notes_bootstrap_grid.md

Cheat Sheet: The Bootstrap Grid

The Bootstrap CSS Framework provides out-of-the-box CSS classifications ("classes") that can apply to your website. Using their predefined classes, however, means learning the assumptions and conventions that your HTML is expected to follow, in order for your page to appear correctly.

In this guide, we will cover what I consider to be the most valuable part of the framework, which is the *grid system*. It helps us layout our pages as a simple set of rows and columns.

The CSS classes defined by Bootstrap that we need to know are:

- `container` and `container-fluid`
- `row`
- `col-sm-1` up through `col-sm-12`

The Container

If you want to use the Bootstrap grid for your page, you must wrap your page content inside a `div` element that applies the `container` class or `container-fluid` class. (What's the difference? Just try it and see!)

Here's an example HTML fragment that illustrates how you would use the container:

```
<body>
  <div class="container-fluid">

    <!-- Your rows and columns go here -->

  </div>
</body>
```

Rows

Now, once you've got a container set up, you **always** divide up your content into horizontal rows. You must have at least one row.

Here's is a snippet that shows how to define two rows.

```
<body>
  <div class="container-fluid">

    <div class="row">

      <!-- Your columns go here -->

    </div>

    <div class="row">

      <!-- Your columns go here -->

    </div>
  </div>
</body>
```

Columns

Now the hard part. Inside each row, you must have at least one column. You can have up to twelve columns.

Here's the key: your must specify the width of each column, and the widths **must add up to 12**.

Here's is a one-row, two-column layout, useful for a page with "main section" with a right-hand "sidebar":

```
<body>
  <div class="container-fluid">

    <div class="row">

      <div class="col-sm-9">
        <p>This is the main section of the page...</p>
      </div>

      <div class="col-sm-3">
        <p>This is the sidebar.</p>
      </div>

    </div>

  </div>
</body>
```

The `-sm-` part means that columns will appear side-by-side on "small"-size screens (generally speaking, tablets and larger), but they will automatically "stack" if the screen is any smaller (like on a phone).

Responsive Design

You can even use multiple designations on a column, which is helpful when you want different column widths depending on the device size, like `<div class="col-md-3 col-xs-6">` which would take up half the width of a small device but only one-fourth of the width on a medium-sized device.