How to use Bootstrap to Create Responsive Inline Navigation Button Groups

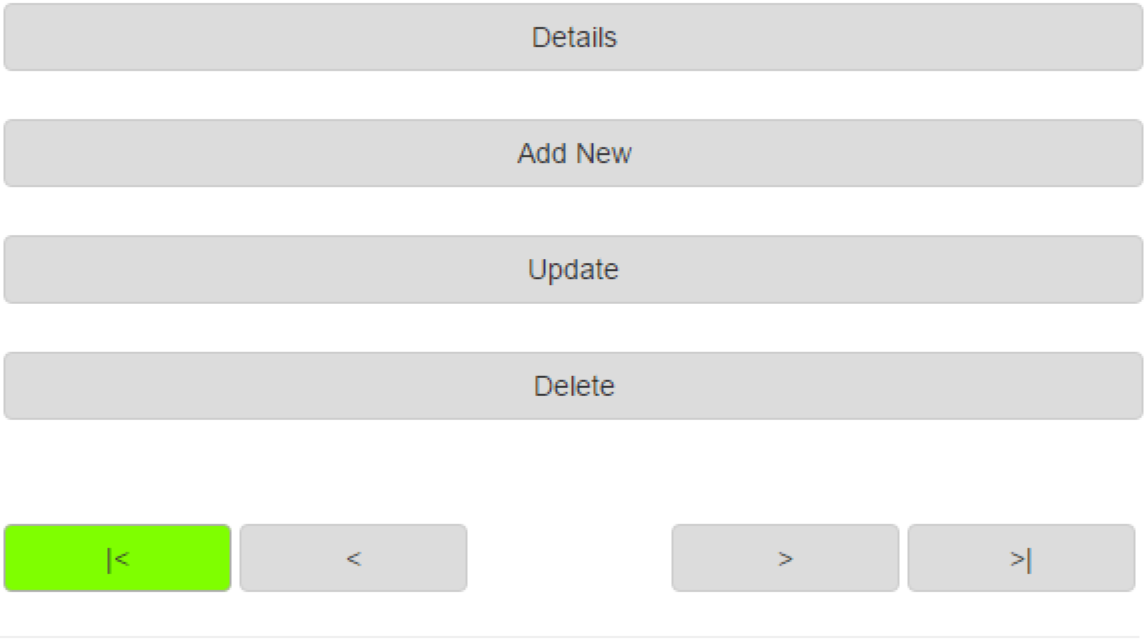
Prepared for:

Web Development Students

Prepared by:

Jason Roundy

November 29, 2016



This handbook is for students, and web developers, to help create responsive inline navigation button groups, on a webpage using Bootstrap. Bootstrap is an HTML, CSS, and JavaScript framework for developing responsive mobile first web projects. Using Bootstrap can be confusing, and time consuming at first, but this handbook will ease some of your fears. A mixture of Bootstrap classes with HTML style attributes, including percentage widths and a margin, create noticeable buttons that line up and respond to the screen size. You should already know basic HTML, and CSS, to set up and style webpages.

WHAT YOU WILL LEARN FROM THIS HANDBOOK

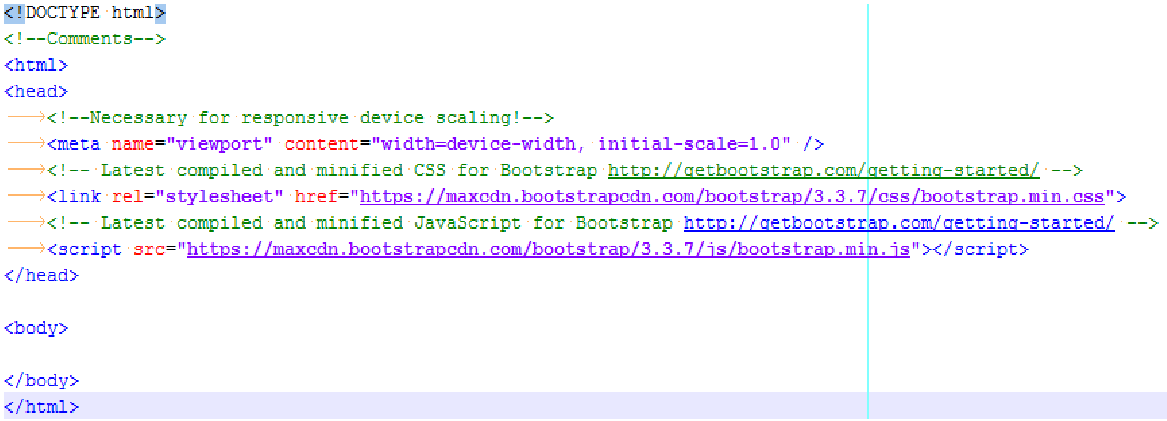
You will create buttons for first, previous, next, and last records. This tutorial will show the requirements needed to use Bootstrap, and how to set up a fluid bootstrap container. Next, you will set up four database navigation buttons with Bootstrap classes, and inline HTML style attributes. You will be able to take this information and change it to suit other button groups by changing key things like the value attribute text, button width percentage, and button margins.

