LAB 05 – RESPONSIVE WEB DESIGN

I/. Responsive Web Design

1/. Viewport

The viewport is the user's visible area of a web page.

The viewport varies with the device, and will be smaller on a mobile phone than on a computer screen.

Before tablets and mobile phones, web pages were designed only for computer screens, and it was common for web pages to have a static design and a fixed size.

Then, when we started surfing the internet using tablets and mobile phones, fixed size web pages were too large to fit the viewport. To fix this, browsers on those devices scaled down the entire web page to fit the screen.

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

This gives the browser instructions on how to control the page's dimensions and scaling.

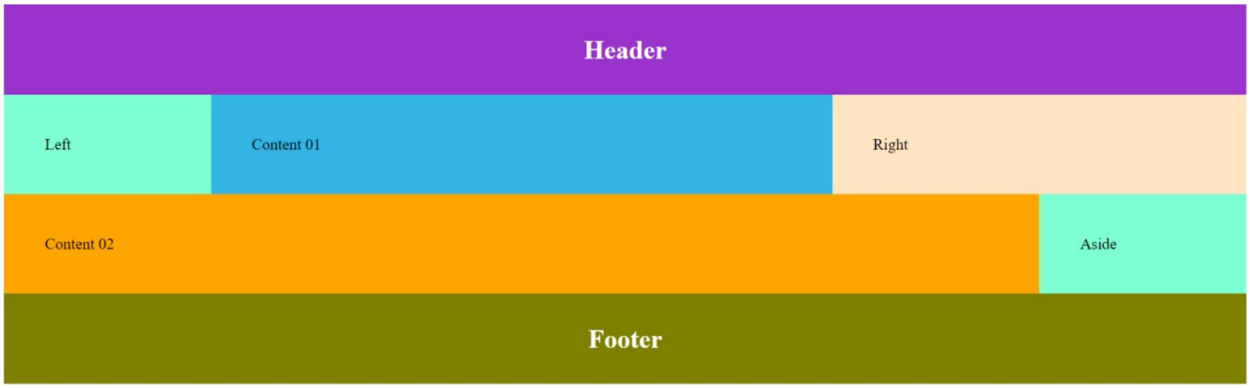
The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

2/. Grid-view

Many web pages are based on a grid-view, which means that the page is divided into columns. Using a grid-view is very helpful when designing web pages. It makes it easier to place elements on the page. A responsive grid-view often has 12 columns, and has a total width of 100%, and will shrink and expand as you resize the browser window.

Example: I have a design like this



Header

Left

Content 01

Right

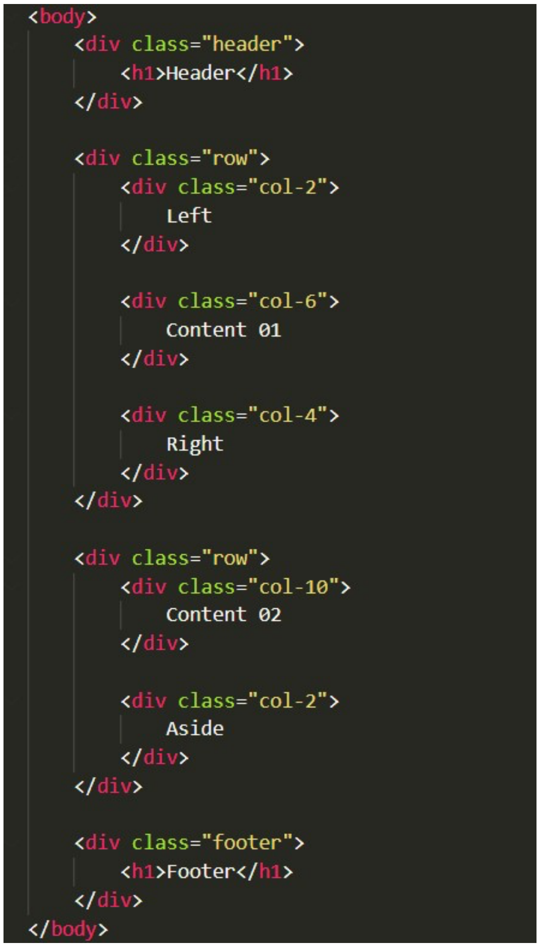
Content 02

Aside

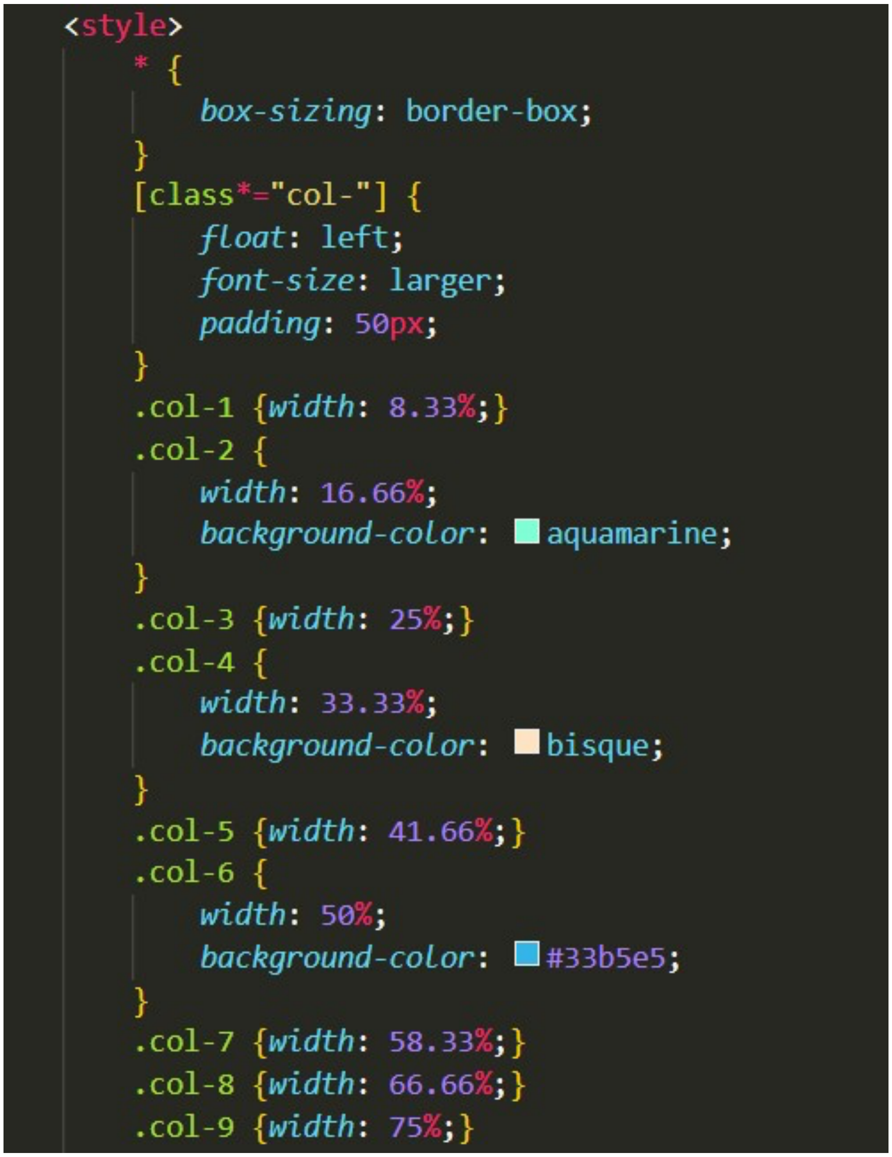
Footer

If you want to build a Responsive Grid-View. How can you do that?

