CHAPTER 2

Advanced Grid Techniques

In the previous chapter, you started learning about the Twitter Bootstrap grid system. In this chapter, you will continue your work on the grid system, but you will use some different, more advanced techniques to do similar and other things.

Learning Goals

You will learn

1. How to use row columns
2. How to vertically align content
3. How to horizontally align
4. How to use less than the 12 available columns
5. How to nest content and a grid inside another
6. What the Twitter Bootstrap defaults are

Shortcuts on Rows: Row Columns

You are now going to explore one more set of alternatives in order to do all the things that you did in Chapter 1, "Getting Started." This is called *row columns*, and it is a set of classes that help you define the columns of a row but using them at the `row div`, not at the `col div`.

P. Matsinopoulos, *Practical Bootstrap*, https://doi.org/10.1007/978-1-4842-6071-5_2

Example: Six-Column Layout for All Devices

Let's suppose that I want to achieve a six-column layout that will be for all device widths. Listing 2-1 is my code.

Listing 2-1. Six-Column Layout Using Row Columns

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-1</title>
</head>
<body>
    <div class="container">
        <div class="row row-cols-6">
            <div class="col">
                1
            </div>
            <div class="col">
                2
            </div>
            <div class="col">
                3
            </div>
```

```
            <div class="col">
                4
            </div>
            <div class="col">
                5
            </div>
            <div class="col">
                6
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

If you save the preceding code into your `index.html` page and load on your browser, you will see something like that in Figure 2-1.

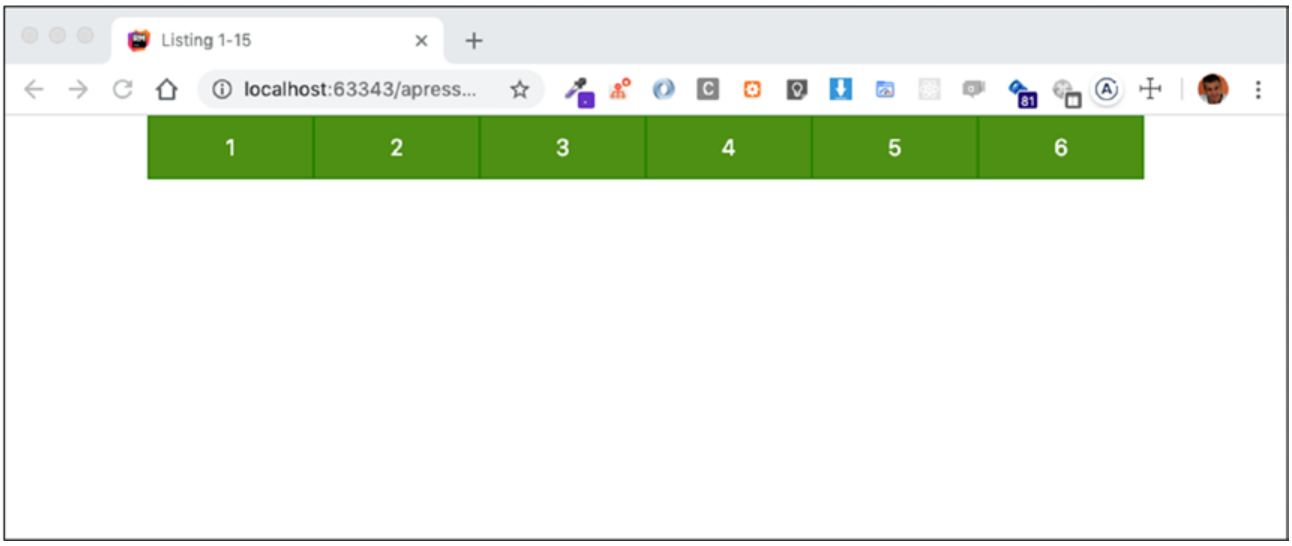

Figure 2-1. *Six Columns Using the row-cols-* Class*

This is exactly what I wanted. How was it created?

1. The row div was attributed with the class `row-cols-6`. The number 6 was exactly the number of the columns I wanted.
2. Then each column `div` only had the `col` class, since I wanted all the columns to have equal width.

Of course, I am now hearing you saying, "Hold on. The same would have been accomplished if I had used only the `row` class without the `row-cols-6` class." And you would have been right. The use of `row-cols-*` in the previous example didn't add too much value. But let's do another example in which the row column classes are much more valuable.

Example: Three Columns for Medium, Six Columns for Large

Now I want to create a responsive grid that would give me

- Three columns for all devices up to and including medium ones
- Six columns for all devices with large or wider displays

You already know how to do that with the `col-*` class at the column `div` level. Here, we are going to do it with `row-cols-*` classes at the row level. See Listing 2-2.

Listing 2-2. Responsive Breakpoints Using Row Columns

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-2</title>
</head>
<body>
    <div class="container">
        <div class="row row-cols-3 row-cols-lg-6">
            <div class="col">
                1
            </div>
            <div class="col">
                2
            </div>
            <div class="col">
                3
            </div>
            <div class="col">
                4
            </div>
```

```
            <div class="col">
                5
            </div>
            <div class="col">
                6
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Do you see how easy it is?

- I keep the column `divs` having only the `col` class.
- I use the two row column classes `row-cols-3` and `row-cols-lg-6` to tell how many columns I want according to the device width, in particular
  - `row-cols-3` for all devices except the ones defined with the following class. Note the number `3` is the number of columns we want to occupy.
  - `row-cols-lg-6` for all devices with large or wider displays, that is, >=992px.

If you save your page and reload on your browser, you will see the following for a page that has width 991px, for example (Figure 2-2).

