CACS-205 Web Technology

(BCA, TU)

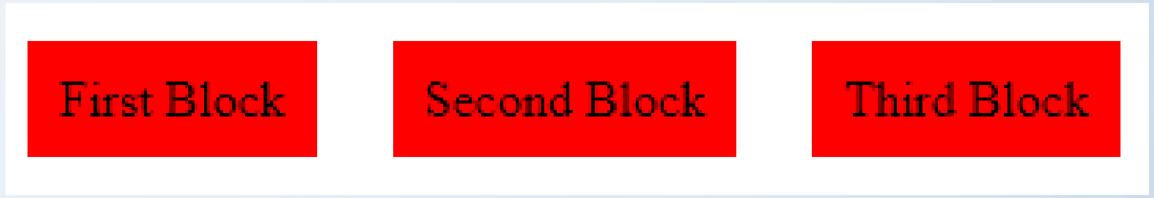
Ganesh Khatri
kh6ganesh@gmail.com

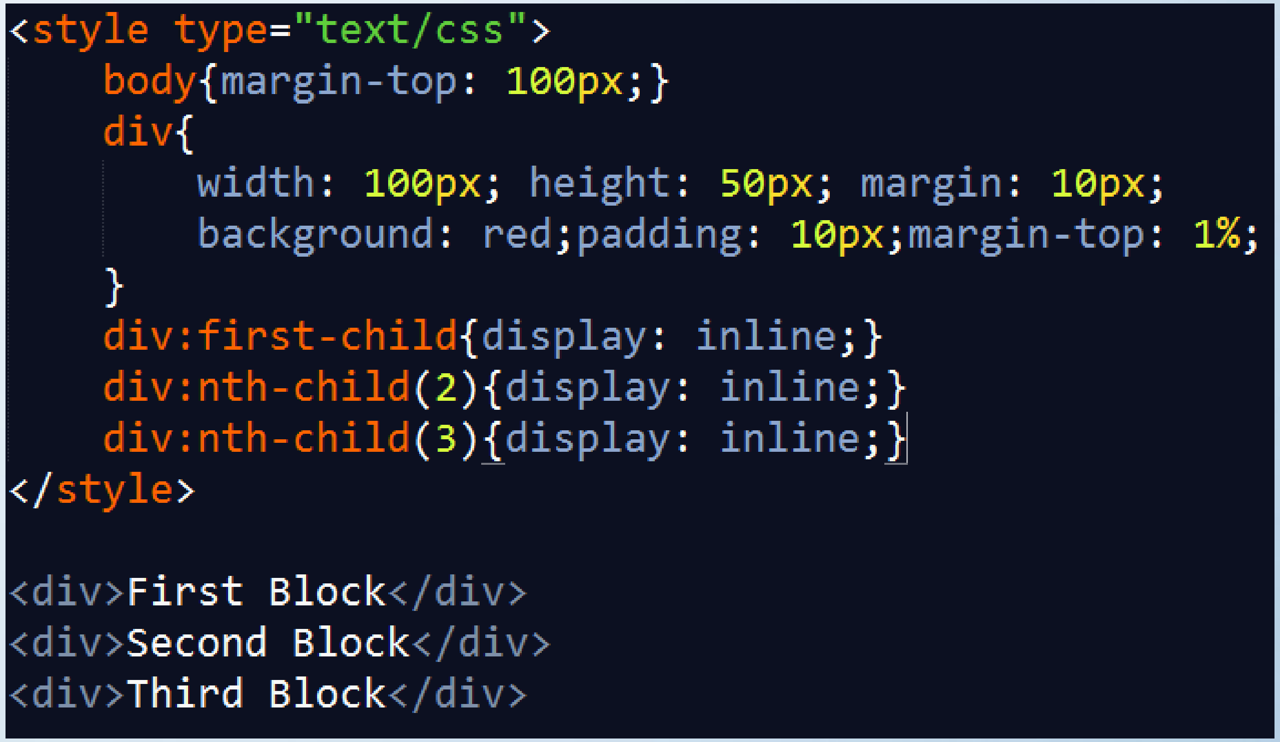
Layouts and Positioning

- Positioning Properties
  - Position
  - Float
  - Display
  - Flex Properties

Inline Elements

- does not start on a new line and only takes up as much width as necessary

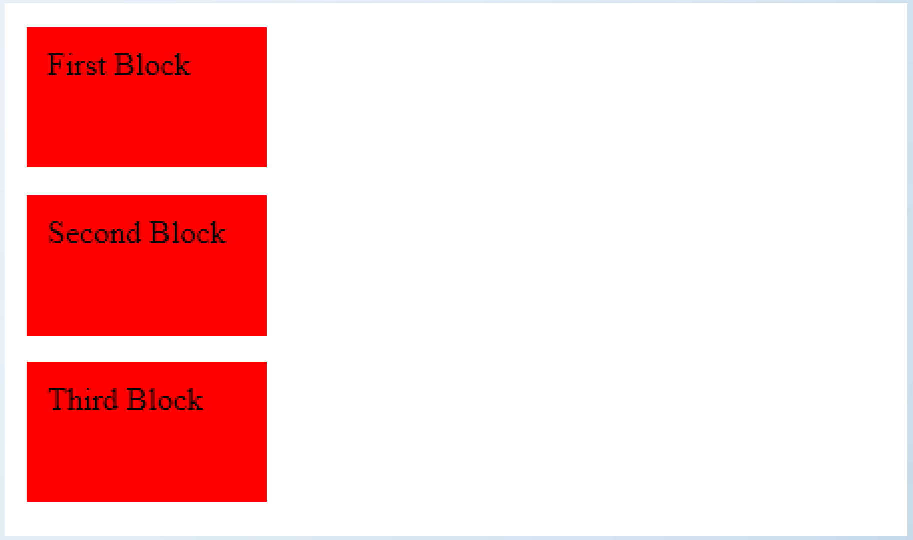


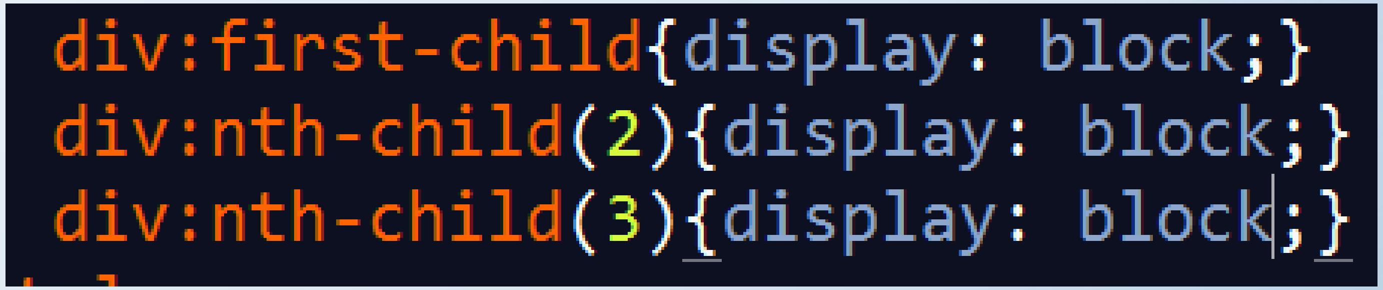


```
<style type="text/css">
    body{margin-top: 100px;}
    div{
        width: 100px; height: 50px; margin: 10px;
        background: red;padding: 10px;margin-top: 1%;
    }
    div:first-child{display: inline;}
    div:nth-child(2){display: inline;}
    div:nth-child(3){display: inline;}
</style>

<div>First Block</div>
<div>Second Block</div>
<div>Third Block</div>
```

Block Elements

- always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can)

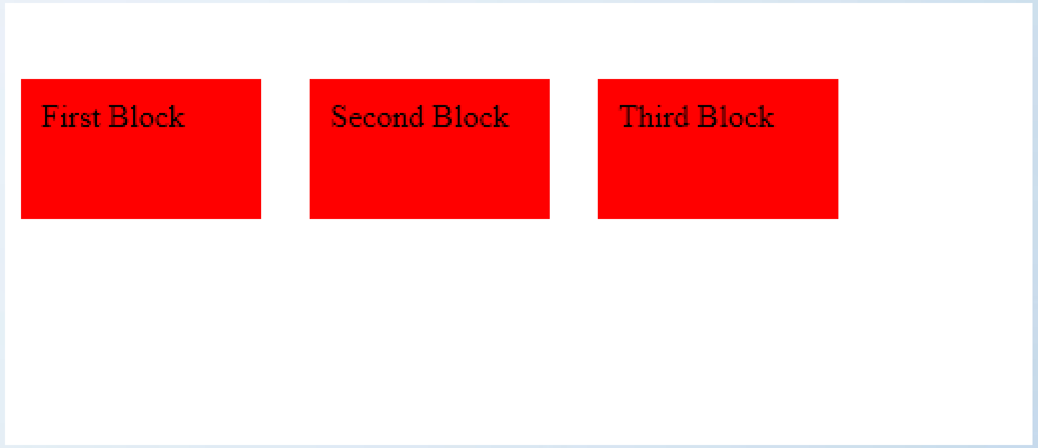


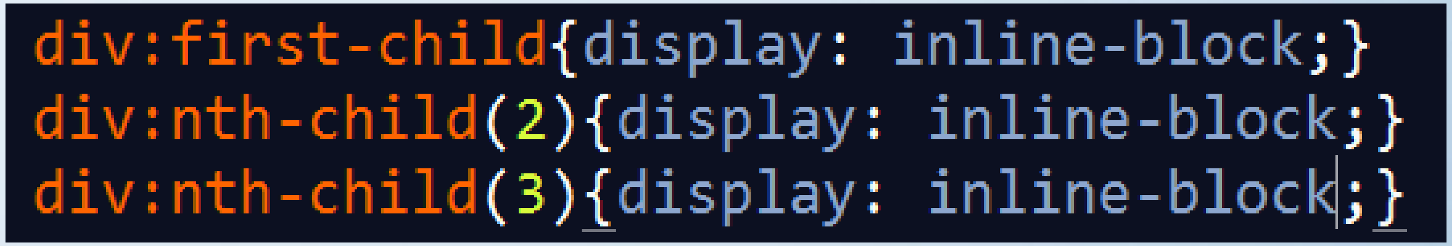


```
div:first-child{display: block;}
div:nth-child(2){display: block;}
div:nth-child(3){display: block;}
```

Inline Block Elements

- Displays an element as an inline-level block container
- The inside of this block is formatted as block-level box, and the element itself is formatted as an inline-level box



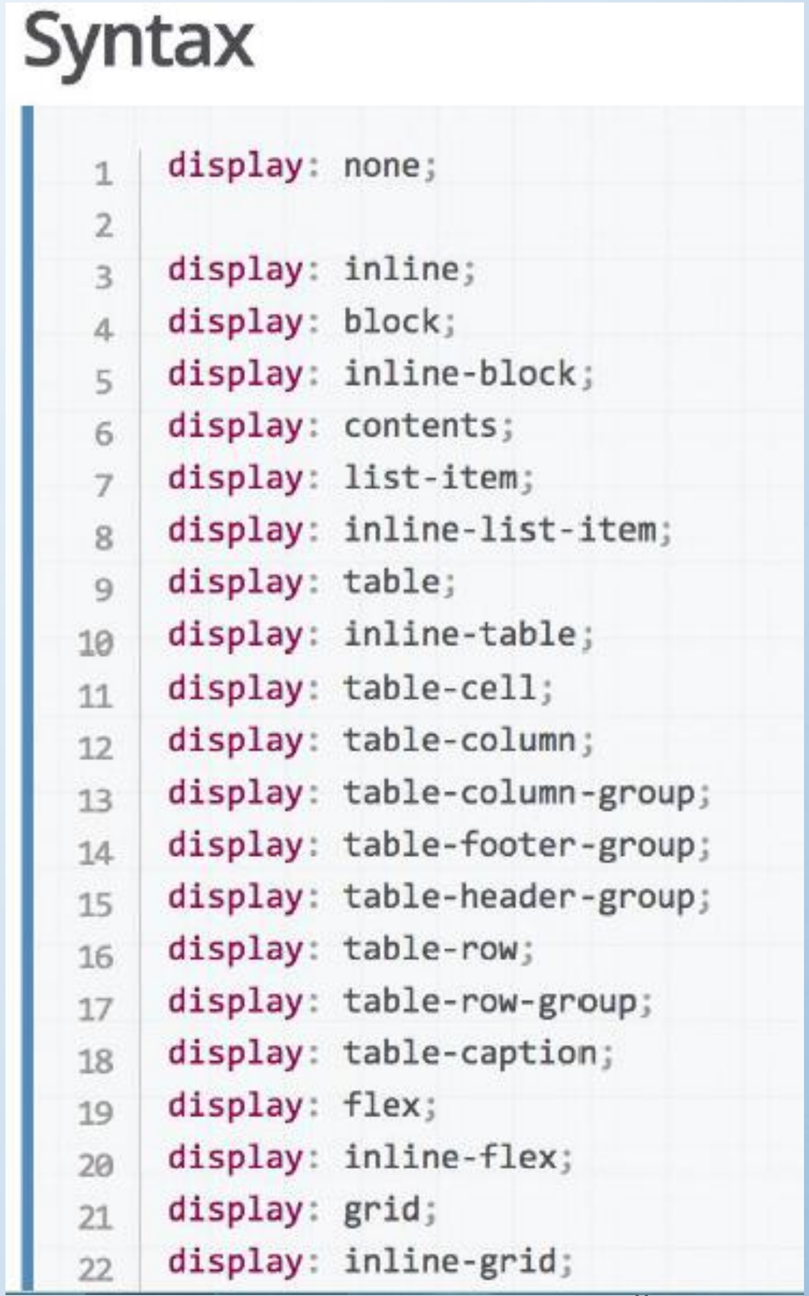


```
div:first-child{display: inline-block;}
div:nth-child(2){display: inline-block;}
div:nth-child(3){display: inline-block;}
```