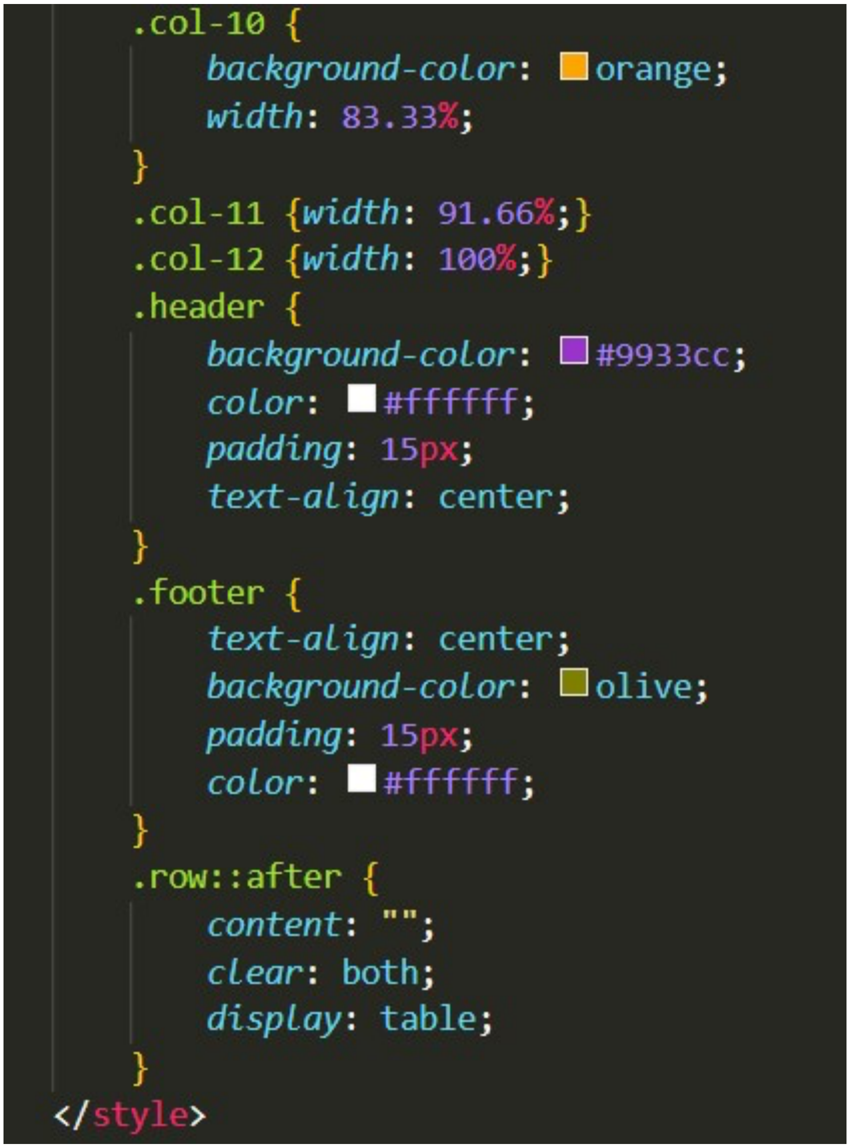
You can try:



```
<body>
    <div class="header">
        <h1>Header</h1>
    </div>

    <div class="row">
        <div class="col-2">
            Left
        </div>

        <div class="col-6">
            Content 01
        </div>

        <div class="col-4">
            Right
        </div>
    </div>

    <div class="row">
        <div class="col-10">
            Content 02
        </div>

        <div class="col-2">
            Aside
        </div>
    </div>

    <div class="footer">
        <h1>Footer</h1>
    </div>
</body>
```



```
    <style>
        * {
            box-sizing: border-box;
        }
        [class*="col-"] {
            float: left;
            font-size: larger;
            padding: 50px;
        }
        .col-1 {width: 8.33%;}
        .col-2 {
            width: 16.66%;
            background-color: aquamarine;
        }
        .col-3 {width: 25%;}
        .col-4 {
            width: 33.33%;
            background-color: bisque;
        }
        .col-5 {width: 41.66%;}
        .col-6 {
            width: 50%;
            background-color: #33b5e5;
        }
        .col-7 {width: 58.33%;}
        .col-8 {width: 66.66%;}
        .col-9 {width: 75%;}
```



```
    .col-10 {
        background-color: orange;
        width: 83.33%;
    }
    .col-11 {width: 91.66%;}
    .col-12 {width: 100%;}
    .header {
        background-color: #9933cc;
        color: #ffffff;
        padding: 15px;
        text-align: center;
    }
    .footer {
        text-align: center;
        background-color: olive;
        padding: 15px;
        color: #ffffff;
    }
    .row::after {
        content: "";
        clear: both;
        display: table;
    }
</style>
```

3/. Media Query

Media query is a CSS technique introduced in CSS3.

It uses the @media rule to include a block of CSS properties only if a certain condition is true. a/. Add a Breakpoint

- Earlier in this tutorial we made a web page with rows and columns, and it was responsive, but it did not look good on a small screen.

- Media queries can help with that. We can add a breakpoint where certain parts of the design will behave differently on each side of the breakpoint.

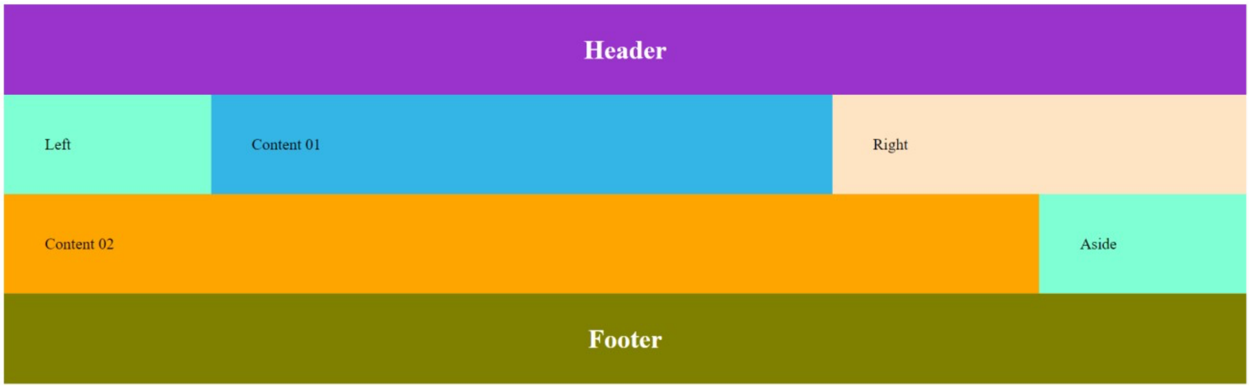
- Always Design for Mobile First

Mobile First means designing for mobile before designing for desktop or any other device (This will make the page display faster on smaller devices).