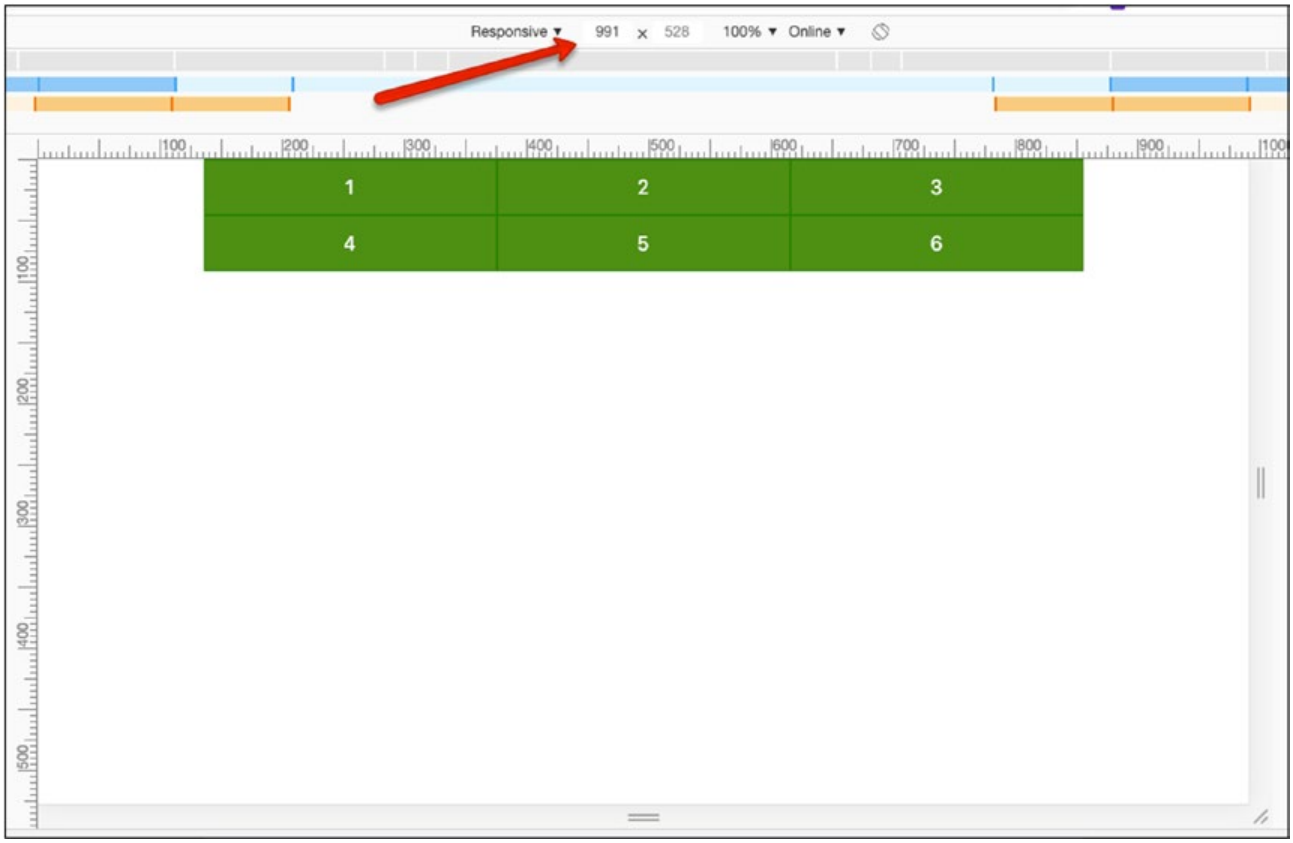

Figure 2-2. *Three Columns When Below 992px*

When you enlarge the width by 1 pixel more, to 992px, then you will immediately get six columns (Figure 2-3).