Display Properties

- specifies the type of rendering box used for an element
- default display property values are taken from behaviors described in the HTML specifications or from the browser/user default stylesheet

- https://developer.mozilla.org/en-US/docs/Web/CSS/display



Syntax

```
display: none;

display: inline;
display: block;
display: inline-block;
display: contents;
display: list-item;
display: inline-list-item;
display: table;
display: inline-table;
display: table-cell;
display: table-column;
display: table-column-group;
display: table-footer-group;
display: table-header-group;
display: table-row;
display: table-row-group;
display: table-caption;
display: flex;
display: inline-flex;
display: grid;
display: inline-grid;
```

Position Properties

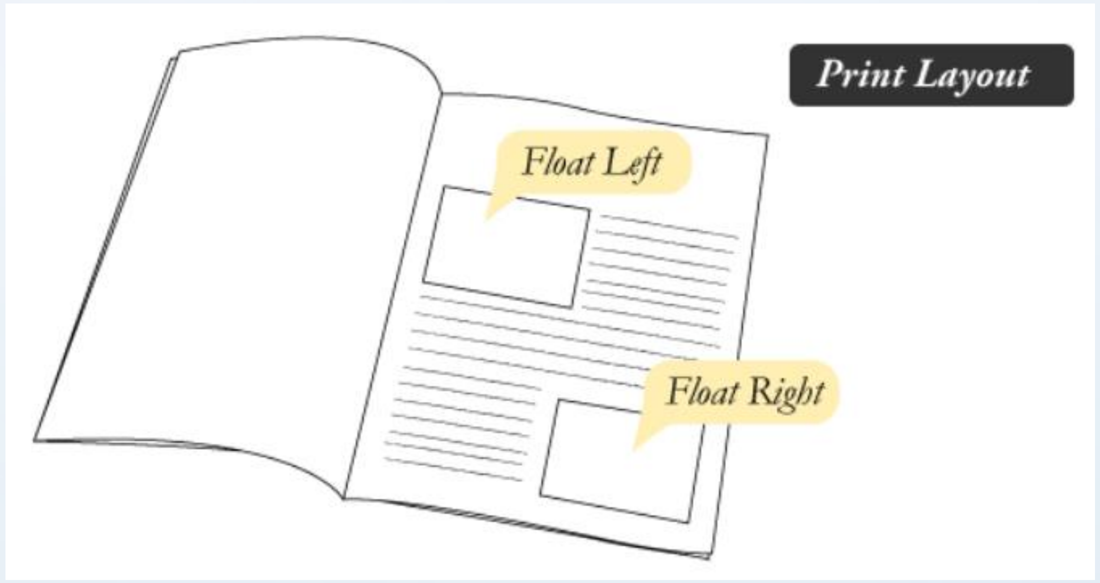
- position: relative
  - The element’s box is offset by some distance. The element retains the shape it would have had were it not positioned, and the space that the element would ordinarily have occupied is preserved
- position: absolute
  - The element’s box is completely removed from the flow of the document and positioned with respect to its containing block, which may be another element in the document or the initial containing block
- position: fixed
  - The element’s box behaves as though it was set to absolute, but its containing block is the viewport itself

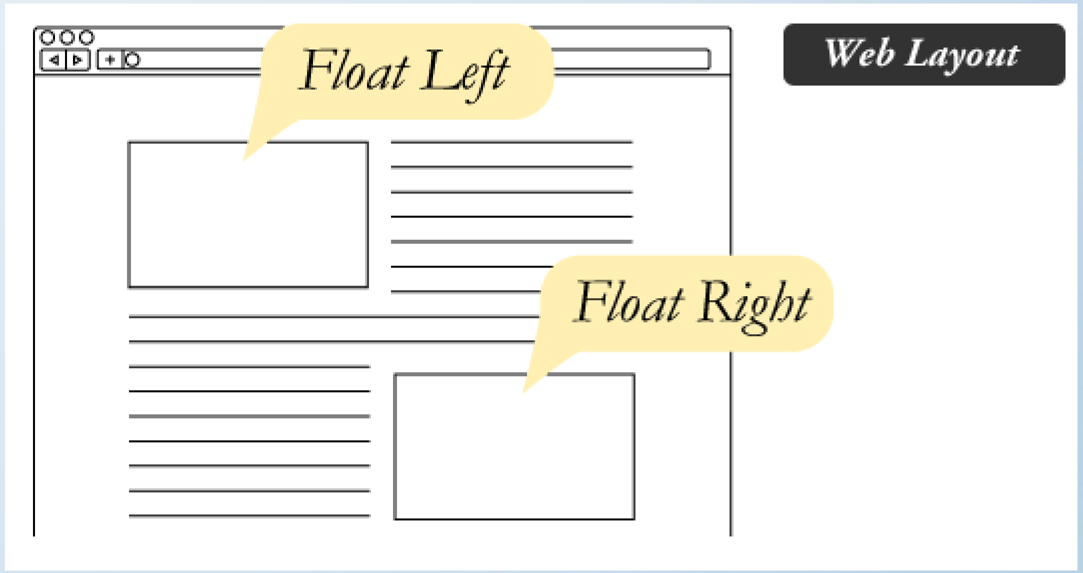
Reference : https://css-tricks.com/almanac/properties/p/position/

Float Properties

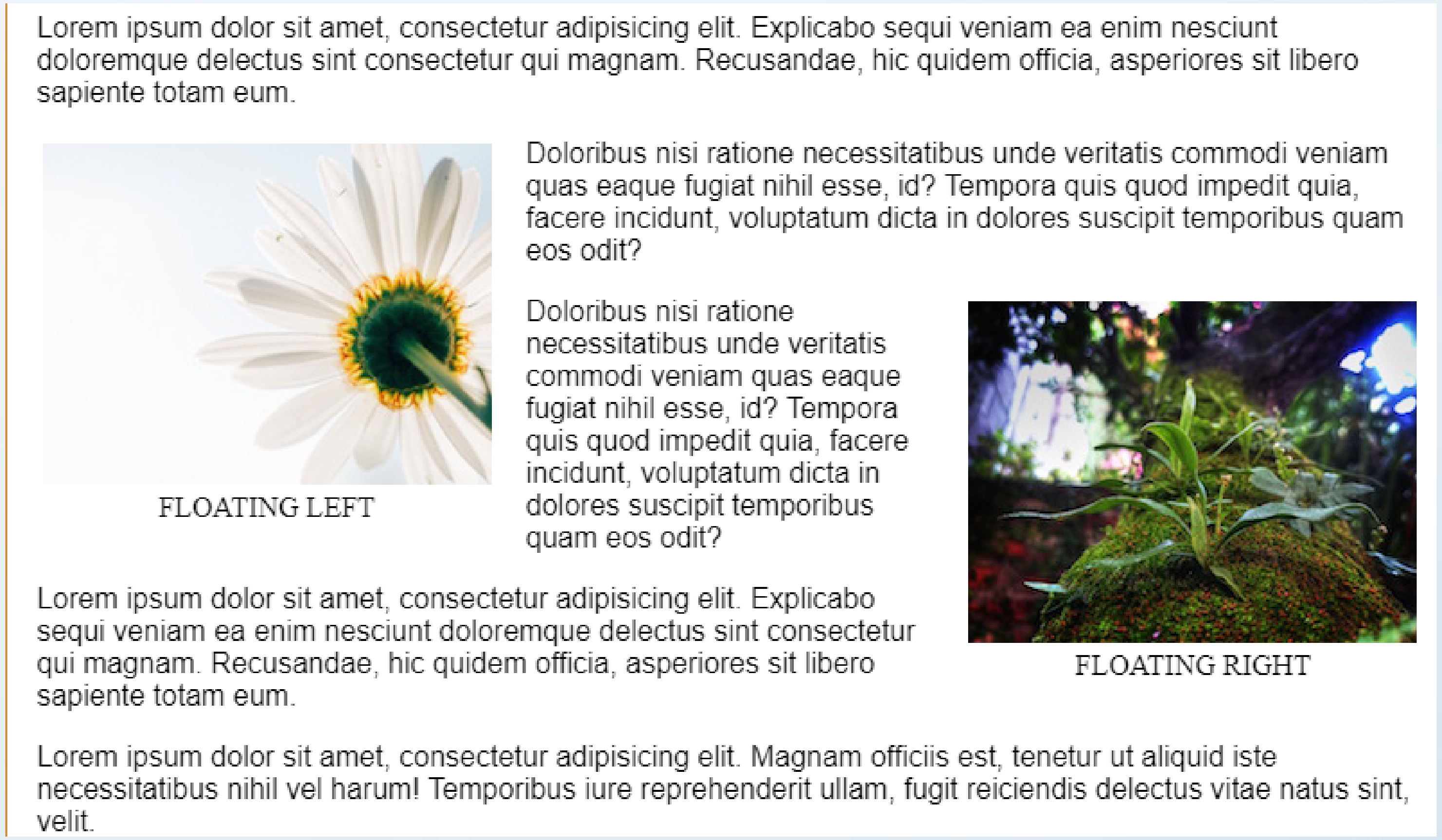
- CSS float property specifies how an element should float.
- CSS clear property specifies what elements can float beside the cleared element and on which side
- The float property can have one of the following values
  - left - The element floats to the left of its container
  - right - The element floats to the right of its container
  - none - The element does not float (will be displayed just where it occurs in the text). This is default
  - inherit - The element inherits the float value of its parent

Float Properties





Float Properties



Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

FLOATING LEFT

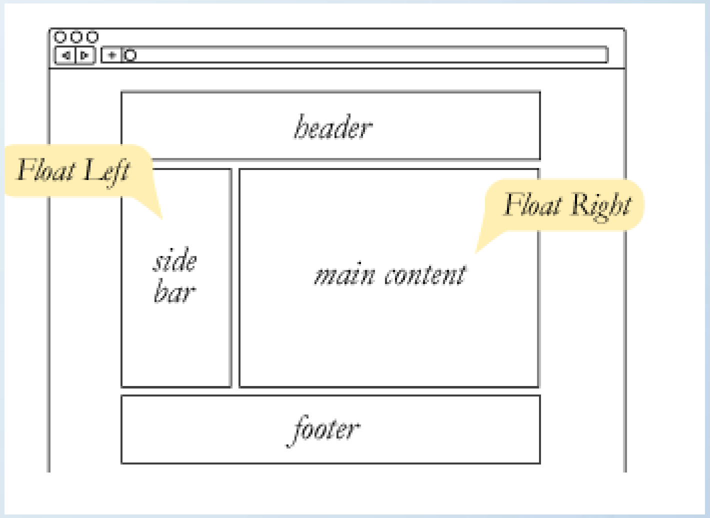
Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

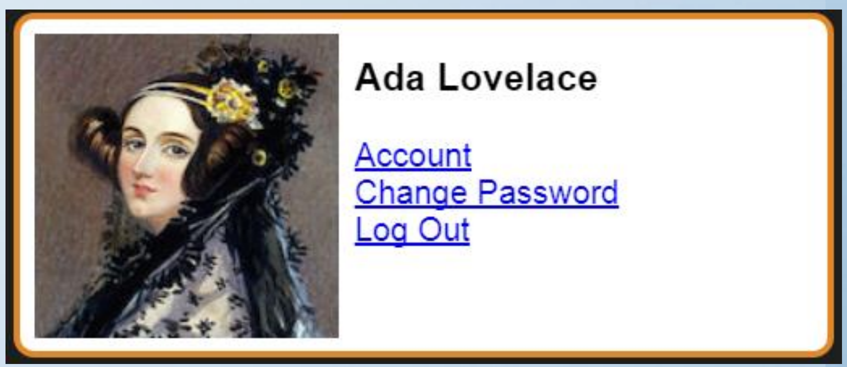
Doloribus nisi ratione necessitatibus unde veritatis commodi veniam quas eaque fugiat nihil esse, id? Tempora quis quod impedit quia, facere incidunt, voluptatum dicta in dolores suscipit temporibus quam eos odit?

FLOATING RIGHT

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Explicabo sequi veniam ea enim nesciunt doloremque delectus sint consectetur qui magnam. Recusandae, hic quidem officia, asperiores sit libero sapiente totam eum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Magnam officiis est, tenetur ut aliquid iste necessitatibus nihil vel harum! Temporibus iure reprehenderit ullam, fugit reiciendis delectus vitae natus sint, velit.