WHAT YOU NEED TO GET STARTED

You are required to have a webpage with head tag elements for responsive scaling and Bootstrap files. Check out getbootstrap.com/getting-started/ (“Getting Started”, n.d.), for the latest links and scripts to copy into your document. Otherwise, use the tag information from the image below.

First, you need to set up the following items:

- HTML webpage
- Meta tag with viewport information for a responsive design
- Bootstrap’s min.css link
- Bootstrap’s min.js script



```
<!DOCTYPE html>
<!--Comments-->
<html>
<head>
	<!--Necessary for responsive device scaling!-->
	<meta name="viewport" content="width=device-width, initial-scale=1.0" />
	<!-- Latest compiled and minified CSS for Bootstrap http://getbootstrap.com/getting-started/ -->
	<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
	<!-- Latest compiled and minified JavaScript for Bootstrap http://getbootstrap.com/getting-started/ -->
	<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>

<body>

</body>
</html>
```

Figure 1 Necessary Web Page Set Up

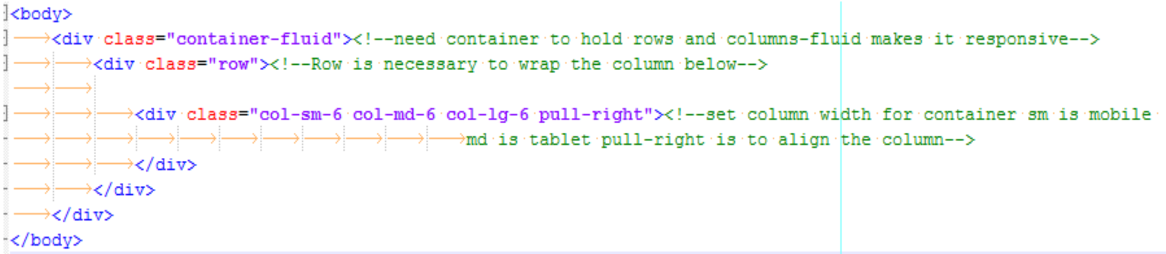
CONTENTS

ILLUSTRATIONS

SET UP A FLUID CONTAINER

Now, you need to set up a fluid container with a row and column, to hold your button group. Fluid containers are responsive to the device screen sizes. A row holds the column in a blocked section, while column lets you choose a portion of the whole row, or page width, based on the screen size.

1. Make a **div** with **class=”container-fluid”** inside the **HTML body element**
2. Make a **div** inside the container-fluid div with **class=”row”**
3. Make a **div** column to hold the button group inside the row div with **class=”col-sm-6 pull-right”**



```
<body>
    <div class="container-fluid"><!--need container to hold rows and columns-fluid makes it responsive-->
        <div class="row"><!--Row is necessary to wrap the column below-->
        
            <div class="col-sm-6 col-md-6 col-lg-6 pull-right"><!--set column width for container sm is mobile
                                            md is tablet pull-right is to align the column-->
            </div>
        </div>
    </div>
</body>
```

Figure 2 A Fluid Container in Bootstrap

Note: Bootstrap uses a 12 column grid. Your column width can be changed by altering the 6 to any number from 1-12. The sm, md, and lg change the column width based on the devices screen size.

SET UP AN INLINE BUTTON GROUP

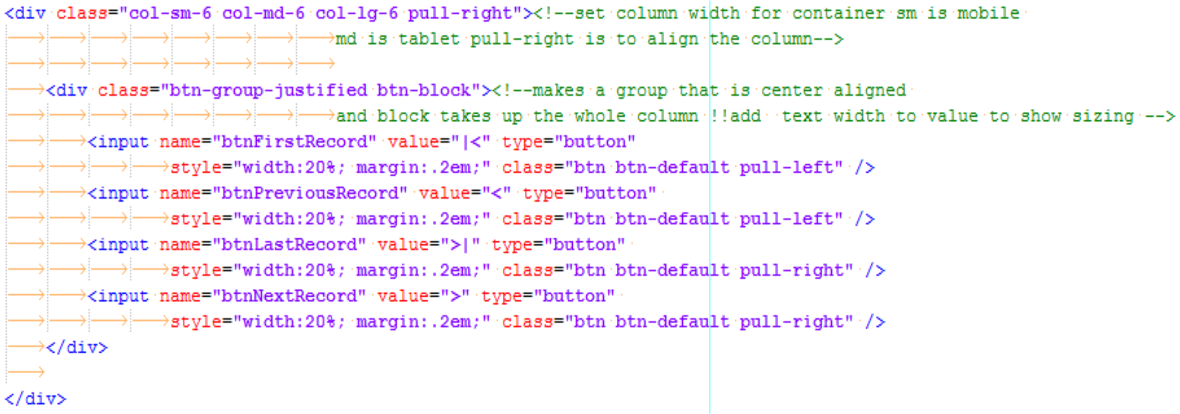
Inside the column div tag, enter the code for our navigation button group.