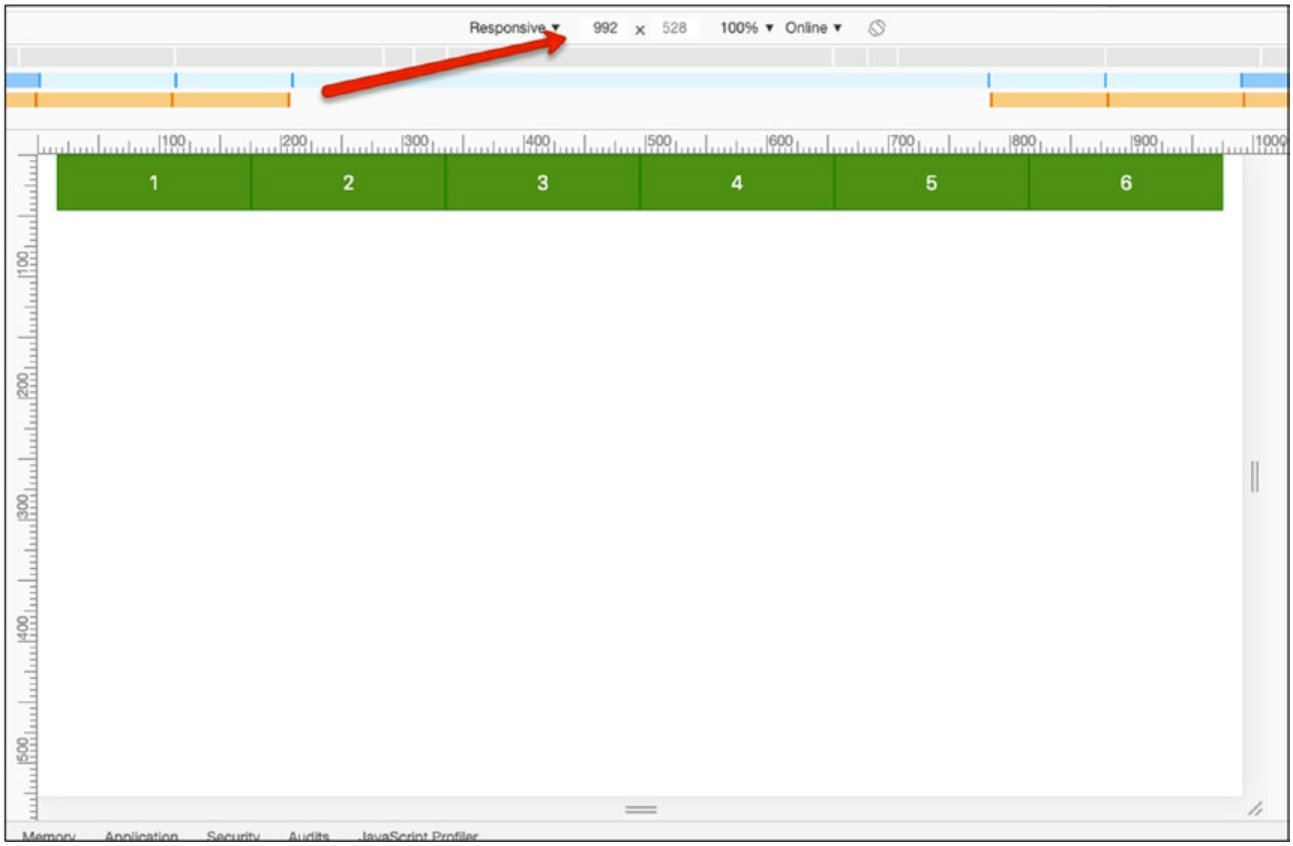

Figure 2-3. *Six Columns for 992px and Wider*

In Table 2-1, I compare the first solution with the `col-*` classes at the column level with the latest solution you just learned with the `row-cols-*` classes at the row level.

Table 2-1. *Compare Code Between* `col-*` *and* `row-cols-*`

`col-*` at Column Level	`row-cols-*` at Row Level

`<div class="row">`
`<div class="col-4 col-lg-2">`
`1`
`</div>`
`<div class="col-4 col-lg-2">`
`2`
`</div>`
`<div class="col-4 col-lg-2">`
`3`
`</div>`
`<div class="col-4 col-lg-2">`
`4`
`</div>`
`<div class="col-4 col-lg-2">`
`5`
`</div>`
`<div class="col-4 col-lg-2">`
`6`
`</div>`
`</div>`
	`<div class="row row-cols-3`
	`row-cols-lg-6">`
	`<div class="col">`
	`1`
	`</div>`
	`<div class="col">`
	`2`
	`</div>`
	`<div class="col">`
	`3`
	`</div>`
	`<div class="col">`
	`4`
	`</div>`
	`<div class="col">`
	`5`
	`</div>`
	`<div class="col">`
	`6`
	`</div>`
	`</div>`

I repeat so that it's crystal clear:

1. When I use classes to specify the column width at the column `div` level, I specify the number of columns (out of the 12) my column will occupy. For example, if I want my column to be three columns wide, I use `col-3`.
2. When I use classes at the row `div` level, I just specify the number of columns I want in the row. Then Bootstrap will automatically calculate the width of each column.

3. Obviously, when I use row classes, I set the width for all the columns of the row to an equal size. If I want to have columns with different widths within the same row, I have to use the column classes at the column-level `div`.

Vertical Alignment

You have learned how to lay out things using a 12-column-wide grid, but Bootstrap allows you to align things inside the row itself, in the vertical axis.

Middle Alignment

How would you put your content in the middle of a tall row like it is depicted in Figure 2-4?