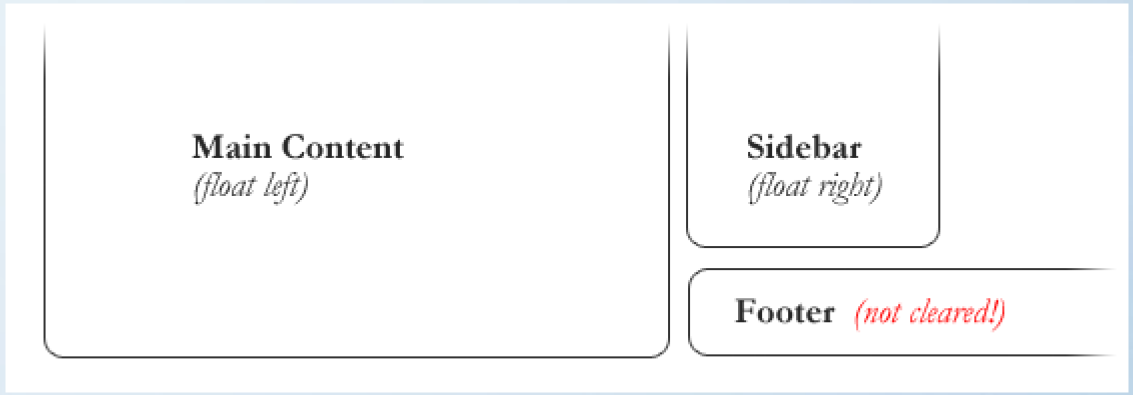


Ada Lovelace

Account
Change Password
Log Out

Clearing the Float

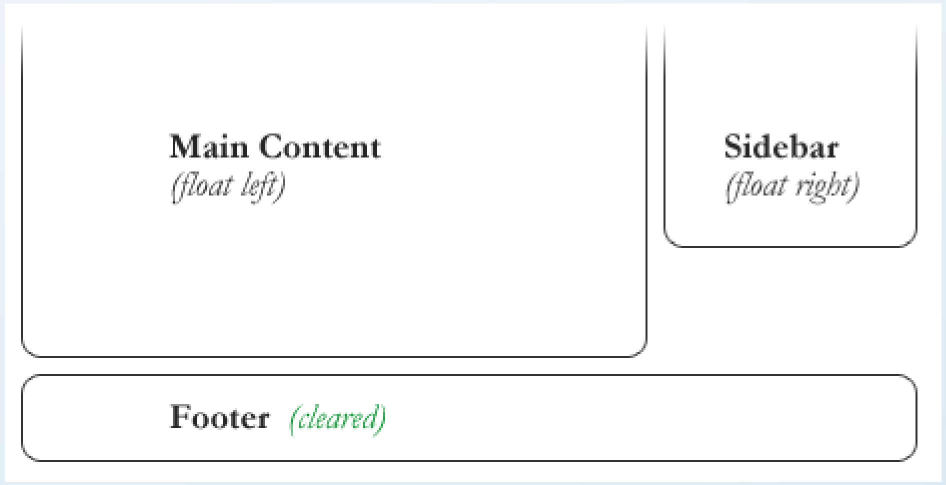
- Float's sister property is clear
- An element that has the clear property set on it will not move up adjacent to the float like the float desires, but will move itself down past the float

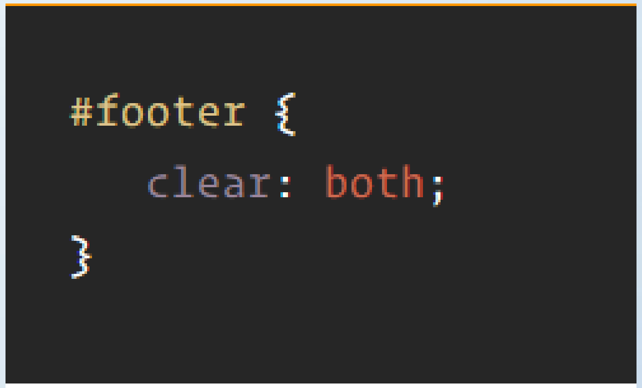


- the sidebar is floated to the right and is shorter than the main content area
- The footer then is required to jump up into that available space as is required by the float

Clearing the Float

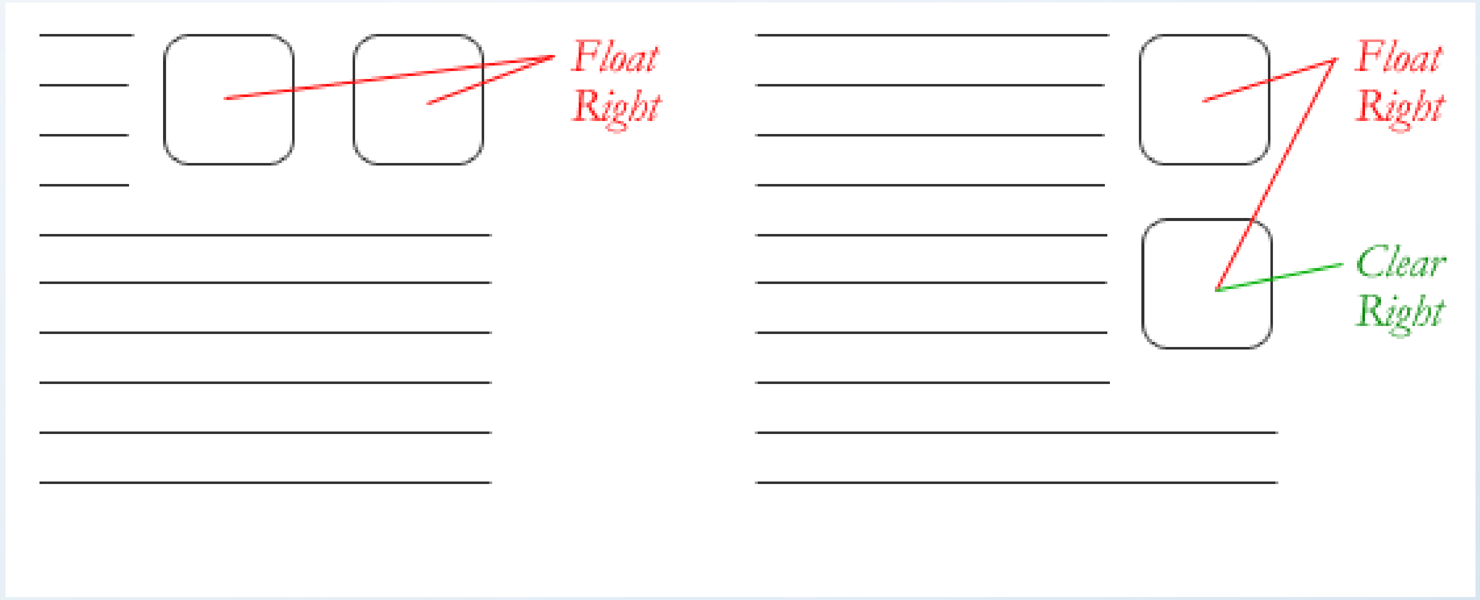
- To fix this problem, the footer can be cleared to ensure it stays beneath both floated columns