1. Place a div inside the column div
   - class=”btn-group-justified btn-block"
2. Add four HTML input buttons inside the btn-group-justified div:
   a. Add an HTML input button with;
      - class=”btn btn-default pull-left", name="btnFirstRecord", value="|<", type=”button”

   b. Add an HTML input button with;
      - class=”btn btn-default pull-left", name="btnPreviousRecord", value="<", type=”button”

   c. Add an HTML input button with;
      - class=”btn btn-default pull-right", name="btnLastRecord", value=">|", type=”button”

   d. Add an HTML input button with;
      - class=”btn btn-default pull-right", name="btnNextRecord", value=">", type=”button”
3. Add the button width and margin with a style attribute for each input button:
   - style="width:20%; margin:.5em;"



```
<div class="col-sm-6 col-md-6 col-lg-6 pull-right"><!--set column width for container sm is mobile
                                md is tablet pull-right is to align the column-->
                                
    <div class="btn-group-justified btn-block"><!--makes a group that is center aligned
                                and block takes up the whole column !!add  text width to value to show sizing -->
        <input name="btnFirstRecord" value="|<" type="button"
                style="width:20%; margin:.2em;" class="btn btn-default pull-left" />
        <input name="btnPreviousRecord" value="<" type="button"
                style="width:20%; margin:.2em;" class="btn btn-default pull-left" />
        <input name="btnLastRecord" value=">|" type="button"
                style="width:20%; margin:.2em;" class="btn btn-default pull-right" />
        <input name="btnNextRecord" value=">" type="button"
                style="width:20%; margin:.2em;" class="btn btn-default pull-right" />
    </div>
    
</div>
```

Figure 3 Responsive Inline Button Group Styling

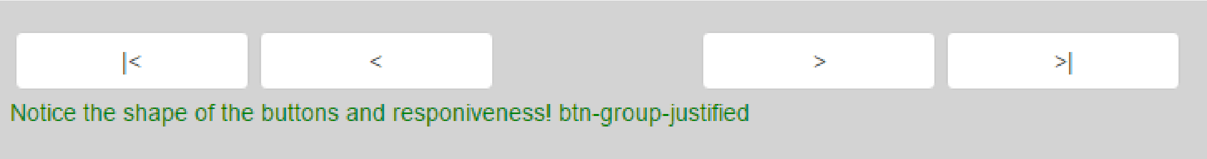
Warning: *Pull-right requires you to order your buttons so that the button farthest to the right is listed before the other pull-right buttons to follow, they will precede it slightly to the left once rendered! Pull-left is the same, each button follows the first button****,*** *and will be slightly to the right of the previous button. Without pull the elements follow their order in the code. This is why the last record comes before the next record!*

Note: You can also change the btn-default to btn-primary, btn-success, btn-info, btn-danger, or another Bootstrap class, which alters the color scheme with Bootstrap prebuilt styling. Just be sure to leave a space in between btn, btn-default, and pull-left or pull-right, this is necessary to use multiple class attributes!

CHECK THE BUTTON GROUP IN A BROWSER

Verify the buttons are taking up the set column width and being responsive without over-running the width or being misaligned.

1. Open the webpage in a browser.
2. Shrink and expand the browser window to observe button group responsiveness.
3. Change the width percentage, and/or margin size, in each button style attribute if the buttons do not line up while checking the responsiveness.
4. Recheck the responsiveness by repeating steps 2 and 3, if necessary.



Notice the shape of the buttons and responiveness! btn-group-justified

Figure 4 Image of the Final Result

Note: To use a different number of buttons, change the width percentage to less than 100 divided by the number of buttons. Play with width and margin until you reach a desired result.

CONCLUSION

You should feel comfortable with using Bootstrap classes for responsive design. Search the Bootstrap website for more information, and feel free to play with different class groupings.

REFERENCES

(Getting Started, n.d.) *Getting Started – Bootstrap.* Retrieved from http://getbootstrap.com/getting-started .