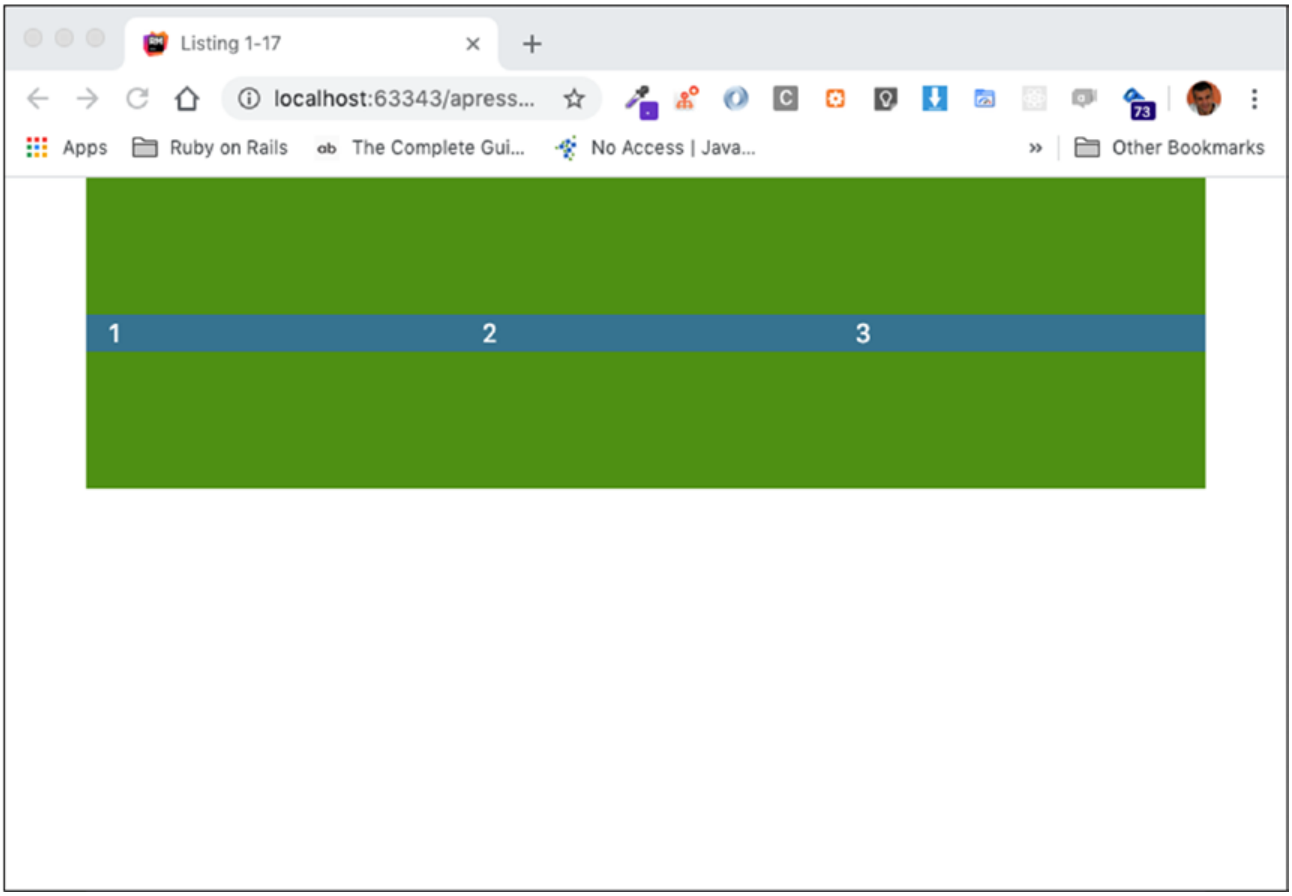

Figure 2-4. *Content in the Middle of a Tall Row*

In the preceding picture, you can see the content of the row, which, in fact, is composed of three columns, to be placed in the center of the vertical axis. In Listing 2-3, you can see how you can create this page.

Listing 2-3. Content in the Middle of a Tall Row

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-3</title>
</head>
<body>
    <div class="container">
        <div class="row align-items-center">
            <div class="col">
                1
            </div>
            <div class="col">
                2
            </div>
            <div class="col">
                3
            </div>
        </div>
    </div>
```

```
    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

The difference is done by the class `align-items-center` at the `row div`. It instructs the browser to draw the columns that are children of this `div` at the center position of the row on the vertical axis.

Before you save and load this page on your browser though, you will have to give your row some minimum height; otherwise, you will not see any difference. Use Listing 2-4 for your `index.css` file.

Listing 2-4. CSS File That Gives the Row a Minimum Height

```
.row {
    background-color: #4e9013;
    color: white;
    min-height: 200px;
}

.col {
    background-color: #31907c;
}
```

The preceding CSS has the rule to apply `min-height` on the row `divs`. This will allow you to see the contents of the children `divs` in the center of the row on the vertical axis direction.

Top and Bottom Alignment

There are also two other classes that you can apply at the row `div` level and that have to do with vertical alignment: `align-items-start` and `align-items-end`. Let's use these to create the page depicted in Figure 2-5.

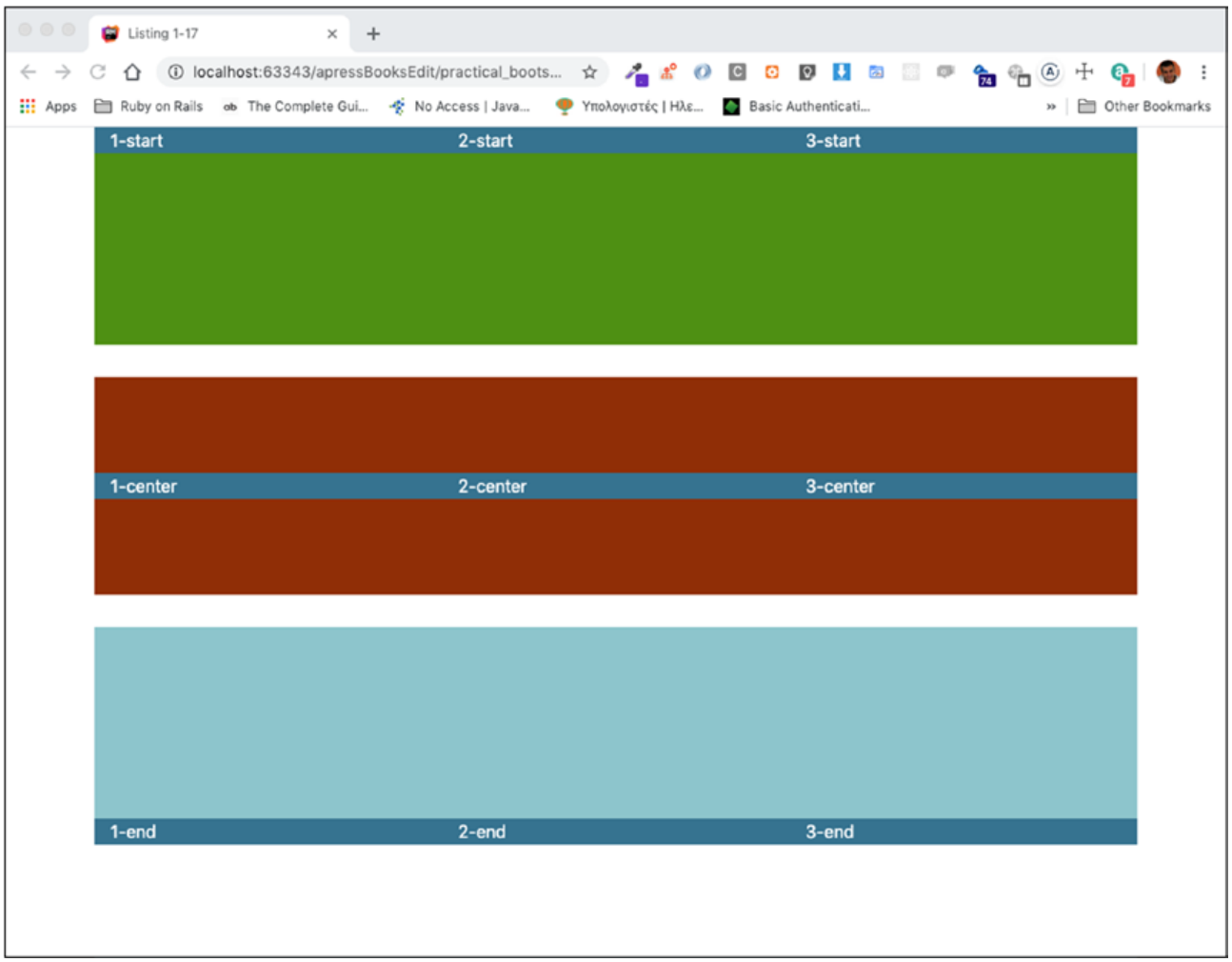

Figure 2-5. *Use Various Vertical Alignments*

As you can see, the middle and bottom rows have alignments `center` and `end`, respectively. I am pretty sure that you write the HTML without my help, but let's include here the code for completeness (Listing 2-5).