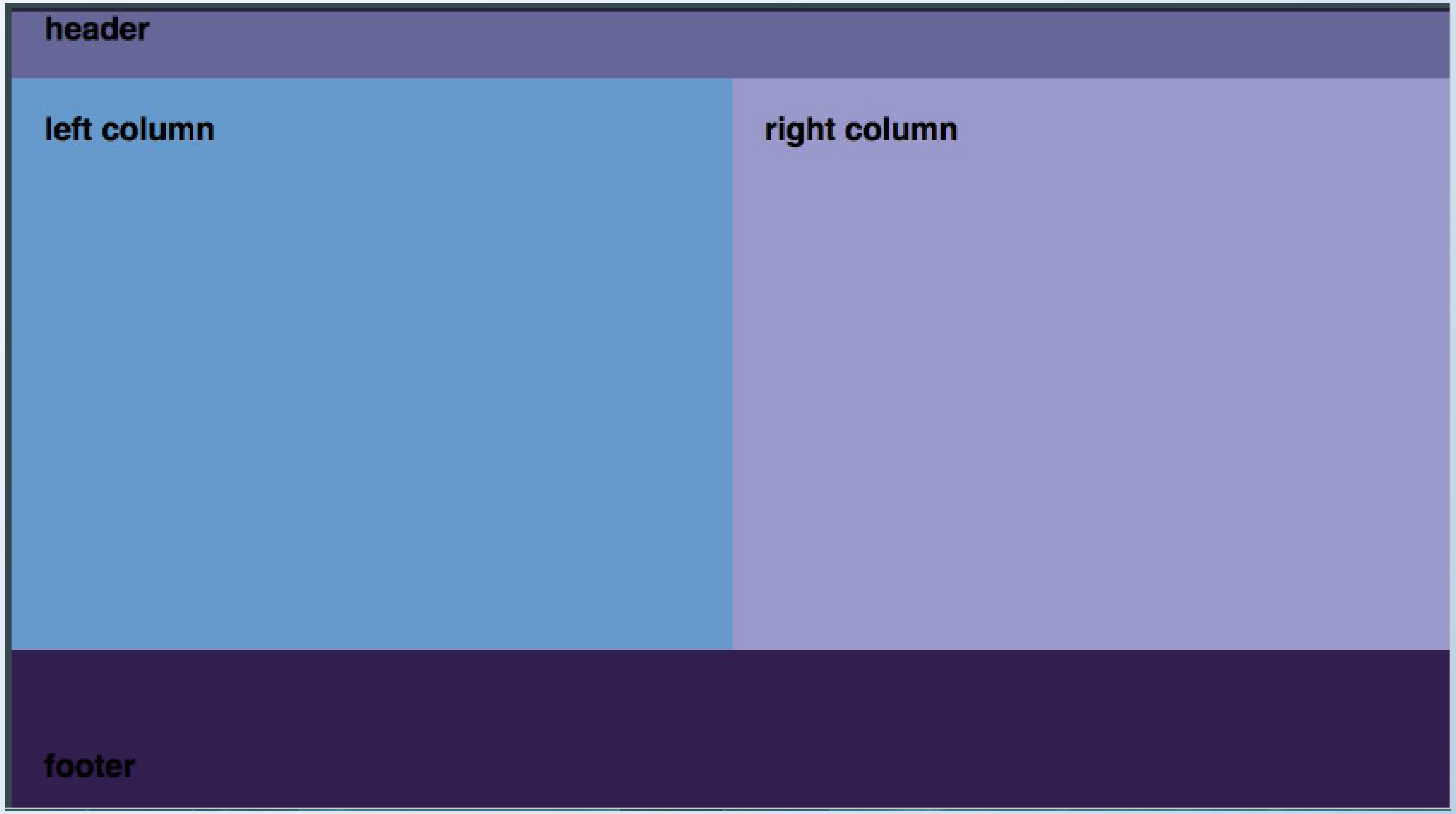
```css
#footer {
   clear: both;
}
```

Clearing the Float

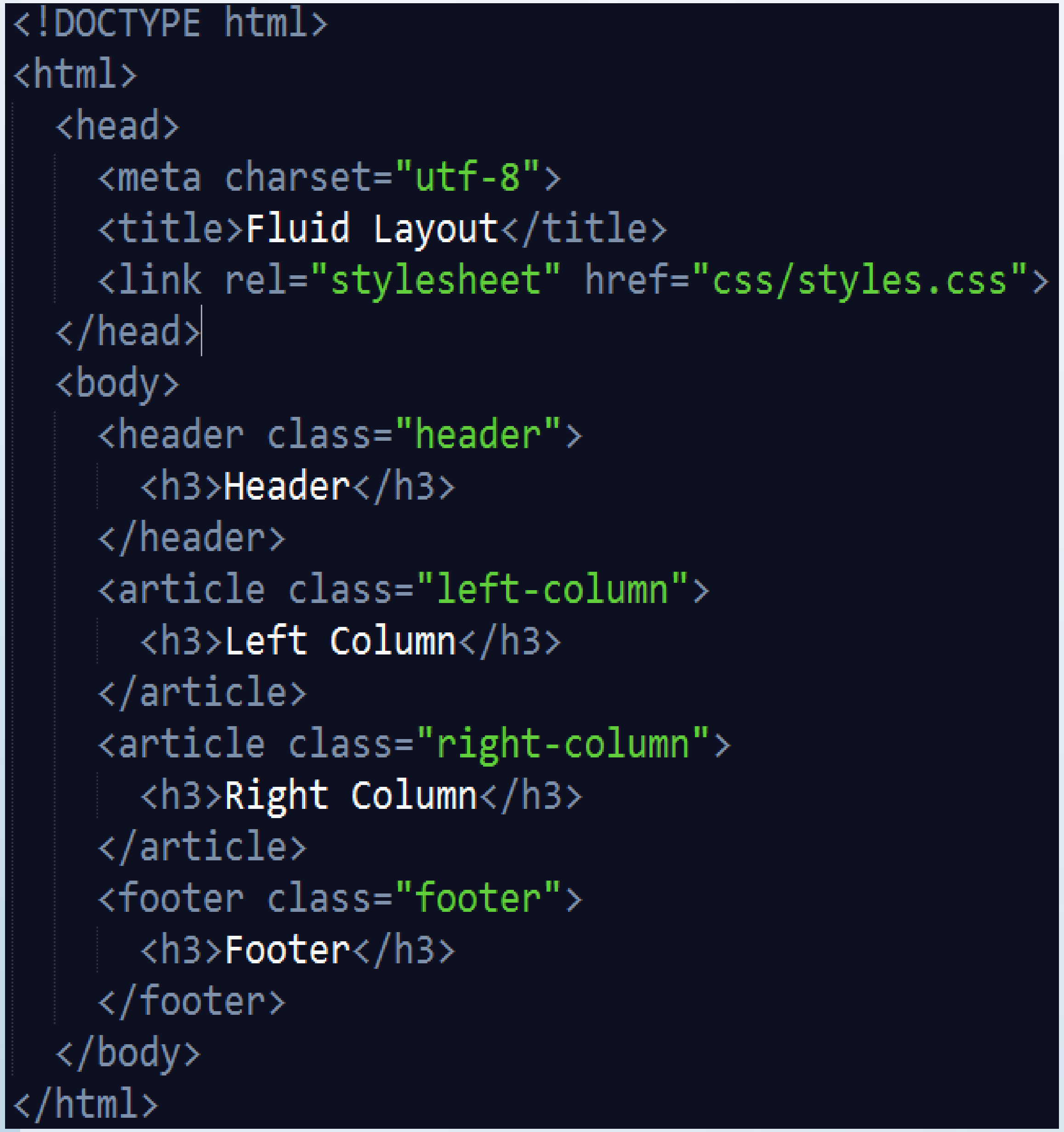


Simple layout - exercise 1

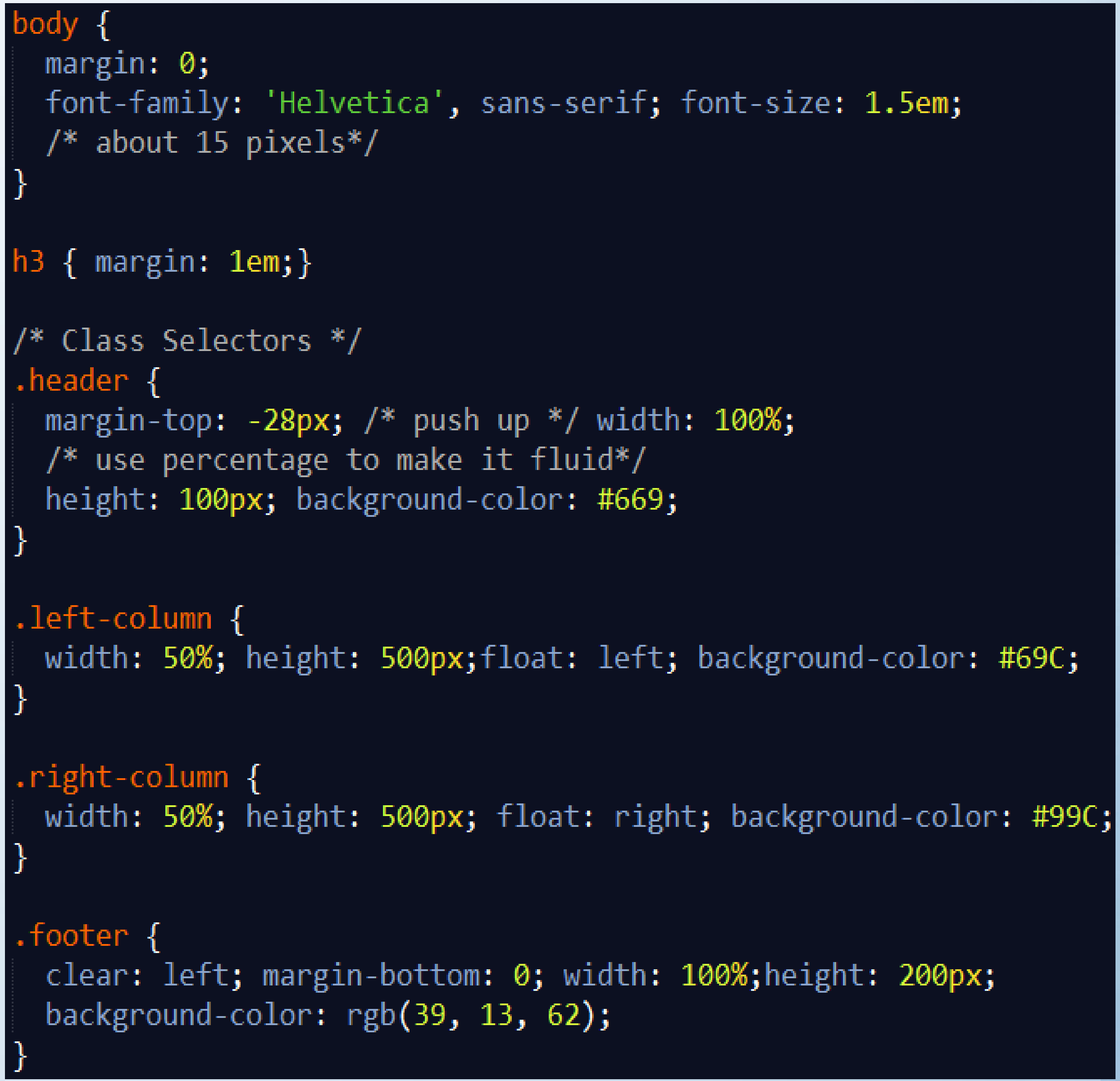
- Use float properties



Exercise 1 Solution



```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Fluid Layout</title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <header class="header">
      <h3>Header</h3>
    </header>
    <article class="left-column">
      <h3>Left Column</h3>
    </article>
    <article class="right-column">
      <h3>Right Column</h3>
    </article>
    <footer class="footer">
      <h3>Footer</h3>
    </footer>
  </body>
</html>
```

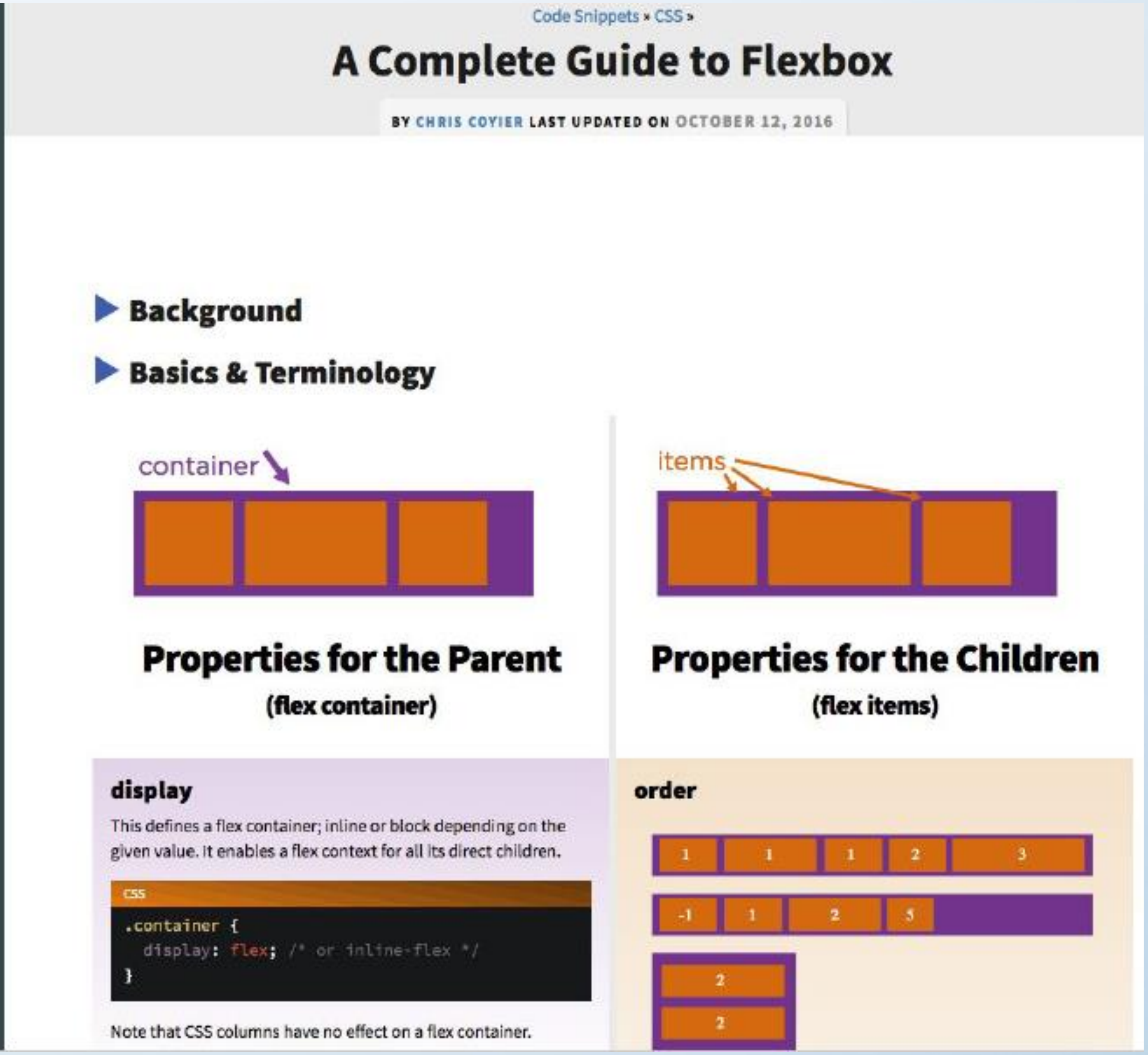


```css
body {
  margin: 0;
  font-family: 'Helvetica', sans-serif; font-size: 1.5em;
  /* about 15 pixels*/
}

h3 { margin: 1em;}

/* Class Selectors */
.header {
  margin-top: -28px; /* push up */ width: 100%;
  /* use percentage to make it fluid*/
  height: 100px; background-color: #669;
}

.left-column {
  width: 50%; height: 500px;float: left; background-color: #69C;
}

.right-column {
  width: 50%; height: 500px; float: right; background-color: #99C;
}

.footer {
  clear: left; margin-bottom: 0; width: 100%;height: 200px;
  background-color: rgb(39, 13, 62);
}
```

Flexbox Properties

Complete Guide to **Flexbox**: CSS Tricks (Chris Coyier)

https://css-tricks.com/snippets/css/a-guide-to-flexbox/



Code Snippets » CSS »

A Complete Guide to Flexbox

BY CHRIS COYIER LAST UPDATED ON OCTOBER 12, 2016

▶ **Background**

▶ **Basics & Terminology**

Properties for the Parent
(flex container)

Properties for the Children
(flex items)

display

This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

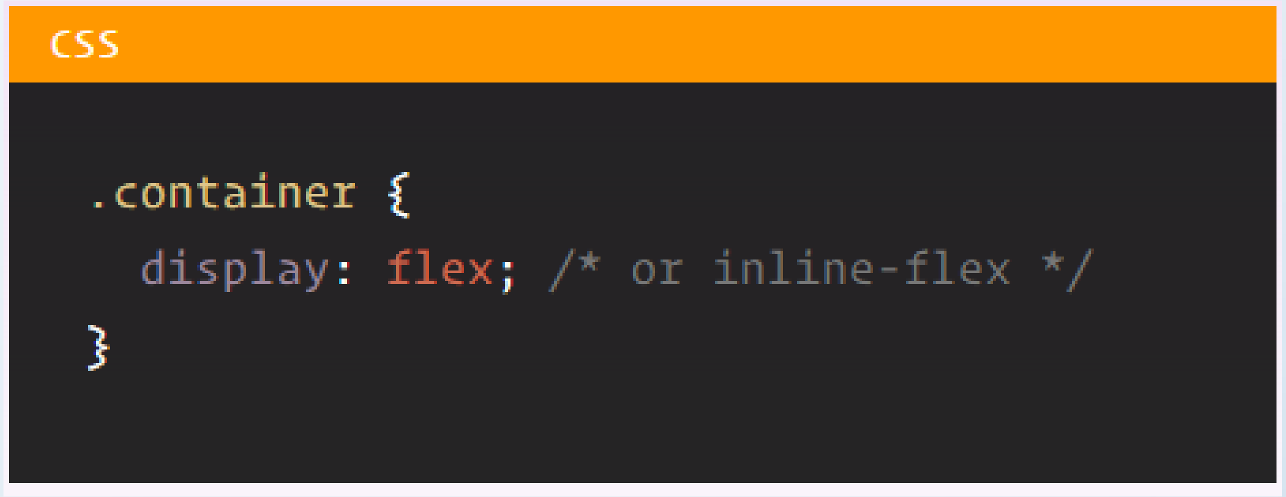
```css
.container {
  display: flex; /* or inline-flex */
}
```

Note that CSS columns have no effect on a flex container.

