from step 3 above. The **show()** method sets the CSS display property to block, by inserting **style="display:block;"** into the HTML element's opening tag.

Still curious about the jQuery **slice()** method? Check out the jQuery project's reference page on **slice()** at https://api.jquery.com/slice/ [(https://api.jquery.com/slice/)](https://api.jquery.com/slice/) – part of their reference section on jQuery methods.

Video Screencast



00:00 / 25:44

[Virtual Pagination with jQuery in JavaScript (http://ethan.com/sric/_videos/CS50B_Week_10_Pagination.m4v)](http://ethan.com/sric/_videos/CS50B_Week_10_Pagination.m4v) (.m4v)