Listing 2-5. Align Top, Middle, and Bottom

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-5</title>
</head>
<body>
    <div class="container">
        <div class="row align-items-start">
            <div class="col">
                1-start
            </div>
            <div class="col">
                2-start
            </div>
            <div class="col">
                3-start
            </div>
        </div>
        <div class="row align-items-center">
            <div class="col">
                1-center
```

```
            </div>
            <div class="col">
                2-center
            </div>
            <div class="col">
                3-center
            </div>
        </div>
        <div class="row align-items-end">
            <div class="col">
                1-end
            </div>
            <div class="col">
                2-end
            </div>
            <div class="col">
                3-end
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Don't forget to make your `index.css` have rules that will make your rows have minimum height so that you can see the vertical alignment. Listing 2-6 contains such rules.

Listing 2-6. index.css File That Gives Rows Minimum Height

```
.row {
    min-height: 200px;
    margin-bottom: 30px;
    color: white;
}

.align-items-start {
    background-color: #4e9013;
}

.align-items-center {
    background-color: #902e07;
}

.align-items-end {
    background-color: #8DC5CC;
}

.col {
    background-color: #367390;
}
```

Different Vertical Alignments Within the Same Row

Finally, sometimes you want the columns within the same row to have different vertical alignments. Figure 2-6 is an example of such a page.

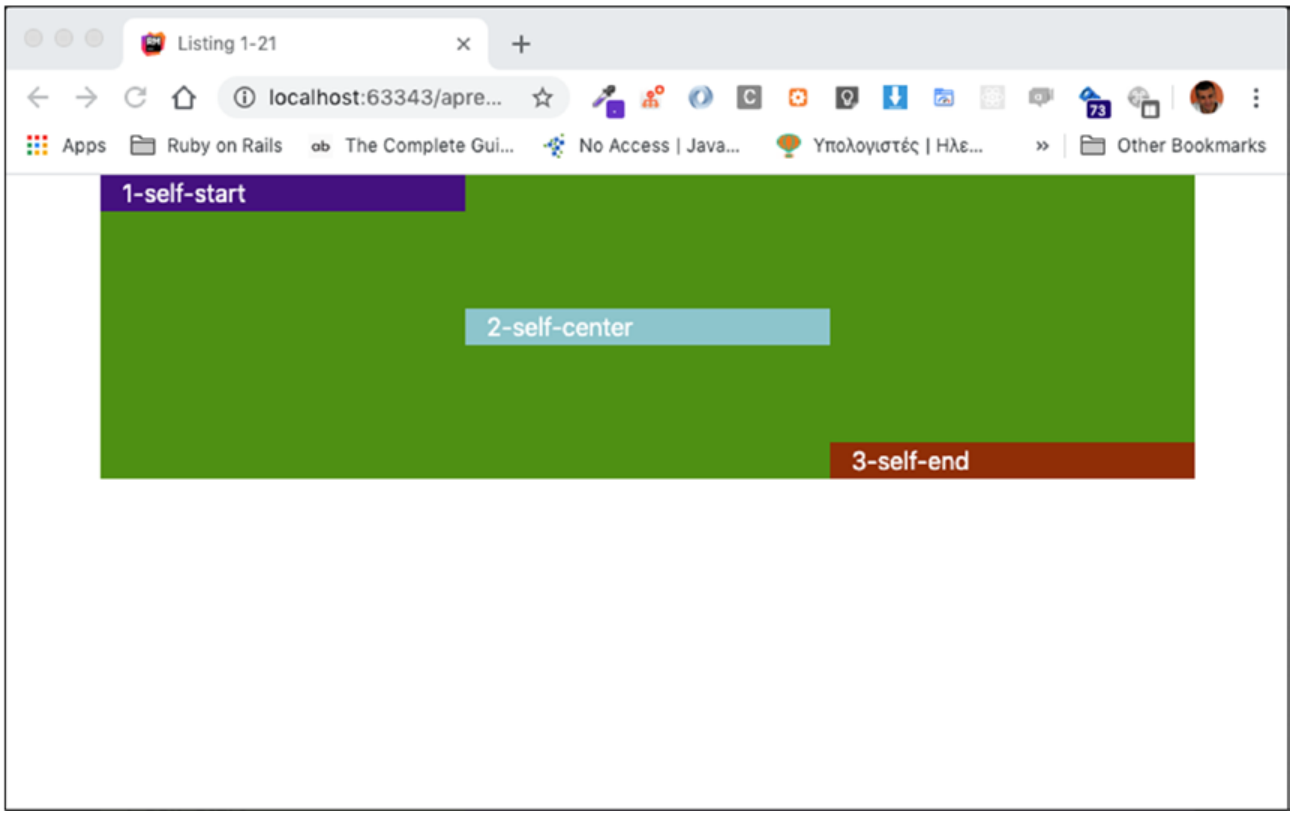

Figure 2-6. *Use Different Vertical Alignments Within the Same Row*

In order to achieve this, I use some special classes at the `col` level. Here is the HTML code for the preceding page (Listing 2-7).