Syntax:

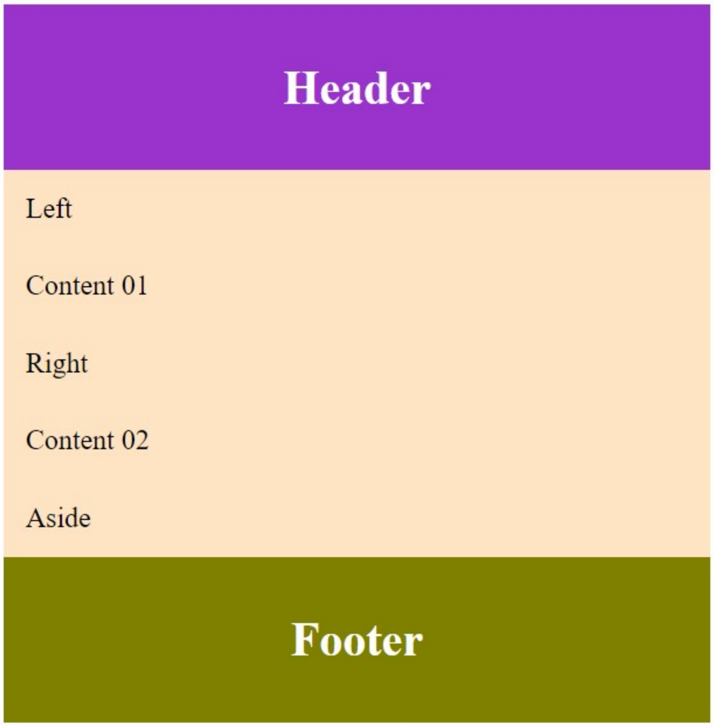
```
// For mobile first

[class*="col-"] { width:

        100%;

}

@media only screen and (min-width: <value1>) {

        ……..

}

@media only screen and (min-width: <value2>) {

        ……..

}
```

Example: *I have a design like this*

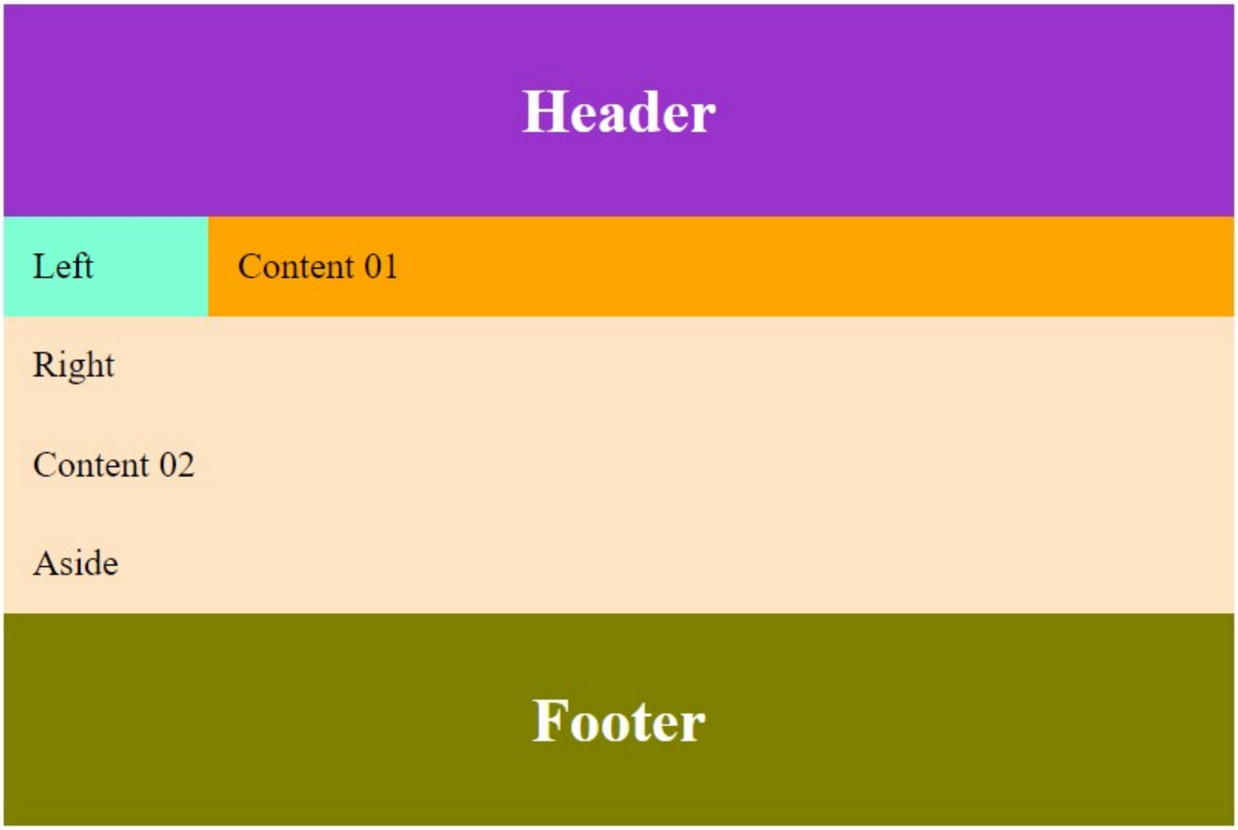


It will show on desktop (min-width is 768px). I want to use media query to show for tablet and mobile like this:

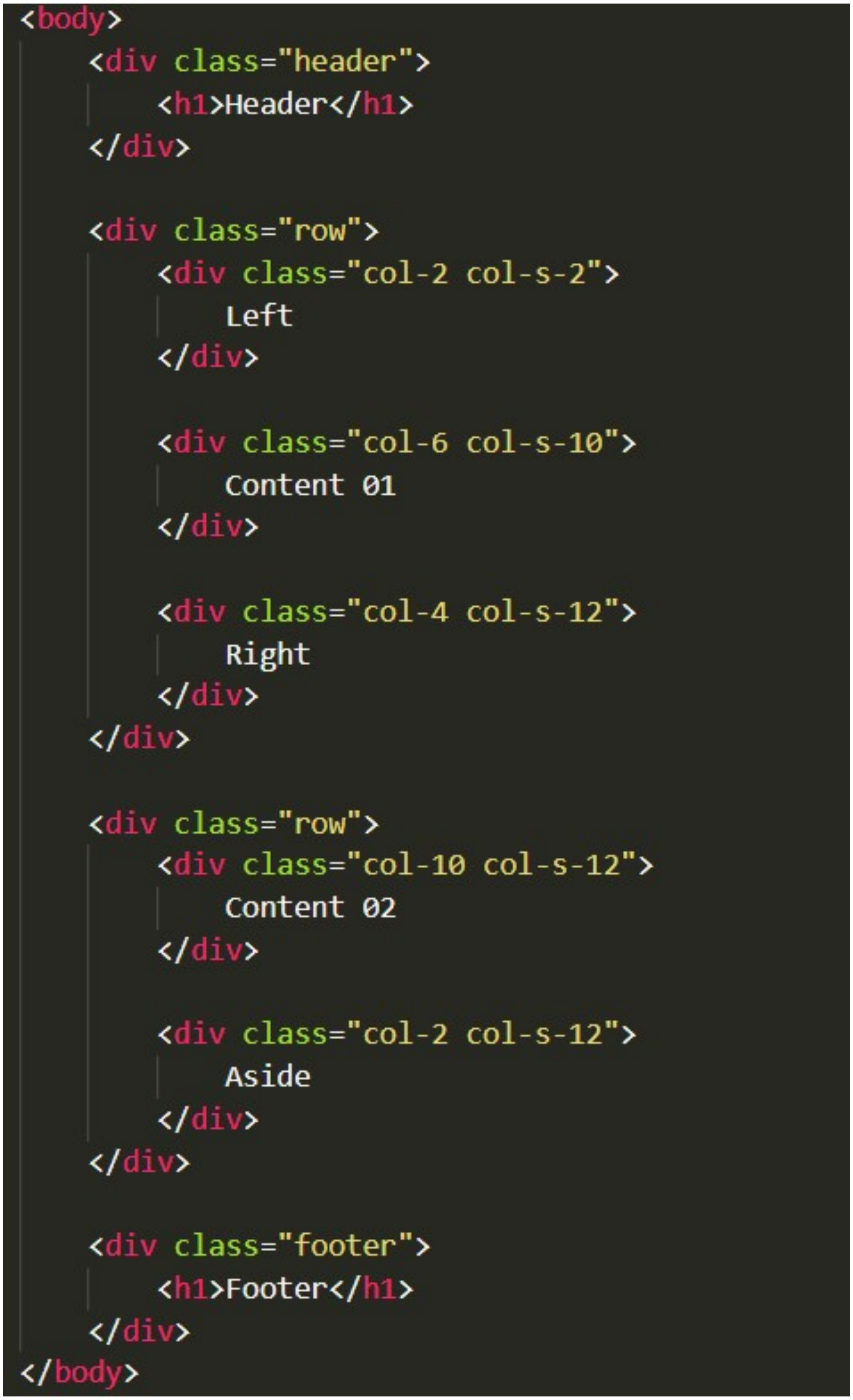
For mobile:



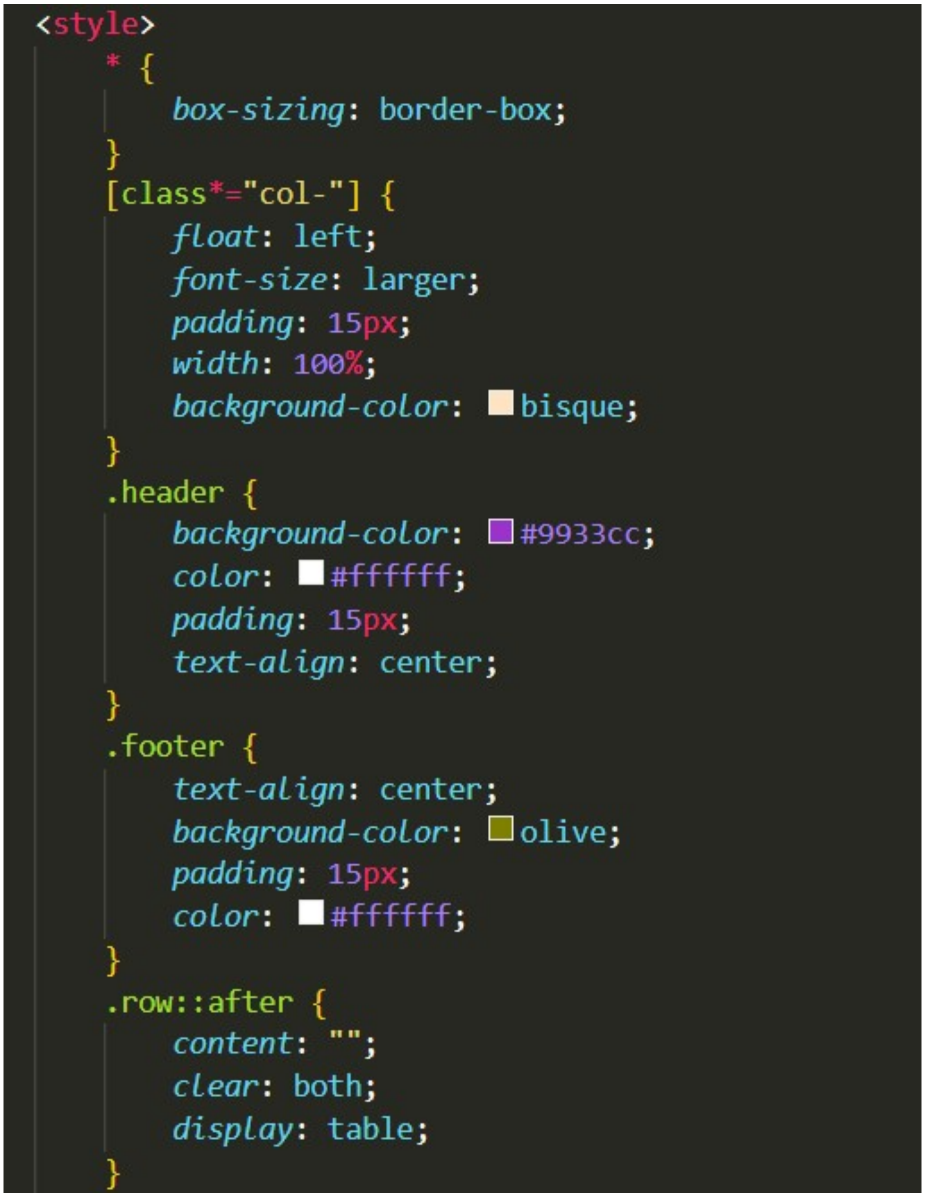
For tablet (min-width is 600px)



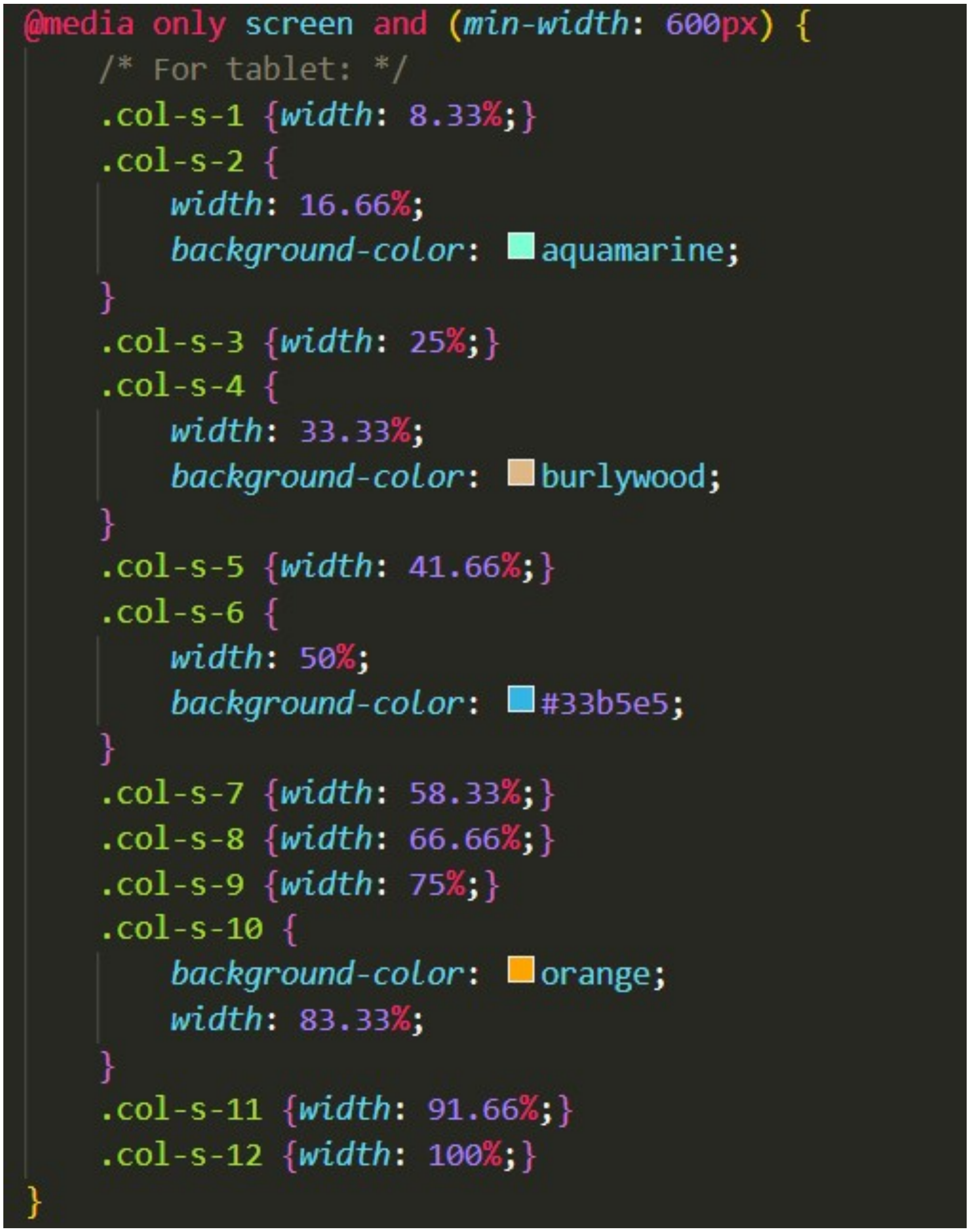
You can try:



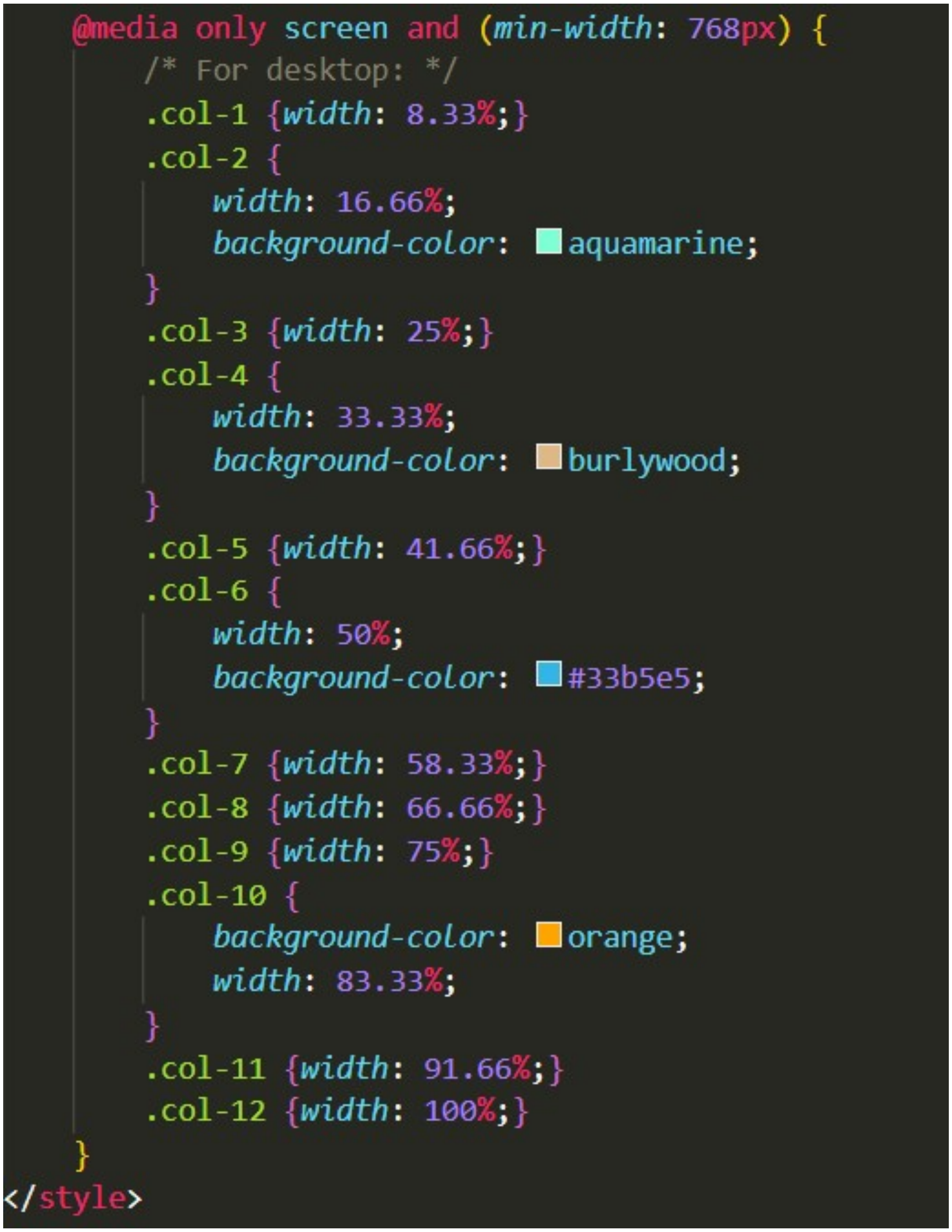
```html
<body>
    <div class="header">
        <h1>Header</h1>
    </div>

    <div class="row">
        <div class="col-2 col-s-2">
            Left
        </div>

        <div class="col-6 col-s-10">
            Content 01
        </div>

        <div class="col-4 col-s-12">
            Right
        </div>
    </div>

    <div class="row">
        <div class="col-10 col-s-12">
            Content 02
        </div>

        <div class="col-2 col-s-12">
            Aside
        </div>
    </div>

    <div class="footer">
        <h1>Footer</h1>
    </div>
</body>
```



```
<style>
    * {
        box-sizing: border-box;
    }
    [class*="col-"] {
        float: left;
        font-size: larger;
        padding: 15px;
        width: 100%;
        background-color: bisque;
    }
    .header {
        background-color: #9933cc;
        color: #ffffff;
        padding: 15px;
        text-align: center;
    }
    .footer {
        text-align: center;
        background-color: olive;
        padding: 15px;
        color: #ffffff;
    }
    .row::after {
        content: "";
        clear: both;
        display: table;
    }
```



```css
@media only screen and (min-width: 600px) {
    /* For tablet: */
    .col-s-1 {width: 8.33%;}
    .col-s-2 {
        width: 16.66%;
        background-color: aquamarine;
    }
    .col-s-3 {width: 25%;}
    .col-s-4 {
        width: 33.33%;
        background-color: burlywood;
    }
    .col-s-5 {width: 41.66%;}
    .col-s-6 {
        width: 50%;
        background-color: #33b5e5;
    }
    .col-s-7 {width: 58.33%;}
    .col-s-8 {width: 66.66%;}
    .col-s-9 {width: 75%;}
    .col-s-10 {
        background-color: orange;
        width: 83.33%;
    }
    .col-s-11 {width: 91.66%;}
    .col-s-12 {width: 100%;}
}
```



```
    @media only screen and (min-width: 768px) {
        /* For desktop: */
        .col-1 {width: 8.33%;}
        .col-2 {
            width: 16.66%;
            background-color: aquamarine;
        }
        .col-3 {width: 25%;}
        .col-4 {
            width: 33.33%;
            background-color: burlywood;
        }
        .col-5 {width: 41.66%;}
        .col-6 {
            width: 50%;
            background-color: #33b5e5;
        }
        .col-7 {width: 58.33%;}
        .col-8 {width: 66.66%;}
        .col-9 {width: 75%;}
        .col-10 {
            background-color: orange;
            width: 83.33%;
        }
        .col-11 {width: 91.66%;}
        .col-12 {width: 100%;}
    }
</style>
```

Typical Device Breakpoints

There are tons of screens and devices with different heights and widths, so it is hard to create an exact breakpoint for each device. To keep things simple you could target five groups:

```
/* Extra small devices (phones, 600px and down) */

@media only screen and (max-width: 600px) {...}

/* Small devices (portrait tablets and large phones, 600px and up) */

@media only screen and (min-width: 600px) {...}

/* Medium devices (landscape tablets, 768px and up) */

@media only screen and (min-width: 768px) {...}

/* Large devices (laptops/desktops, 992px and up) */

@media only screen and (min-width: 992px) {...}

/* Extra large devices (large laptops and desktops, 1200px and up) */

@media only screen and (min-width: 1200px) {...}
```

b/. Orientation: Portrait / Landscape: Media queries can also be used to change layout of a page depending on the orientation of the browser.