order

Flexbox Properties

- Parent Box CSS (Flex Container)
  - display : flex



CSS

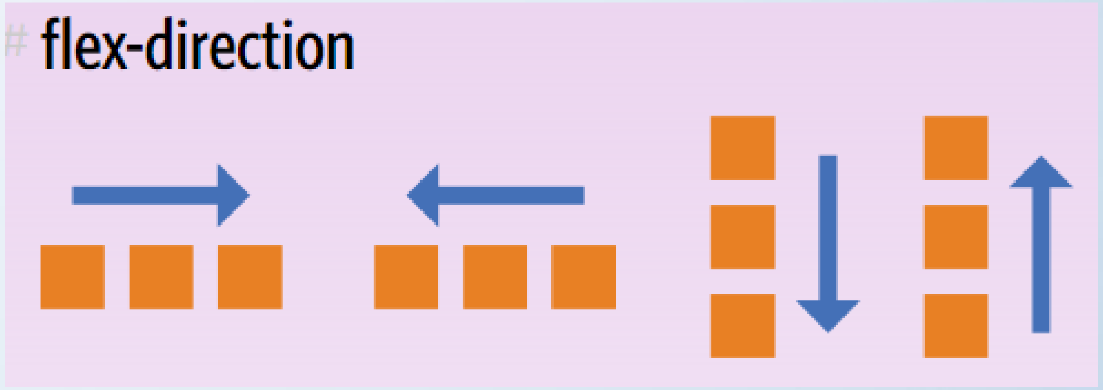
```css
.container {
  display: flex; /* or inline-flex */
}
```

Flexbox Properties

- Parent Box CSS (Flex Container)

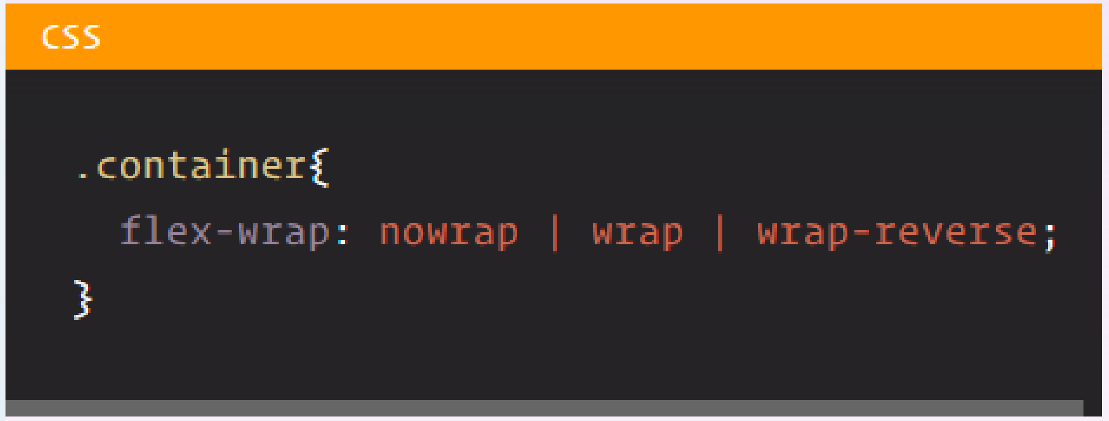
flex-direction: row | row-reverse |

column | column-reverse;



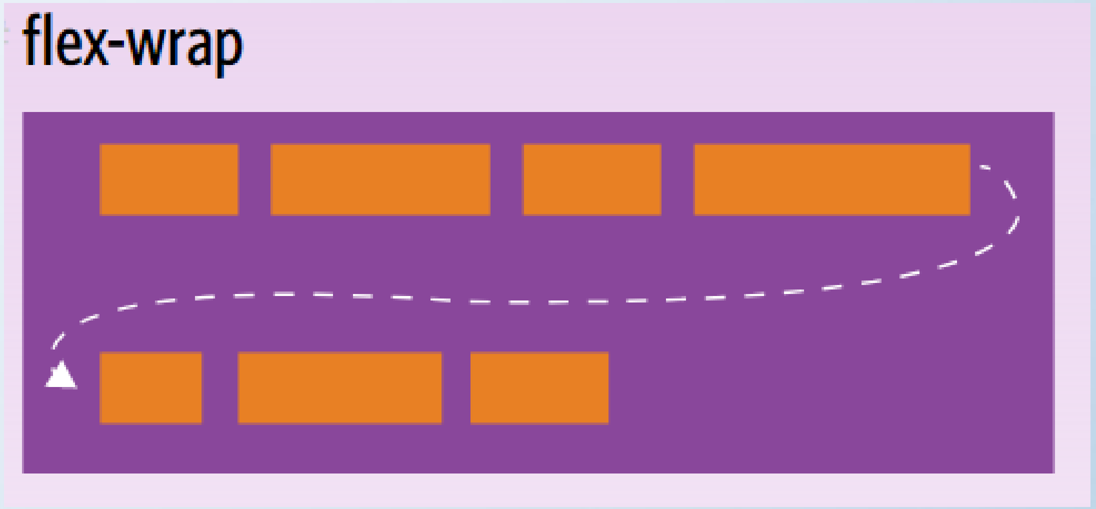
Flexbox Properties

- Parent Box CSS (Flex Container)



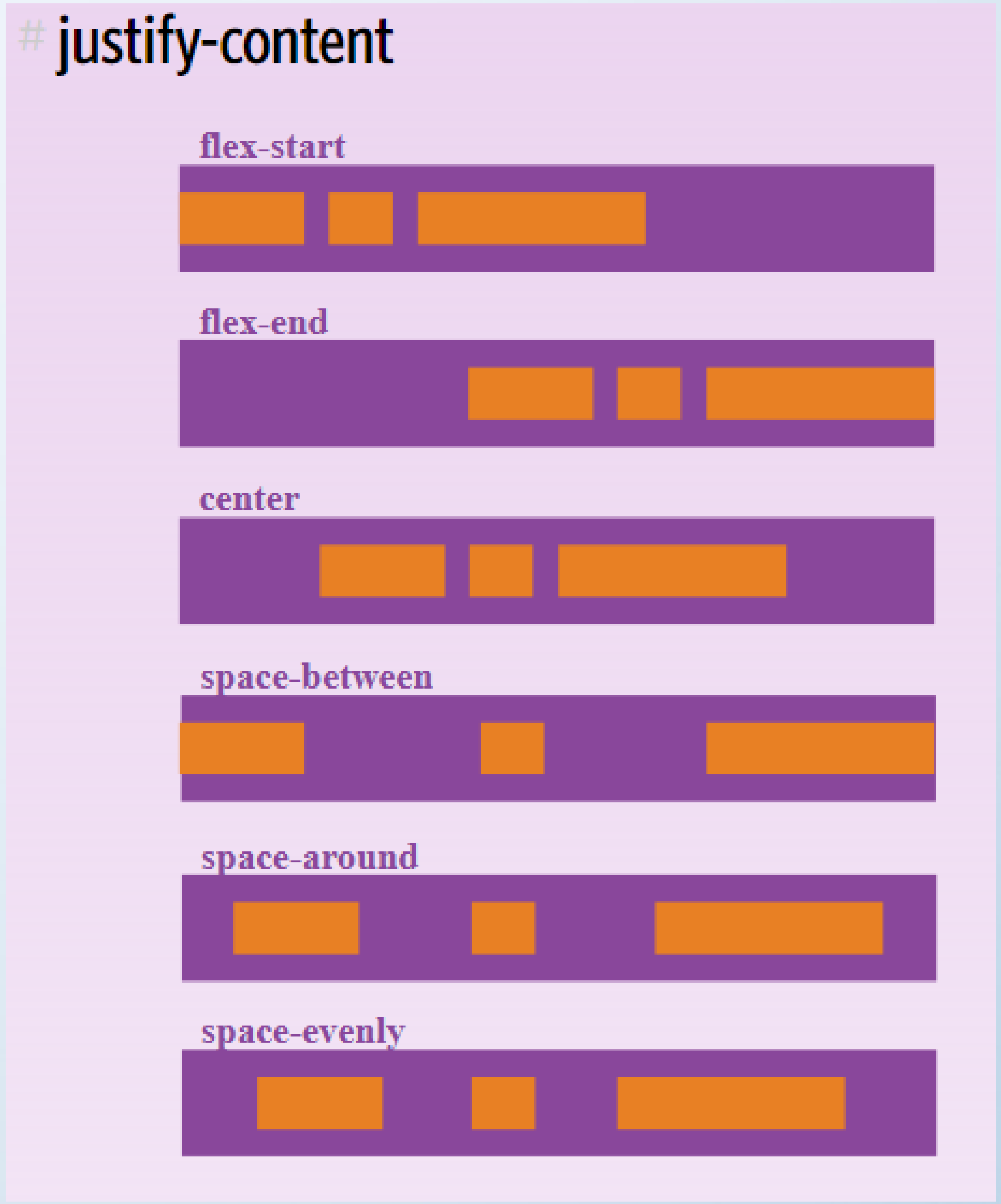
CSS

```css
.container{
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```



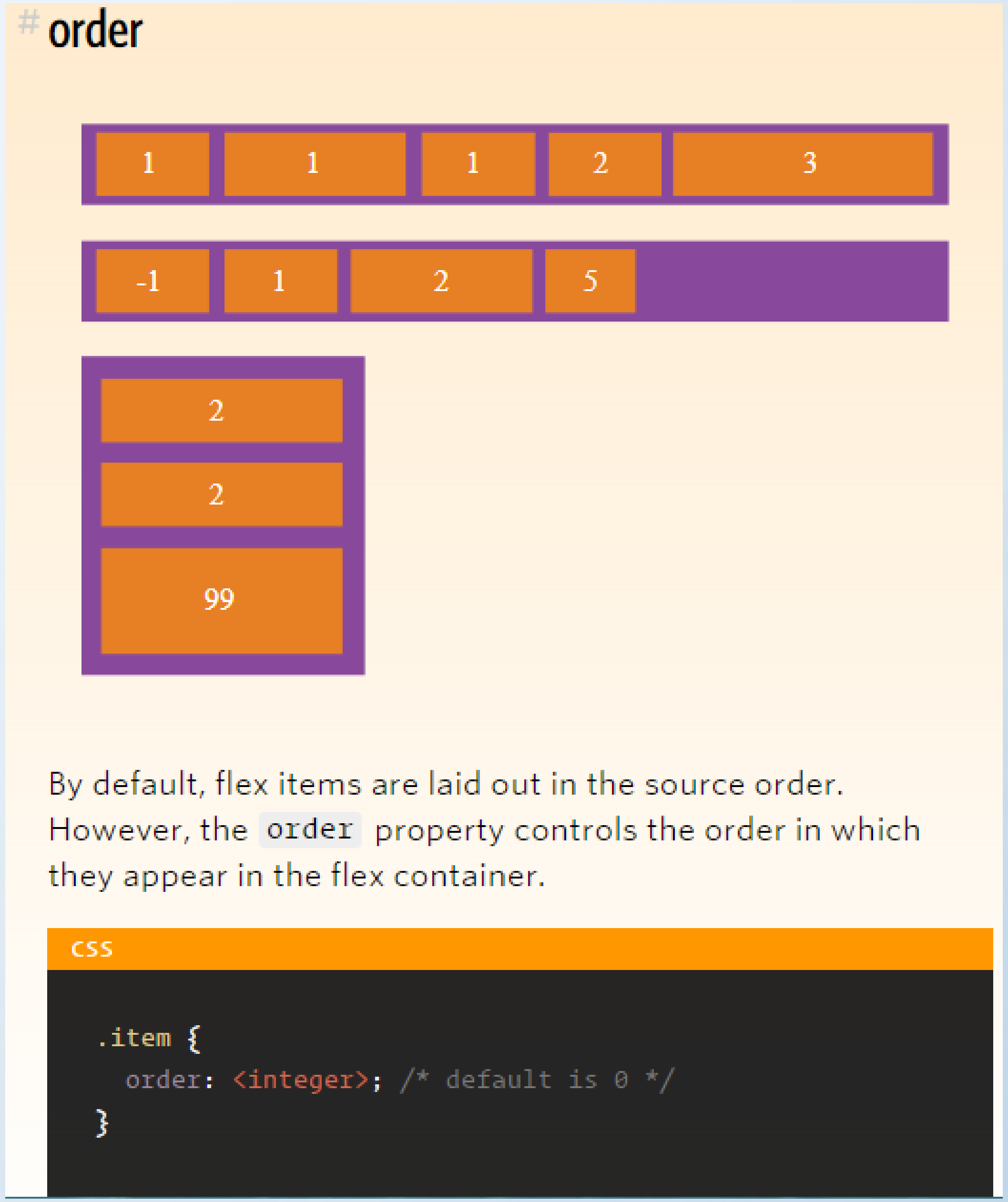
Flexbox Properties

Parent Box CSS (Flex Container)



Flexbox Properties

Children Box CSS (Flex Items)



order

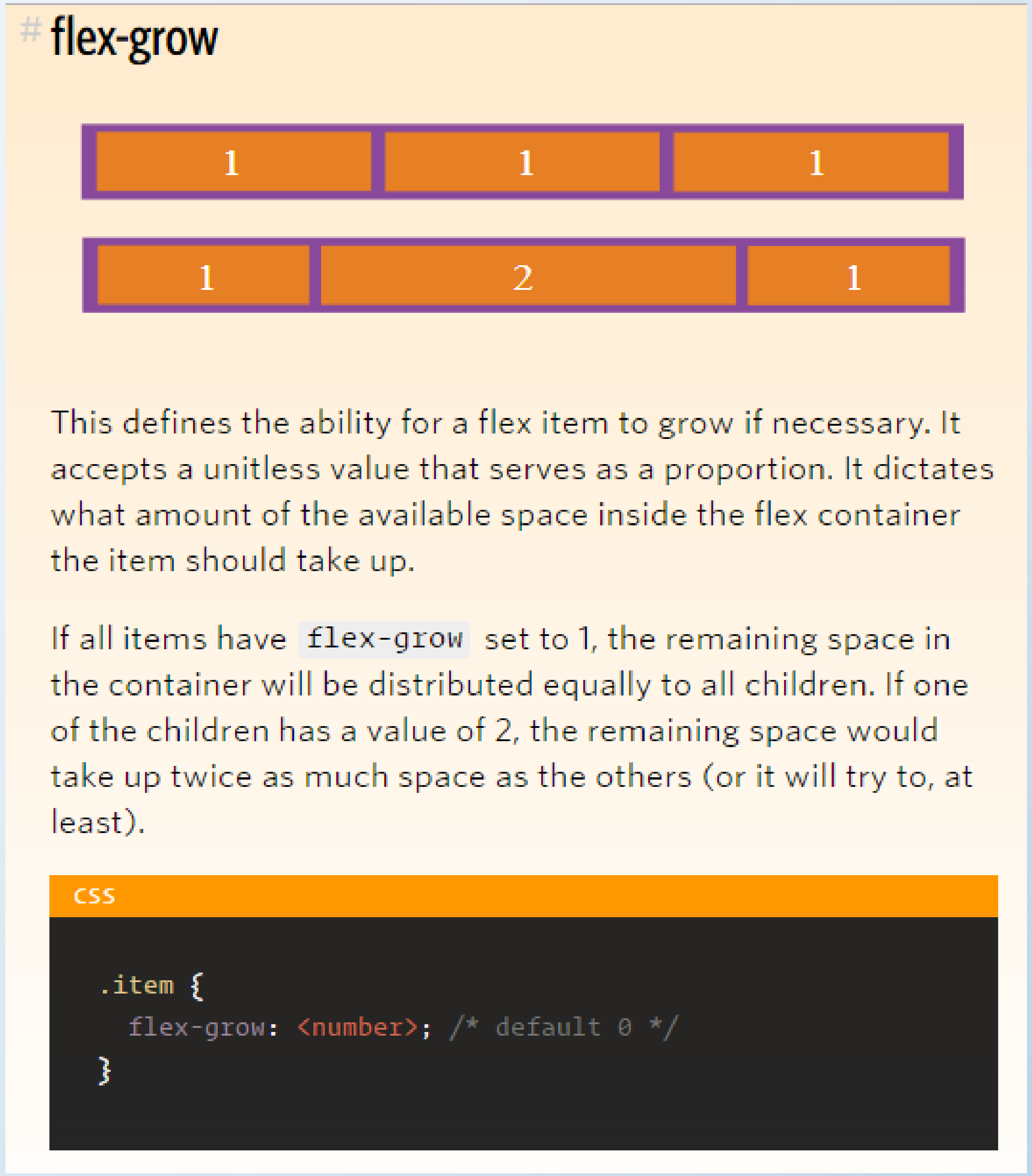
By default, flex items are laid out in the source order. However, the `order` property controls the order in which they appear in the flex container.

css