Listing 2-7. Different Vertical Alignments Within the Same Row

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">
```

```
    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-7</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col align-self-start">
                1-self-start
            </div>
            <div class="col align-self-center">
                2-self-center
            </div>
            <div class="col align-self-end">
                3-self-end
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Use the following `index.css` (Listing 2-8) in order to style the preceding HTML content so that you can get the result presented in Figure 2-6.

Listing 2-8. CSS for Listing 2-7

```
.row {
    background-color: #4e9013;
    min-height: 200px;
    margin-bottom: 30px;
    color: white;
}

.align-self-start {
    background-color: #441180;
}

.align-self-center {
    background-color: #8DC5CC;
}

.align-self-end {
    background-color: #902e07;
}
```

Less Than 12 and Horizontal Alignment

I continue by explaining to you how you can align columns in the horizontal direction, which is more useful when you have content that does not occupy the whole width of the page.

Left Alignment

Until now, your rows had content left to right. They occupied the whole container width. There are cases in which you want to have part of the width occupied. For example, you want to have content on the first two columns and the rest of the horizontal space to be empty. Look at the following example in Figure 2-7.

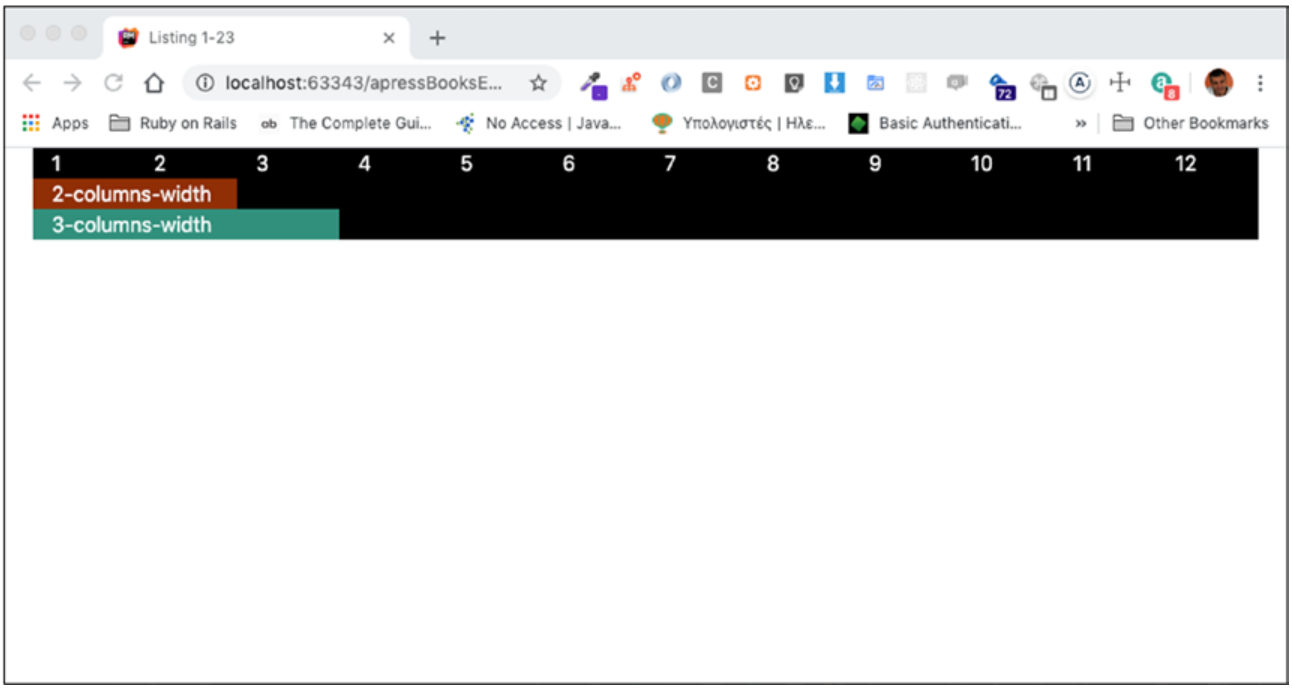

Figure 2-7. *Left-Aligned Columns*

As you can see in Figure 2-7, the second and third rows have columns that occupy the left side of the grid. The second row occupies the first two columns, and the third row occupies the first three columns. How can you achieve this? The HTML code is given in Listing 2-9.

Listing 2-9. Left-Aligned Columns

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">
```

```
    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-9</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">1</div>
            <div class="col">2</div>
            <div class="col">3</div>
            <div class="col">4</div>
            <div class="col">5</div>
            <div class="col">6</div>
            <div class="col">7</div>
            <div class="col">8</div>
            <div class="col">9</div>
            <div class="col">10</div>
            <div class="col">11</div>
            <div class="col">12</div>
        </div>
        <div class="row justify-content-start row-1">
            <div class="col-2 col">
                2-columns-width
            </div>
        </div>
        <div class="row justify-content-start row-2">
            <div class="col-3 col">
                3-columns-width
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

```
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Before loading the preceding HTML into your browser, make sure that you have correct CSS rules inside your `index.css` file, something like that in Listing 2-10.

Listing 2-10. CSS for Listing 2-9