Example: The web page will have a lightblue background if the orientation is in landscape mode

```
@media only screen and (orientation: landscape) {

body {

   background-color: lightblue;

 }

}
```

II/. Bootstrap (Bootstrap 5)

Link: https://www.w3schools.com/bootstrap5/index.php

Bootstrap is a free front-end framework for faster and easier web development.

Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many other, as well as optional JavaScript plugins.

Bootstrap also gives you the ability to easily create responsive designs.

There are two ways to start using Bootstrap 5 on your own web site.

You can:

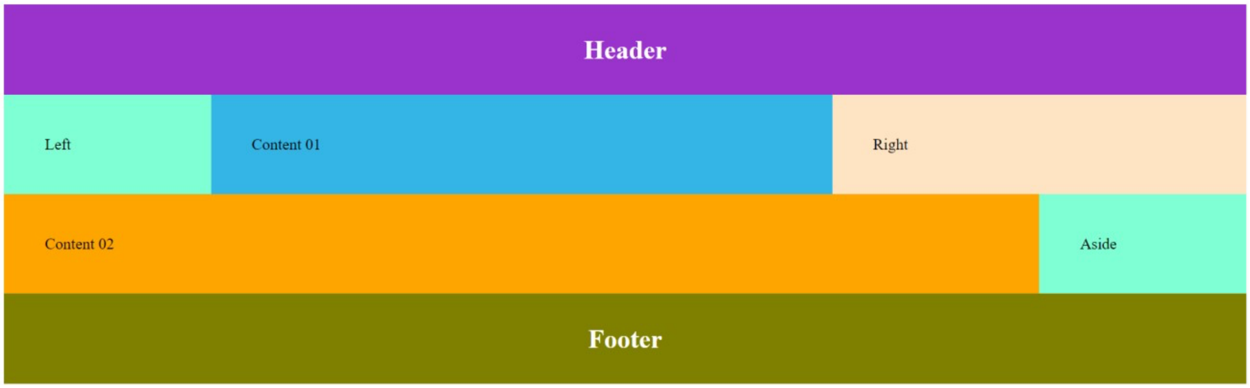
- Include Bootstrap 5 from a CDN
- Download Bootstrap 5 from getbootstrap.com

(*) Layout with Bootstrap

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

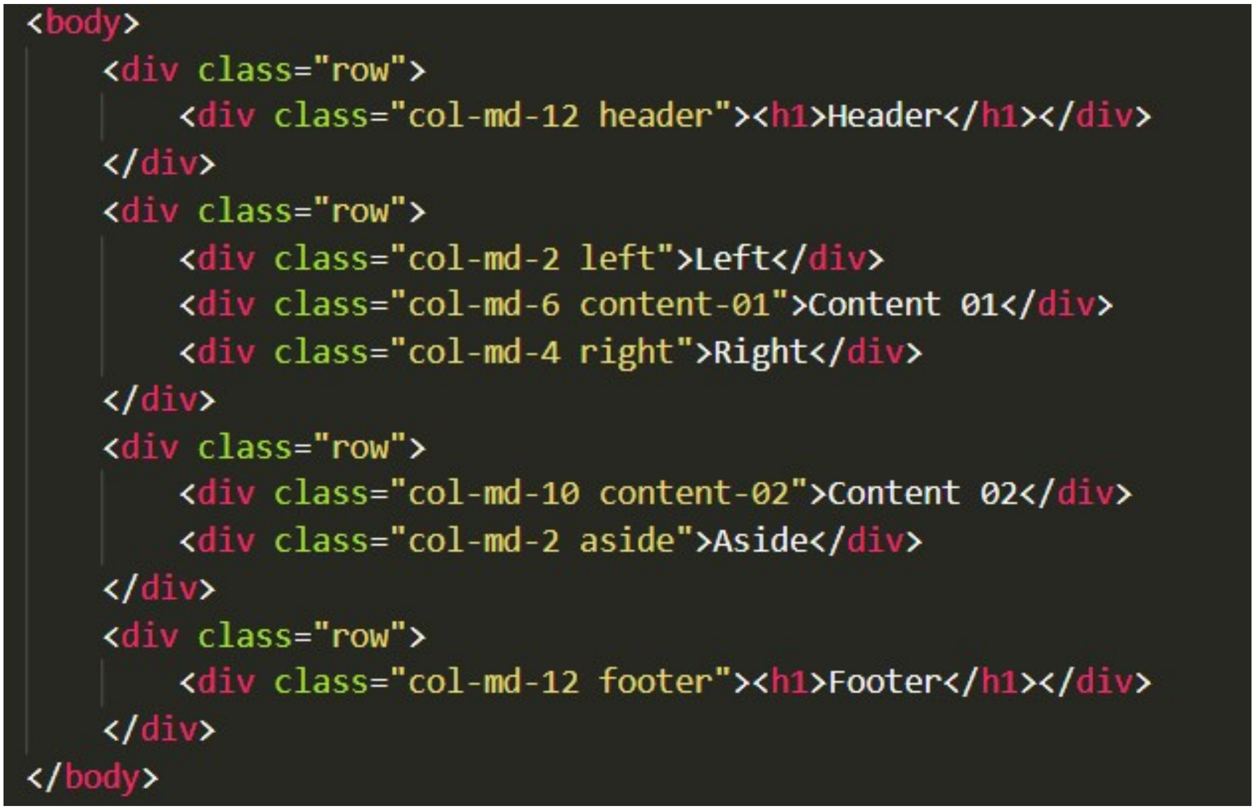
The Bootstrap 5 grid system has six classes:

- .col- (extra small devices - screen width less than 576px)
- .col-sm- (small devices - screen width equal to or greater than 576px)
- .col-md- (medium devices - screen width equal to or greater than 768px)
- .col-lg- (large devices - screen width equal to or greater than 992px)
- .col-xl- (xlarge devices - screen width equal to or greater than 1200px)
- .col-xxl- (xxlarge devices - screen width equal to or greater than 1400px) *Example 01: I have a design like this*