```css
.item {
  order: <integer>; /* default is 0 */
}
```

Flexbox Properties

Children Box CSS (Flex Items)



flex-grow

This defines the ability for a flex item to grow if necessary. It accepts a unitless value that serves as a proportion. It dictates what amount of the available space inside the flex container the item should take up.

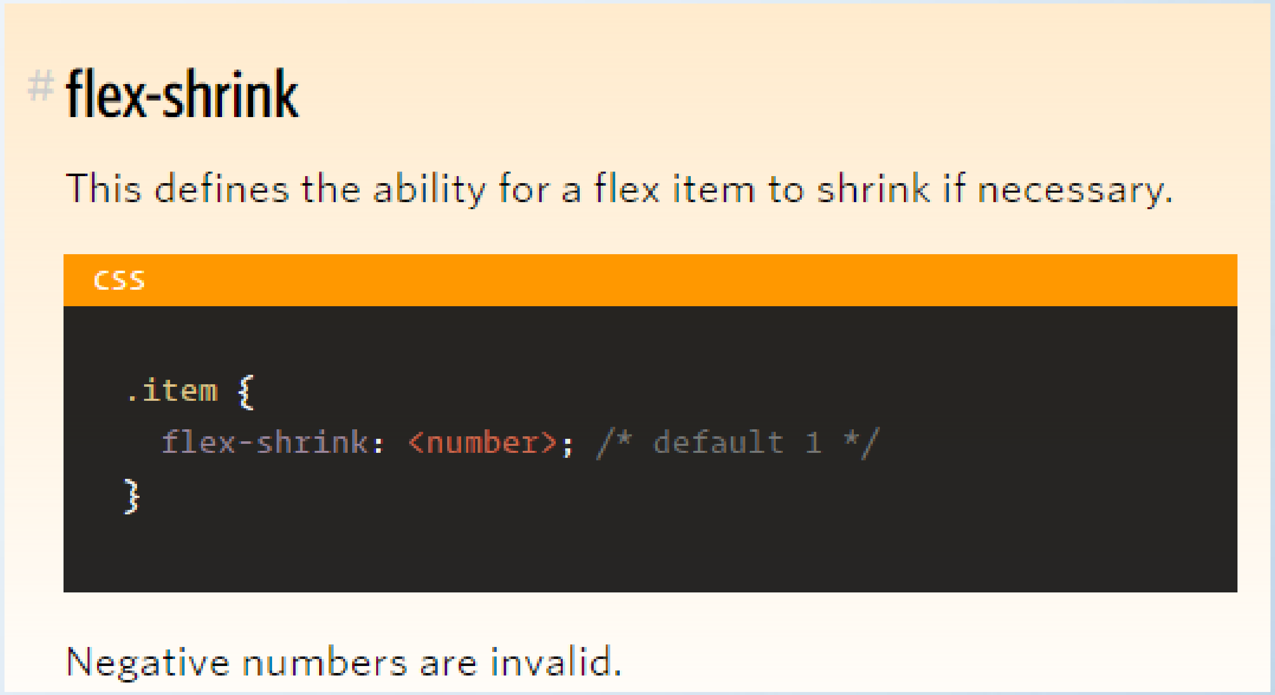
If all items have `flex-grow` set to 1, the remaining space in the container will be distributed equally to all children. If one of the children has a value of 2, the remaining space would take up twice as much space as the others (or it will try to, at least).

CSS

```css
.item {
  flex-grow: <number>; /* default 0 */
}
```

Flexbox Properties

Children Box CSS (Flex Items)



flex-shrink

This defines the ability for a flex item to shrink if necessary.

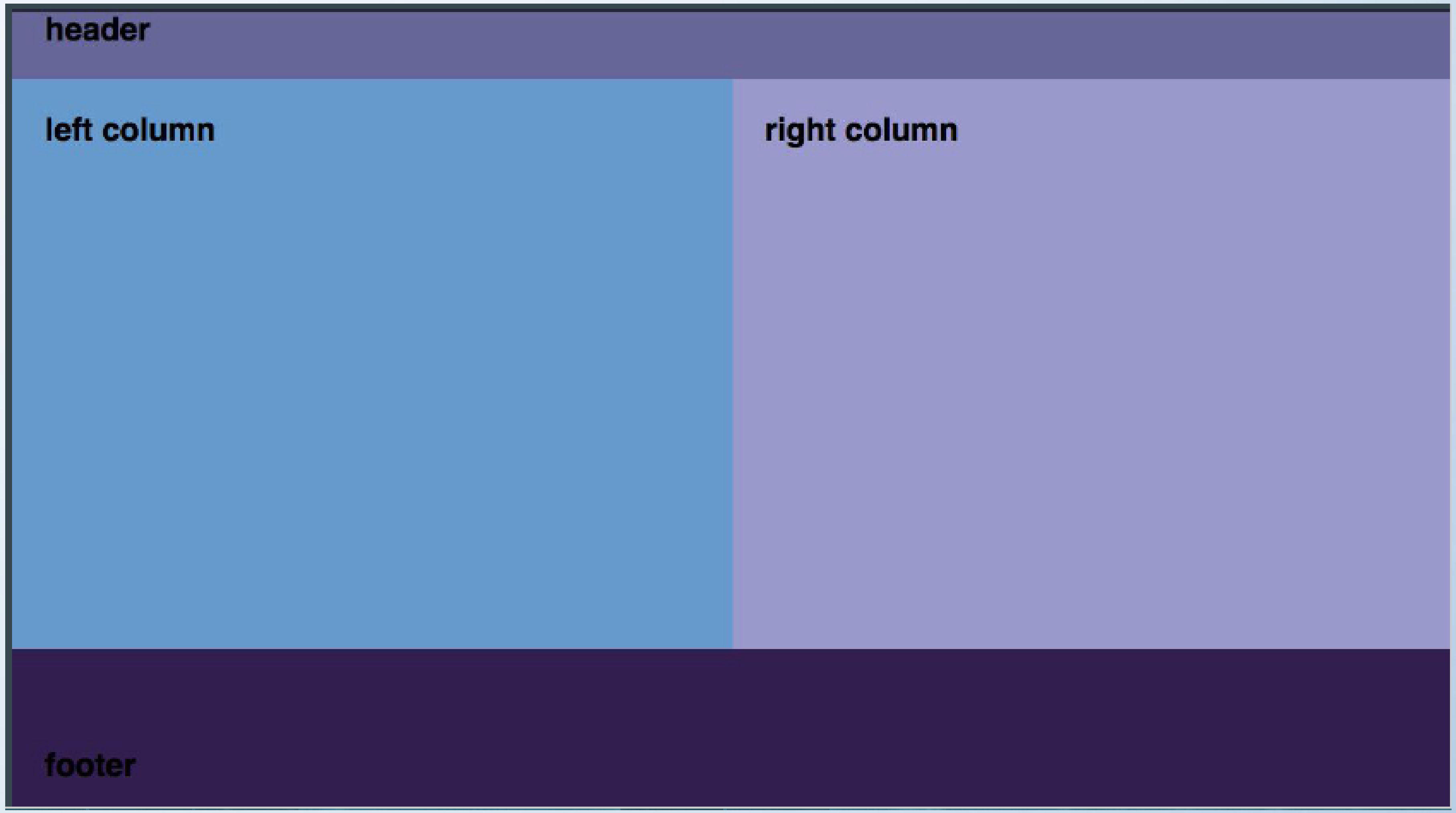
css