```
* {
    color: white;
}

.row {
    background-color: black;
}

.row-1 .col {
    background-color: #902e07;
}

.row-2 .col {
    background-color: #31907c;
}
```

In Listing 2-9, the class `justify-content-start` is attached on the second and third row `div`s.

Center and Right Alignment

But this, in fact, was not very useful. The same would have been accomplished even if you didn't have this class attached to the row divs. Things are getting more interesting when you use the corresponding *-center and *-end classes like in Listing 2-11.

Listing 2-11. Use *-center and *-end Classes

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-11</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">1</div>
            <div class="col">2</div>
            <div class="col">3</div>
            <div class="col">4</div>
            <div class="col">5</div>
            <div class="col">6</div>
            <div class="col">7</div>
            <div class="col">8</div>
```

```
            <div class="col">9</div>
            <div class="col">10</div>
            <div class="col">11</div>
            <div class="col">12</div>
        </div>
        <div class="row justify-content-center row-1">
            <div class="col-2 col">
                2-columns-width
            </div>
        </div>
        <div class="row justify-content-end row-2">
            <div class="col-3 col">
                3-columns-width
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Save the preceding page and load it on your browser. You will see something like what is depicted in Figure 2-8.

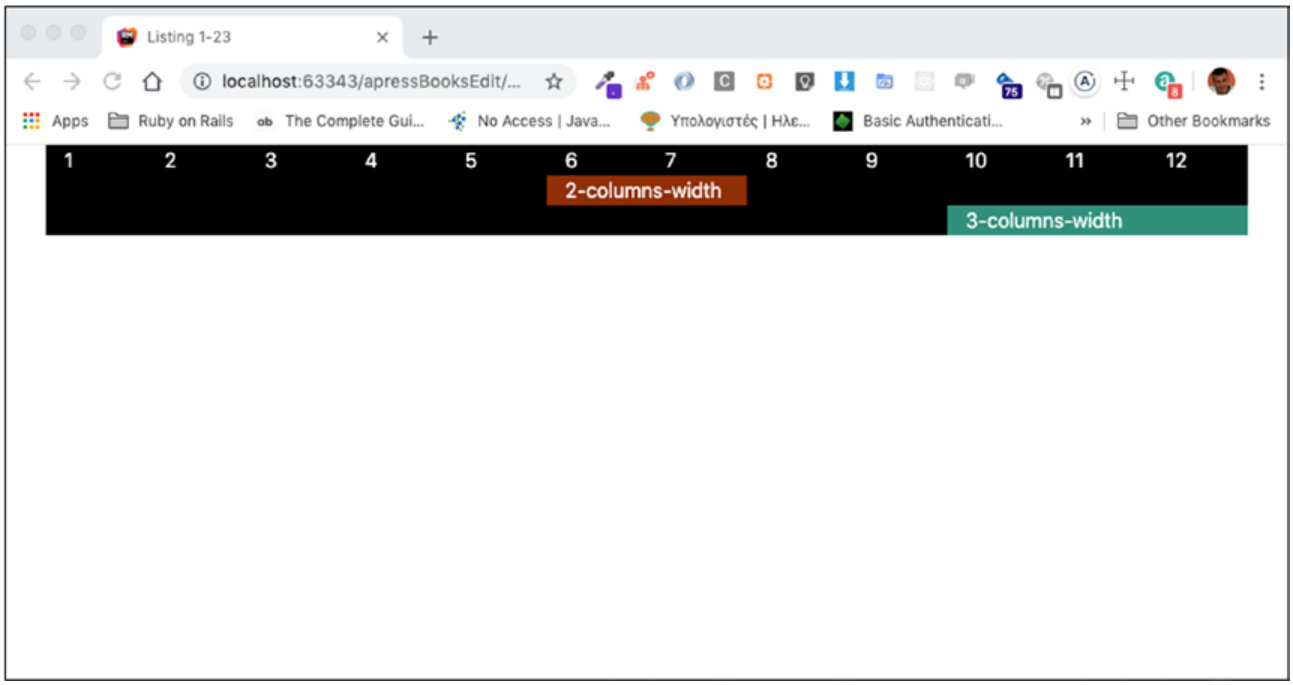

Figure 2-8. *Center and Right Alignment*

Hence, these `justify-content-*` classes at the row `divs` are useful, because they allow you to align without having to use empty divs.

Space Between

What if you had three content columns that you wanted horizontally aligned in such a way that they had equal space between them? Figure 2-9 has an example of such a case.

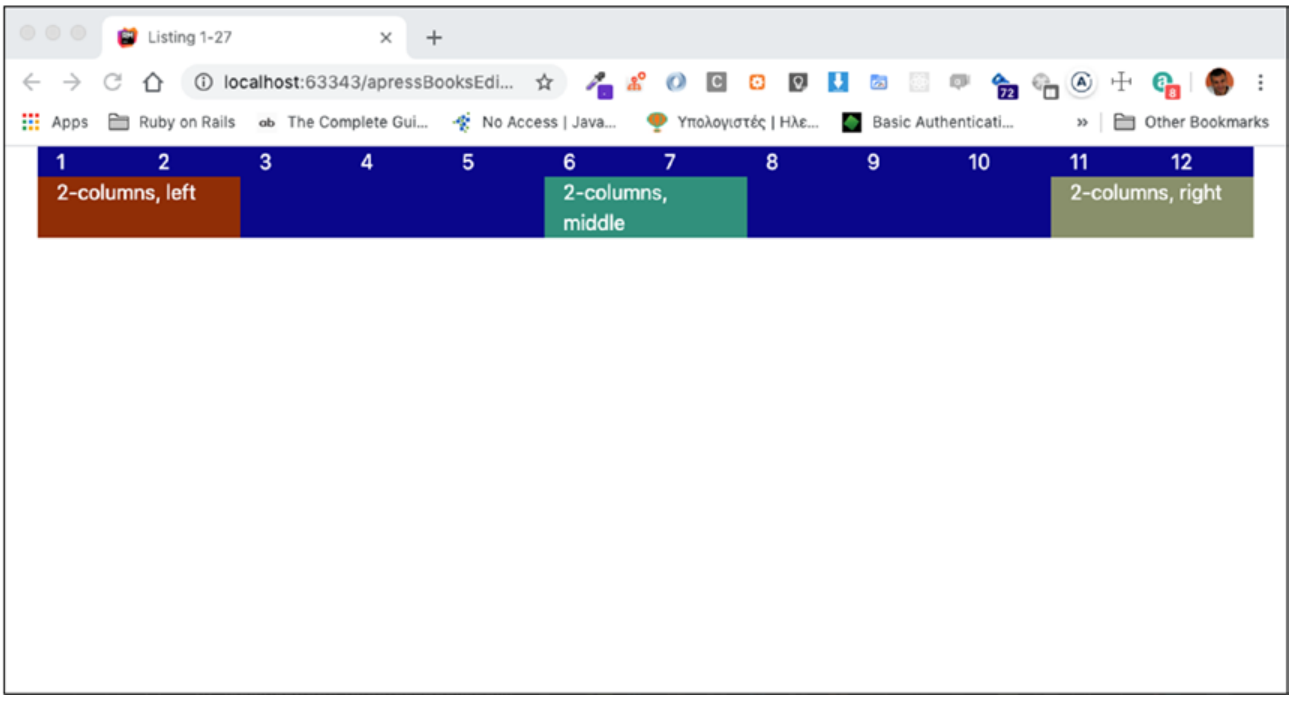

Figure 2-9. *Columns with Space Between*

The class that you have to use in order to achieve is the `justify-content-between`. The HTML code is given in Listing 2-12.

Listing 2-12. Using `justify-content-between`

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">
```

```
    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-12</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">1</div>
            <div class="col">2</div>
            <div class="col">3</div>
            <div class="col">4</div>
            <div class="col">5</div>
            <div class="col">6</div>
            <div class="col">7</div>
            <div class="col">8</div>
            <div class="col">9</div>
            <div class="col">10</div>
            <div class="col">11</div>
            <div class="col">12</div>
        </div>
        <div class="row justify-content-between">
            <div class="col-2">
                2-columns, left
            </div>
            <div class="col-2">
                2-columns, middle
            </div>
            <div class="col-2">
                2-columns, right
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