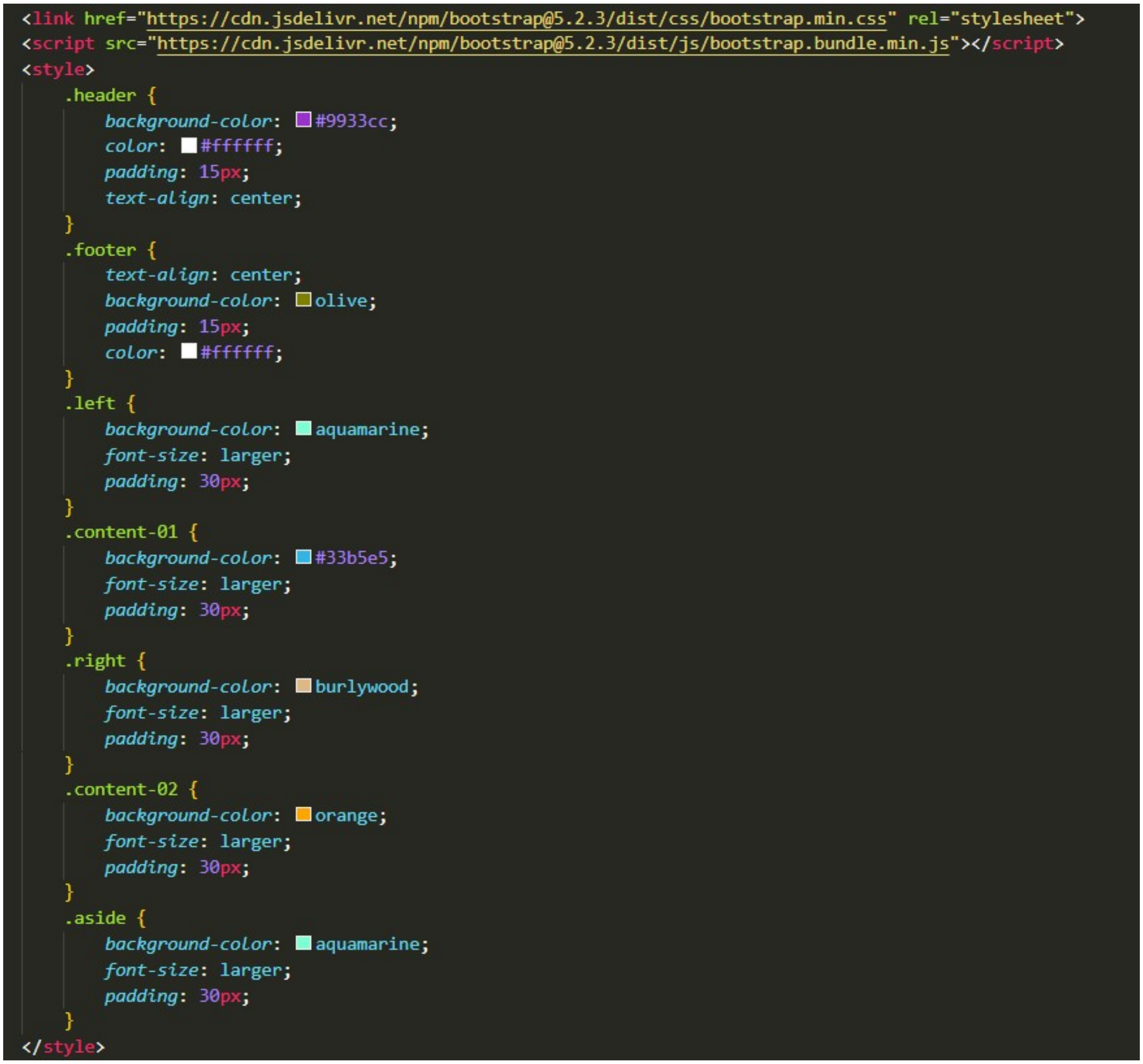


If you use classes in Bootstrap to design. How can you do that?

You can try:



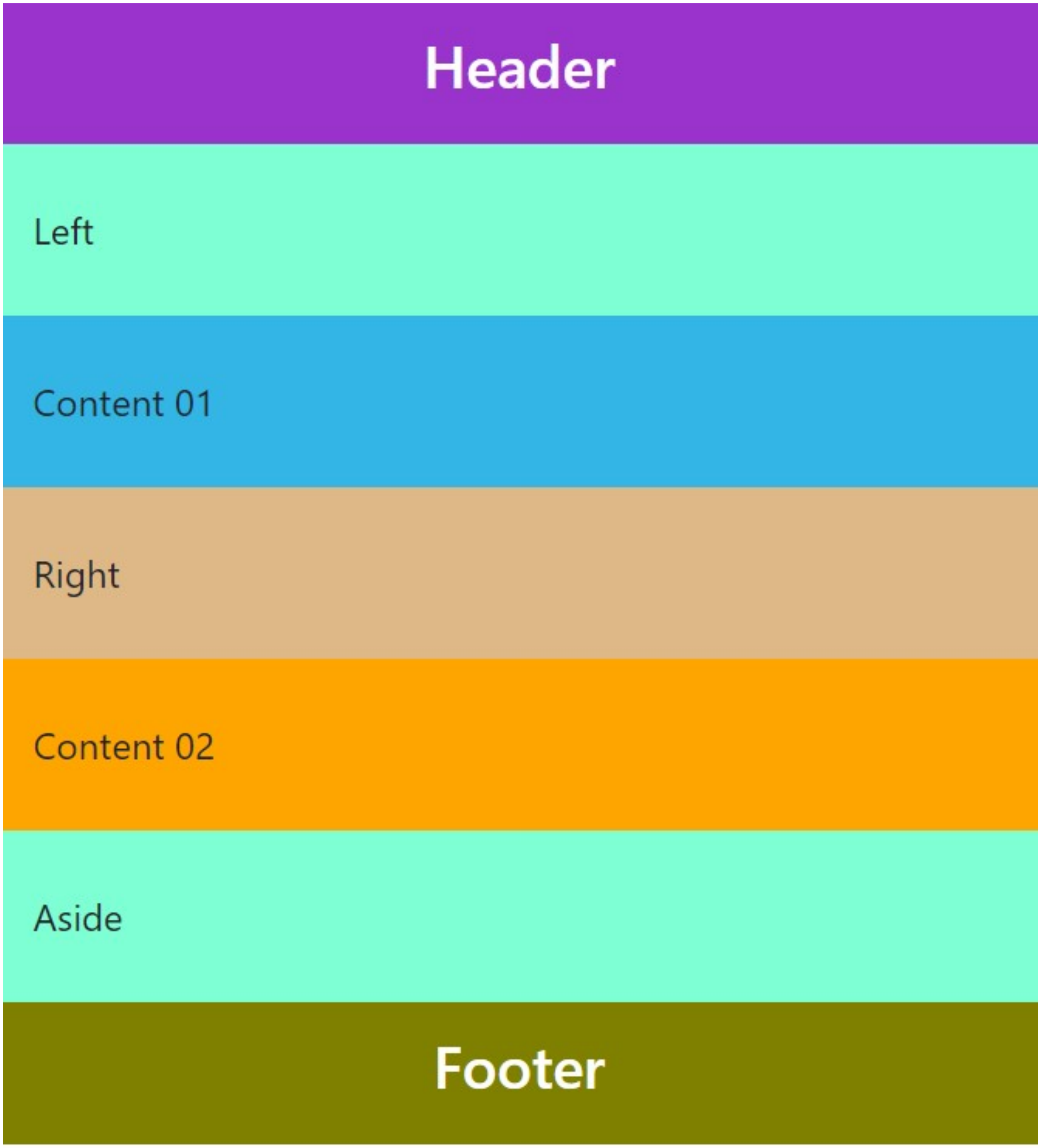
```html
<body>
    <div class="row">
        <div class="col-md-12 header"><h1>Header</h1></div>
    </div>
    <div class="row">
        <div class="col-md-2 left">Left</div>
        <div class="col-md-6 content-01">Content 01</div>
        <div class="col-md-4 right">Right</div>
    </div>
    <div class="row">
        <div class="col-md-10 content-02">Content 02</div>
        <div class="col-md-2 aside">Aside</div>
    </div>
    <div class="row">
        <div class="col-md-12 footer"><h1>Footer</h1></div>
    </div>
</body>
```