```css
.item {
  flex-shrink: <number>; /* default 1 */
}
```

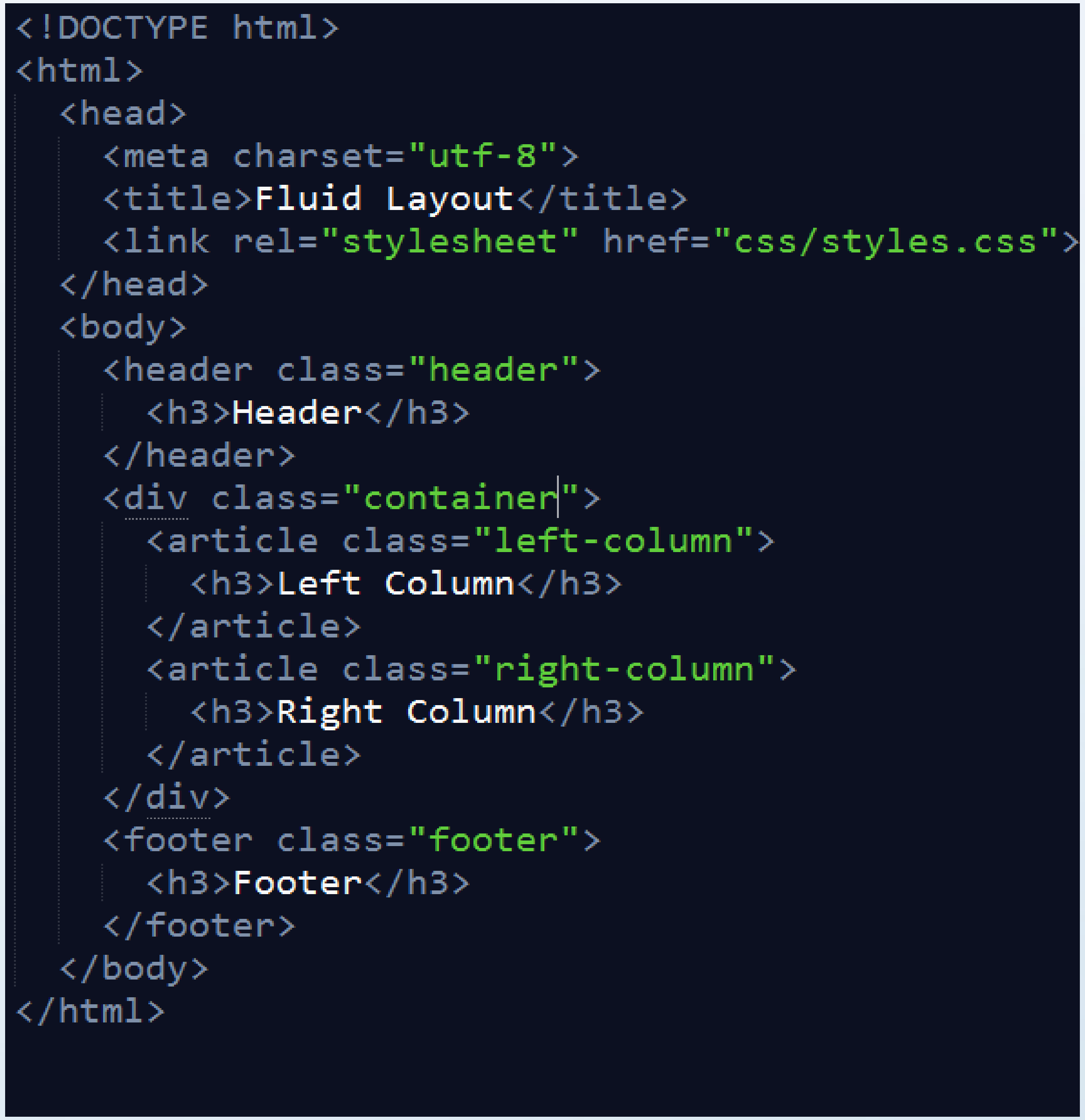
Negative numbers are invalid.

Simple layout - exercise 2

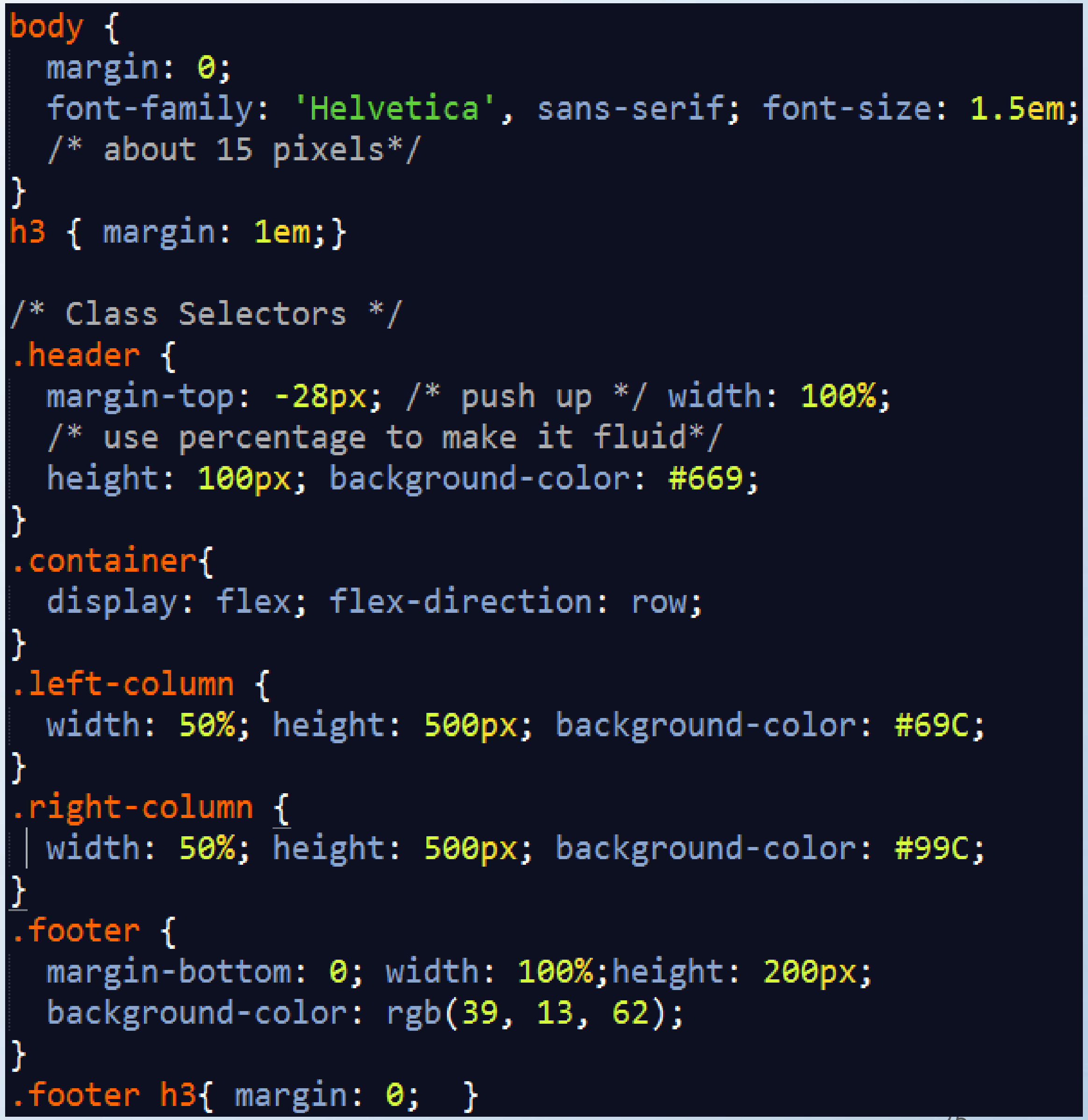
- Use flex properties



Exercise 2 Solution



```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Fluid Layout</title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <header class="header">
      <h3>Header</h3>
    </header>
    <div class="container">
      <article class="left-column">
        <h3>Left Column</h3>
      </article>
      <article class="right-column">
        <h3>Right Column</h3>
      </article>
    </div>
    <footer class="footer">
      <h3>Footer</h3>
    </footer>
  </body>
</html>
```



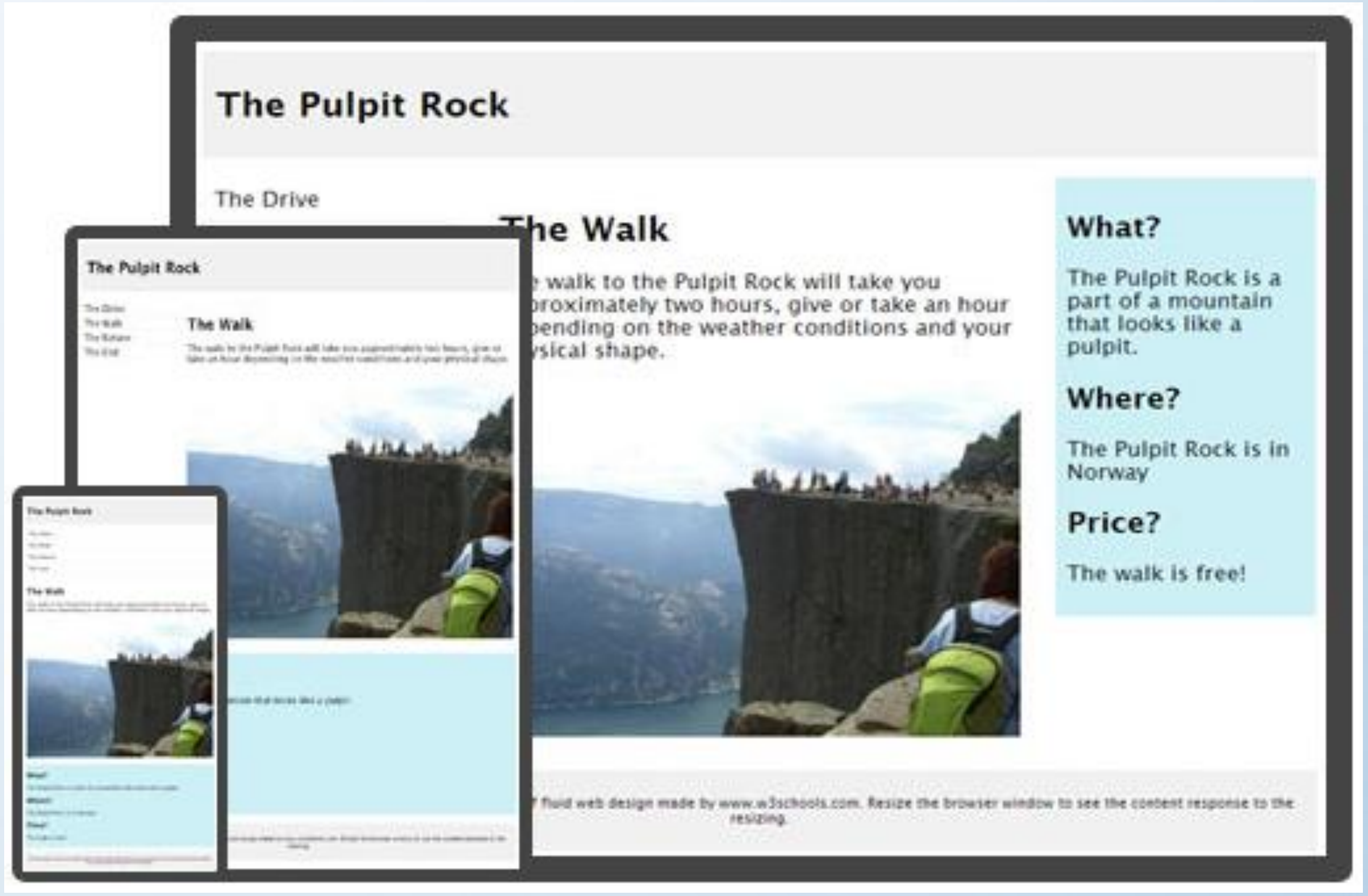
```css
body {
  margin: 0;
  font-family: 'Helvetica', sans-serif; font-size: 1.5em;
  /* about 15 pixels*/
}
h3 { margin: 1em;}

/* Class Selectors */
.header {
  margin-top: -28px; /* push up */ width: 100%;
  /* use percentage to make it fluid*/
  height: 100px; background-color: #669;
}
.container{
  display: flex; flex-direction: row;
}
.left-column {
  width: 50%; height: 500px; background-color: #69C;
}
.right-column {
  width: 50%; height: 500px; background-color: #99C;
}
.footer {
  margin-bottom: 0; width: 100%;height: 200px;
  background-color: rgb(39, 13, 62);
}
.footer h3{ margin: 0;  }
```

Responsive Web Design

- Responsive vs Adaptive Designs

Responsive Web Design

- Responsive Web Design makes your web page look good on all devices (desktops, tablets, and phones)



The Pulpit Rock

The Drive

he Walk

e walk to the Pulpit Rock will take you
proximately two hours, give or take an hour
pending on the weather conditions and your
ysical shape.

What?

The Pulpit Rock is a part of a mountain that looks like a pulpit.

Where?

The Pulpit Rock is in Norway

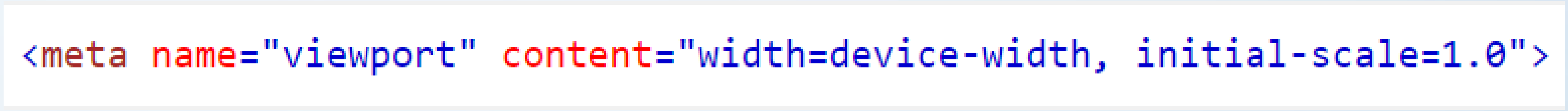
Price?

The walk is free!

f fluid web design made by www.w3schools.com. Resize the browser window to see the content response to the resizing.

Media Queries for Responsive Design

- Include this in head section

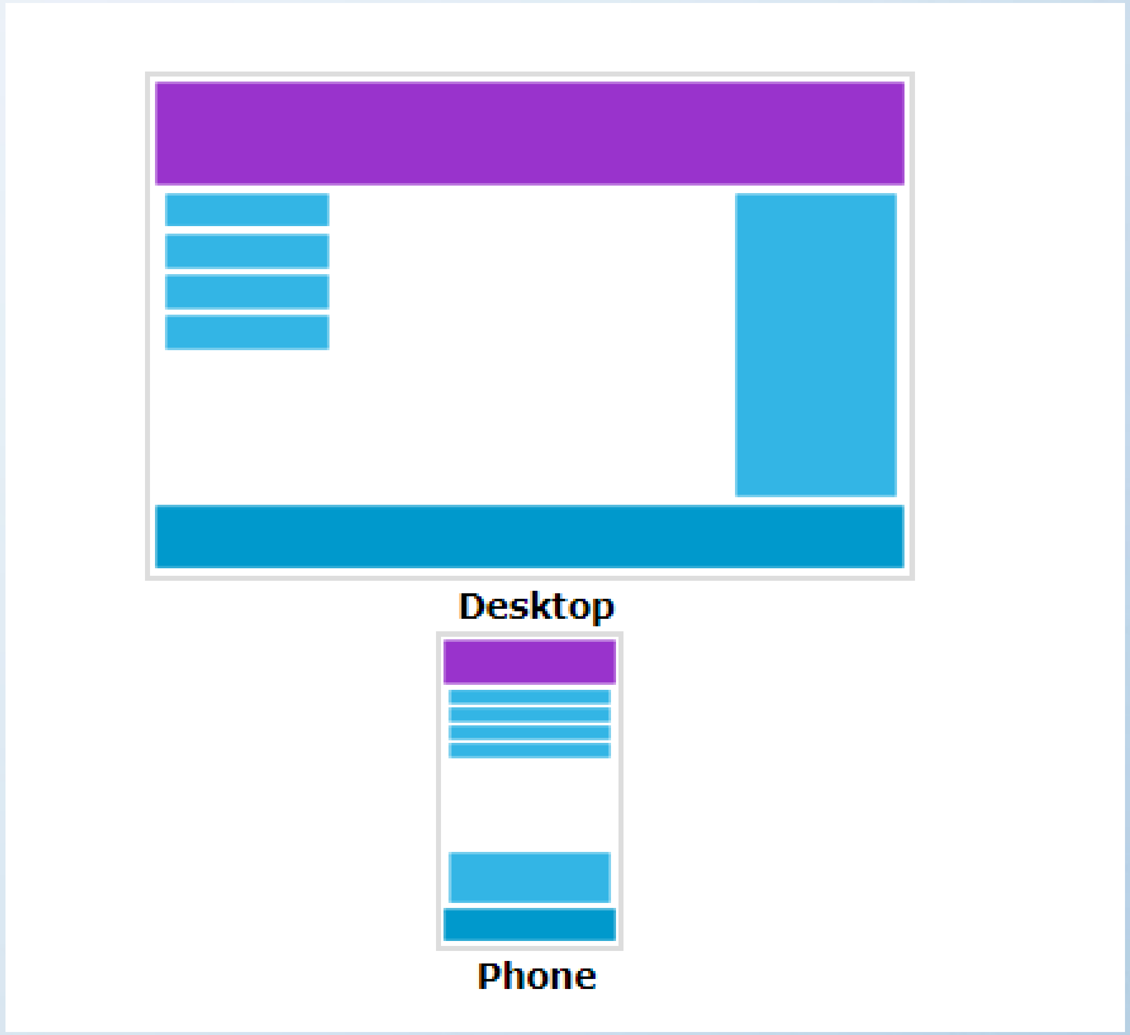