```
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Space Around

Finally, sometimes you want the space to be around your content. See an example in Figure 2-10.

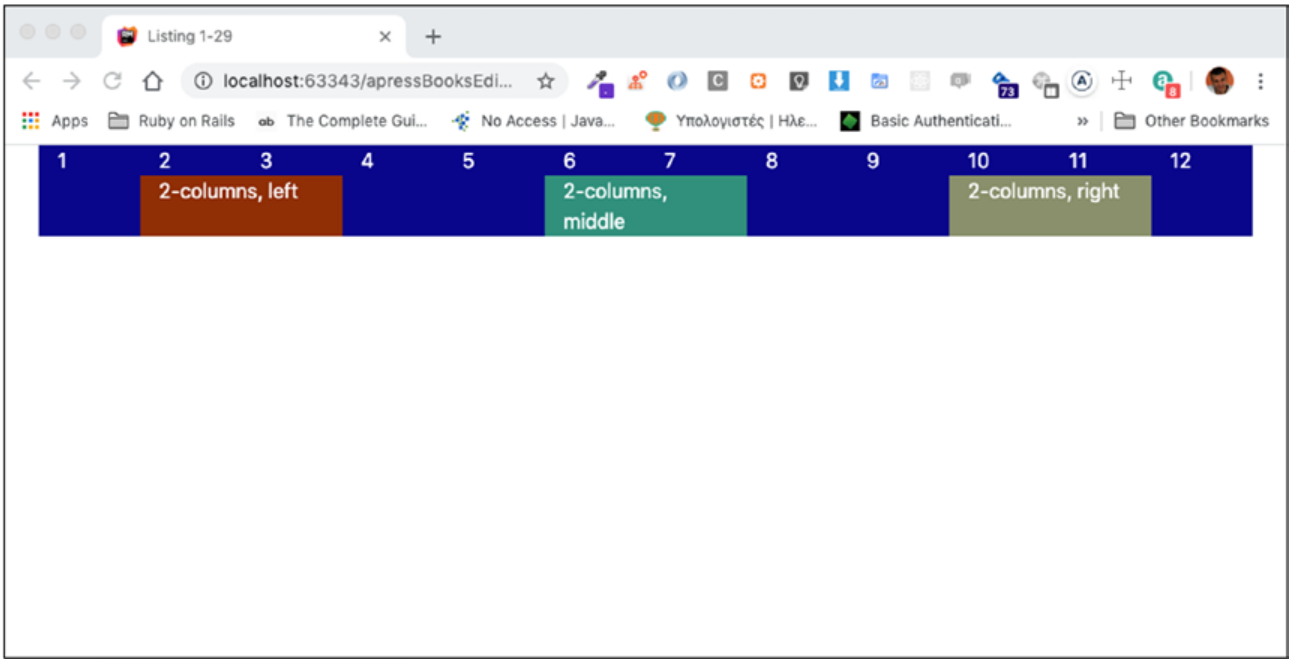

Figure 2-10. *Content with Space Around*

In order to achieve this, you need to use the class `justify-content-around`. Attach it at the `row div` level, as in Listing 2-13.

Listing 2-13. Content with Space Around

```
<!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1,
    shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
    Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
    crossorigin="anonymous">

    <!-- Custom CSS -->
    <link rel="stylesheet" href="stylesheets/index.css" type="text/css">

    <title>Listing 2-13</title>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">1</div>
            <div class="col">2</div>
            <div class="col">3</div>
            <div class="col">4</div>
            <div class="col">5</div>
            <div class="col">6</div>
            <div class="col">7</div>
            <div class="col">8</div>
            <div class="col">9</div>
            <div class="col">10</div>
```

```
            <div class="col">11</div>
            <div class="col">12</div>
        </div>
        <div class="row justify-content-around">
            <div class="col-2">
                2-columns, left
            </div>
            <div class="col-2">
                2-columns, middle
            </div>
            <div class="col-2">
                2-columns, right
            </div>
        </div>
    </div>

    <!-- Optional JavaScript -->
    <!-- jQuery first, then Popper.js, then Bootstrap JS -->
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
    integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1
    yYfoRSJoZ+n" crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/
    popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3
    zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/
    bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7Yw
    aYd1iqfktj0Uod8GCExl3Og8ifwB6" crossorigin="anonymous"></script>
</body>
</html>
```

Load the preceding code using an `index.css` file like the one given in Listing 2-14.