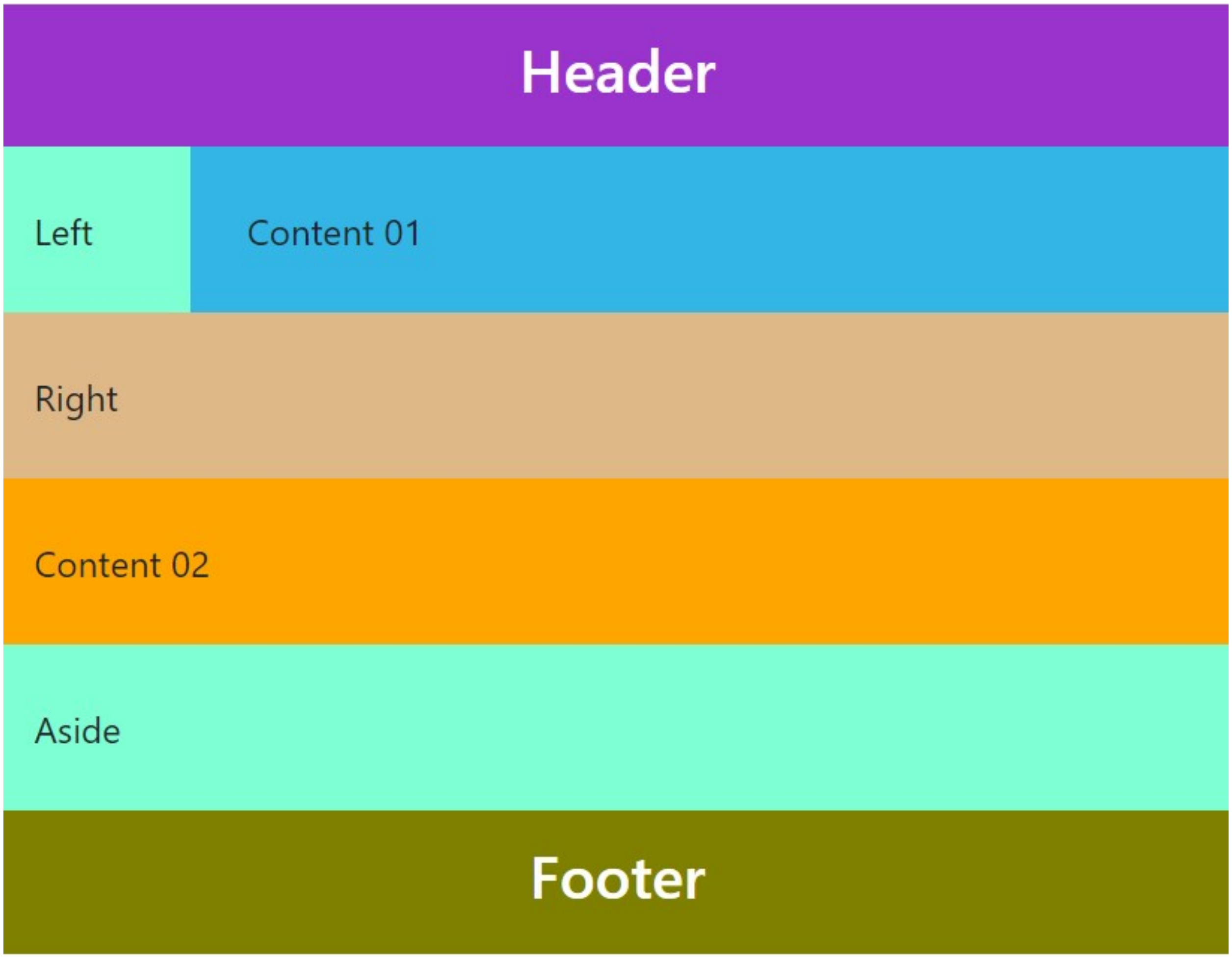
```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet">
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
<style>
    .header {
        background-color: #9933cc;
        color: #ffffff;
        padding: 15px;
        text-align: center;
    }
    .footer {
        text-align: center;
        background-color: olive;
        padding: 15px;
        color: #ffffff;
    }
    .left {
        background-color: aquamarine;
        font-size: larger;
        padding: 30px;
    }
    .content-01 {
        background-color: #33b5e5;
        font-size: larger;
        padding: 30px;
    }
    .right {
        background-color: burlywood;
        font-size: larger;
        padding: 30px;
    }
    .content-02 {
        background-color: orange;
        font-size: larger;
        padding: 30px;
    }
    .aside {
        background-color: aquamarine;
        font-size: larger;
        padding: 30px;
    }
</style>
```

<u>Example 02:</u> If I want to change layout of web page in Exercise 01 for mobile and tablet devices. How can you do that?

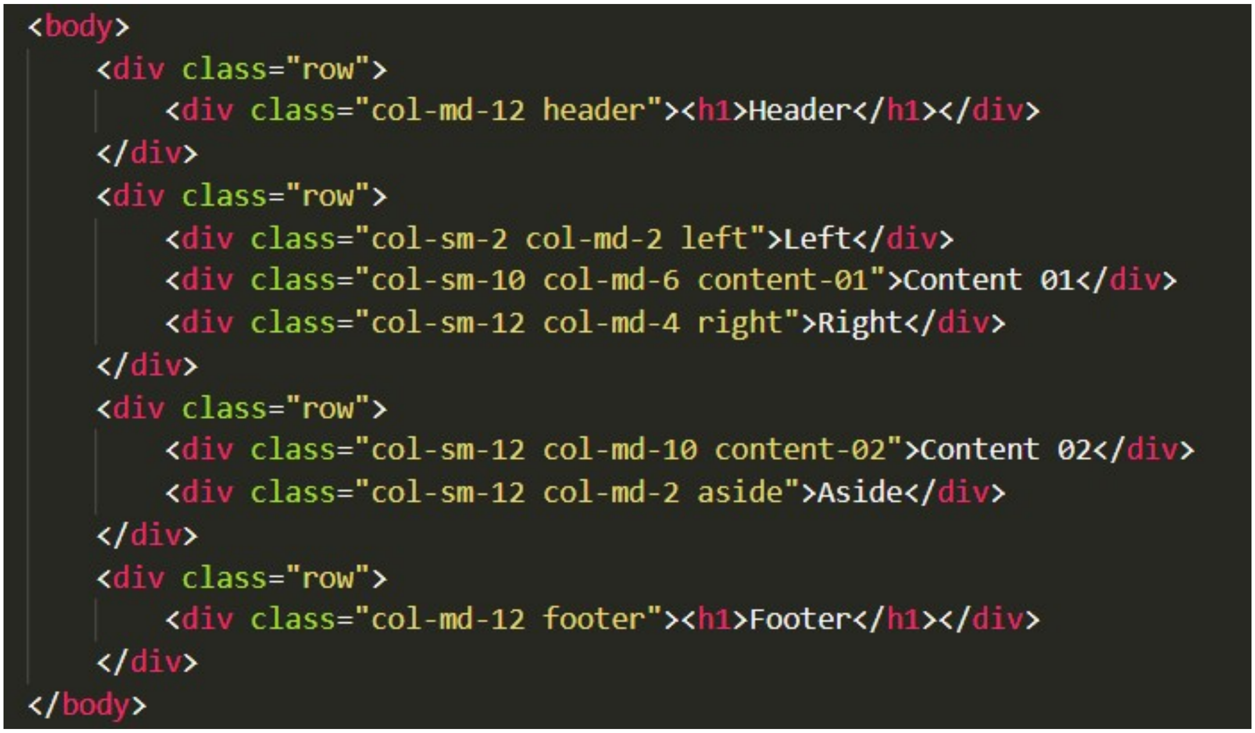
If web page for mobile:



If web page for tablet:



You can try:



```html
<body>
    <div class="row">
        <div class="col-md-12 header"><h1>Header</h1></div>
    </div>
    <div class="row">
        <div class="col-sm-2 col-md-2 left">Left</div>
        <div class="col-sm-10 col-md-6 content-01">Content 01</div>
        <div class="col-sm-12 col-md-4 right">Right</div>
    </div>
    <div class="row">
        <div class="col-sm-12 col-md-10 content-02">Content 02</div>
        <div class="col-sm-12 col-md-2 aside">Aside</div>
    </div>
    <div class="row">
        <div class="col-md-12 footer"><h1>Footer</h1></div>
    </div>
</body>
```