Bootstrap Grid

Contents

Overview

When formatting our HTML pages, historically, it has been difficult to correctly position elements.
That's why older websites were very simple in terms of layout design.

Until our friend, CSS introduced a technology called **flexbox**.
Flexbox allowed us more control over how elements were positioned, and the relative sizes depending on available space.

Building on flexbox, Bootstrap has its own implementation that allows us even more choice and control: **Bootstrap Grid**.

Tutorial

So, here is what we currently have for our bootstrap page: A dropdown navigation bar.

```
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <title>Bootstrap Tutorial</title>
           <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
    </head>

    <body>
        <nav class="navbar navbar-light bg-light">
            <a href="link/to/page" class="navbar-brand">My Company</a>
            <button class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#hasToMatch">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="hasToMatch">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a href="link/to/login" class="nav-link">Login</a>
                    </li>
                    <li class="nav-item">
                        <a href="link/to/home" class="nav-link">Home</a>
                    </li>
                    <li class="nav-item">
                        <a href="link/to/memes" class="nav-link">Memes</a>
                    </li>
                </ul>
            </div>
        </nav>
       <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta1/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW"
        crossorigin="anonymous"></script>
    </body>

</html>
```

The Bootstrap Grid is divided into rows and columns, we can make as many rows as we need for our page.
However, each row can only have a maximum of 12 elements, although, this is probably for the best.
So, lets get a simple implementation up.

```
<!--<head> and <nav>-->
<div class="container">
    <div class="row">
        <div class="col-2">
            Row 1 Column 1
        </div>
        <div class="col-10">
            Row 1 Column 2
        </div>
    </div>
    <div class="row">
        <div class="col-12">
            Row 2 Column 1
        </div>
    </div>
</div>
<!--Script imports-->
```

So, we have our wrapping `<div>` with a class `container`.

We then declare a row by giving the class `row` to a new `<div>`.

Now we can start dividing our page accordingly - you'll notice each column has a number with its class name.
You can think of this as the share of the screen this column is taking, where a full screen is 12.
So in our example, `Row 1 Column 1` would take up **1/6** of the screen and `Row 1 Column 2` would take up **5/6** of the screen (measuring horizontally).

We can have columns that do not add up to 12, and Bootstrap will leave that portion of space empty.
However, we cannot have columns that add up to greater than 12 in each row.

```
<!--<head> and <nav>-->
<div class="container">
    <div class="row">
        <div class="col-2">
            Column 1
        </div>
        <div class="col-4">
            Column 2
        </div>
        <div class="col-4">
            Column 3
        </div>
    </div>
</div>
<!--Script imports-->
```

In the above example, our columns add up to 10, leaving 2 unused units of space.
We could just leave them to automatically fill in the right of the screen, or we could let Bootstrap know how to use the blank space for formatting.
We do this with `justify-content`.

Justify content will position our grid boxes using the spare blank space;

- `justify-content-start` Default behaviour, moves content to left, blank space on right.
- `justify-content-end` Moves content to right, blank space on left.
- `justify-content-centre` Moves content to centre, blank space on left and right.
- `justify-content-between` Moves content to edges, blank space in centre.
- `justify-content-around` Puts an even amount of blank space either side of each column. Spacing them out evenly.

```
<!--<head> and <nav>-->
<div class="container">
    <div class="row justify-content-around">
        <div class="col-2">
            Column 1
        </div>
        <div class="col-4">
            Column 2
        </div>
        <div class="col-4">
            Column 3
        </div>
    </div>
</div>
<!--Script imports-->
```

Let's put it all together and have a look - here, some custom CSS has been added so that we can see the structure of the grid:

```
<!DOCTYPE html>
<html lang="en">

    <head>
        <meta charset="UTF-8">
        <title>Bootstrap Tutorial</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta1/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1"
crossorigin="anonymous">
        <style>
            .wireframe {
                border: 1px solid black;
            }
            .row-wireframe {
                border: 1px solid black;
                height: 400px
            }
        </style>
    </head>

    <body>
        <nav class="navbar navbar-light bg-light">
            <a href="link/to/page" class="navbar-brand">My Company</a>
            <button class="navbar-toggler" data-bs-toggle="collapse" data-bs-
target="#hasToMatch">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="hasToMatch">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a href="link/to/login" class="nav-link">Login</a>
                    </li>
                    <li class="nav-item">
                        <a href="link/to/home" class="nav-link">Home</a>
                    </li>
                    <li class="nav-item">
                        <a href="link/to/memes" class="nav-link">Memes</a>
                    </li>
                </ul>
            </div>
        </nav>
        <div class="container wireframe">
            <div class="row justify-content-around row-wireframe">
                <div class="col-2 wireframe">
                    Row 1 Column 1
                </div>
                <div class="col-4 wireframe">
                    Row 1 Column 2
                </div>
                <div class="col-4 wireframe">
                    Row 1 Column 3
                </div>
            </div>
            <div class="row justify-content-between row-wireframe">
                <div class="col-4 wireframe">
                    Row 2 Left
                </div>
                <div class="col-4 wireframe">
                    Row 2 Right
                </div>
            </div>
        </div>
   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-
beta1/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-
ygbV9kiqUc6oa4msXn9868pTtWMgiQaeYH7/t7LECLbyPA2x65Kgf80OJFdroafW"
        crossorigin="anonymous"></script>
    </body>
```

```
</html>
```

Different Device Sizes

Web sites are accessed on many different devices; thus, we need to account for the fact that we won't have the same amount of space to be working with all the time.

As with all things in Bootstrap, this is quite easy.
Before, when we labelled our columns we used `col-12` or `col-6`.
Now, all we have to do is specify a screen size;

- `col-xs-6`: Extra small screens like smart phones.
- `col-sm-6`: Small screens like tablets.
- `col-md-6`: Medium screens like laptops or smaller desktop monitors.
- `col-lg-6`: Large screens like larger desktop monitors.
- `col-xl-6`: Extra large screens like TVs etc.

Then we simply list all the rules for the screen sizes we want. Bootstrap will automatically apply the correct formatting, depending on the user's screen size:

```
<!--<head> and <nav>-->
<div class="container">
    <div class="row justify-content-around">
        <div class="col-sm-3 col-md-2 col-lg-3">
            Column 1
        </div>
        <div class="col-sm-3 col-md-4 col-lg-4">
            Column 2
        </div>
        <div class="col-sm-3 col-md-4 col-lg-4">
            Column 3
        </div>
    </div>
</div>
<!--Script imports-->
```

Exercises

There are no exercises for this module.