```html
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- With media queries, you can define completely different styles for different browser sizes

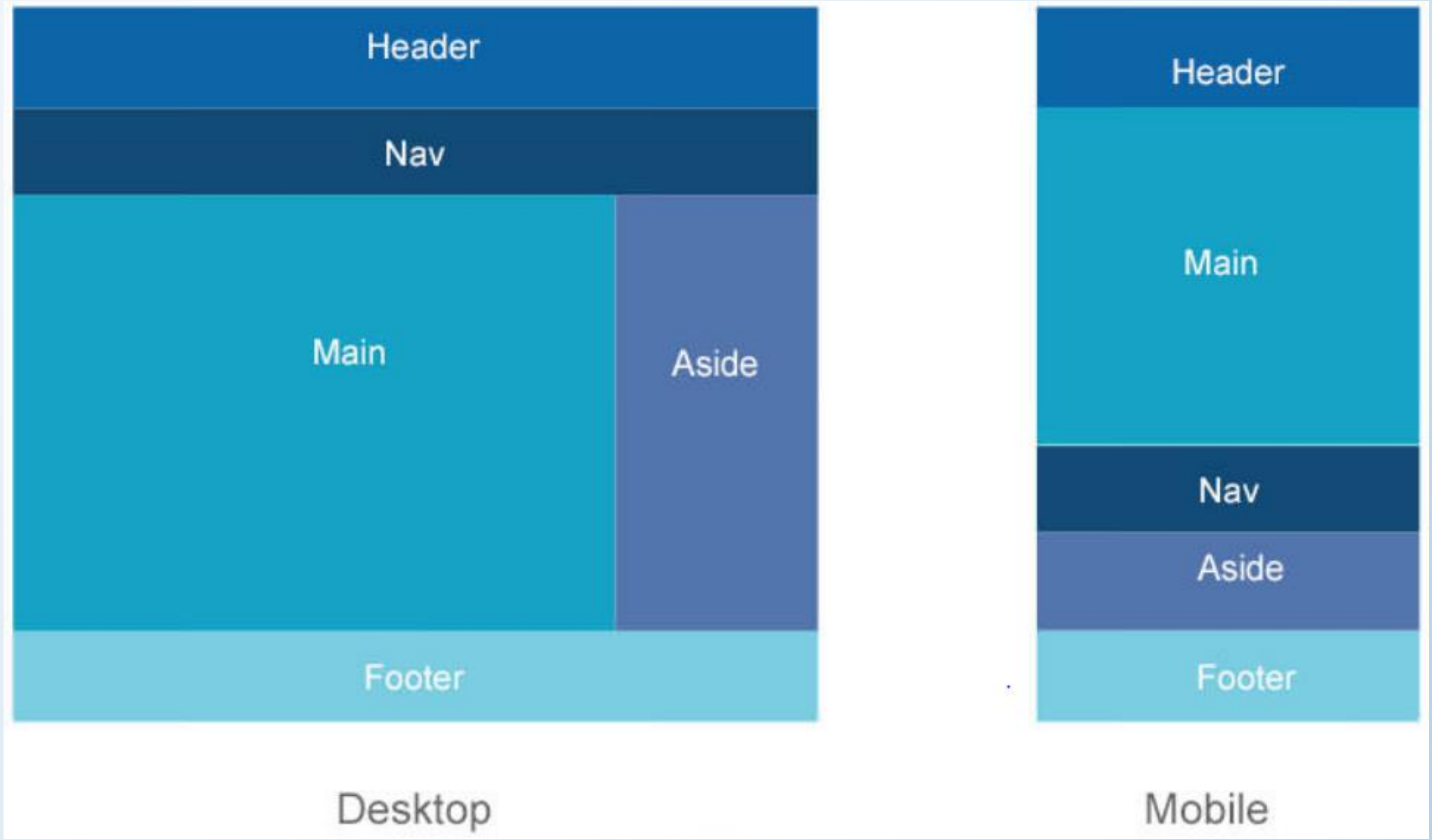
Media Queries for Responsive Design

```
@media screen and (max-width : 500px){
        tag/id/class{
                // CSS code here
        }
  }


@media screen and (min-width : 500px) and
(max-width: 800px)
{
        tag/id/class{
                // CSS code here
        }
  }
```

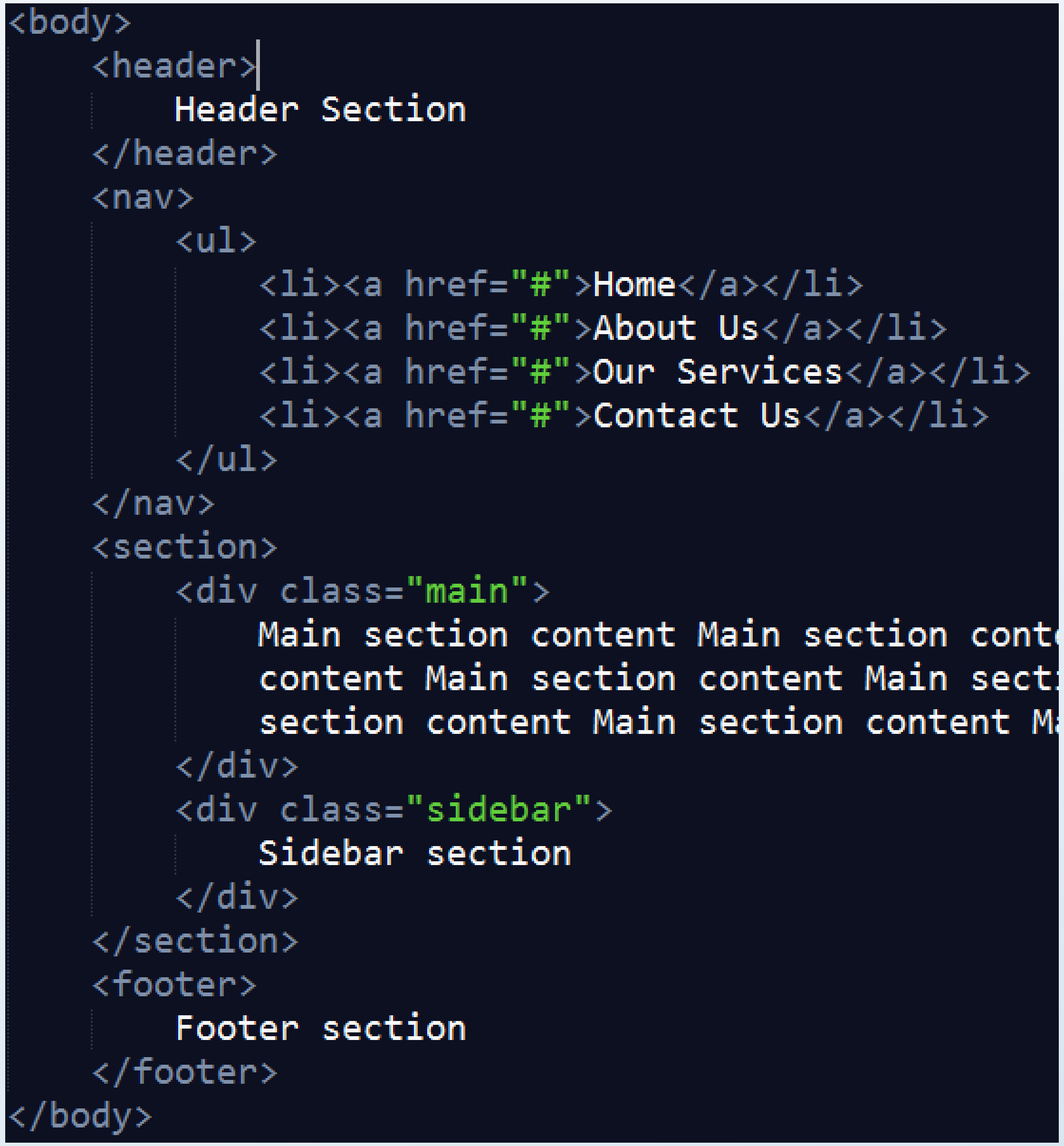


Exercise 3

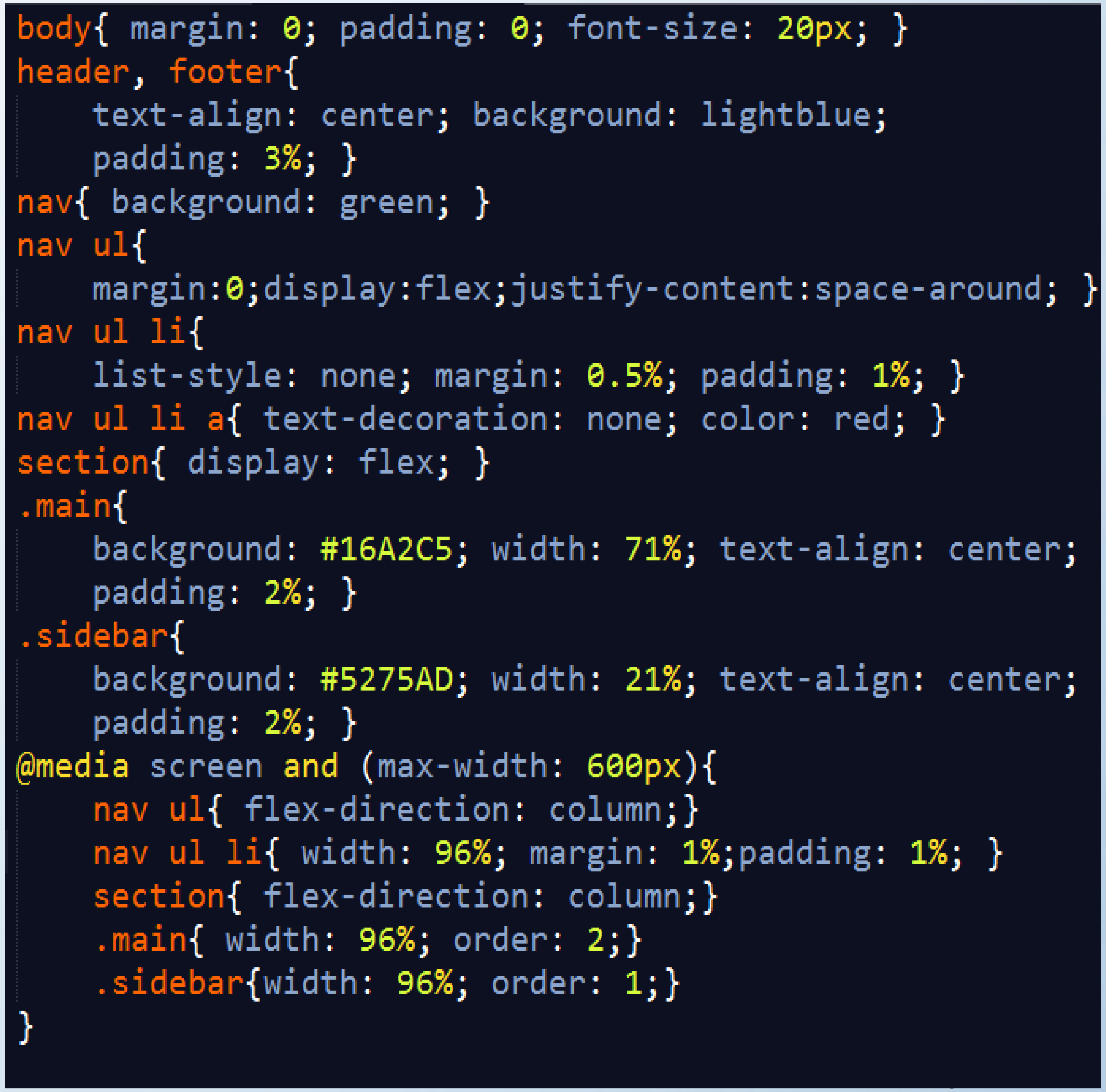
Make this webpage responsive using media queries



Exercise 3 Solution



```html
<body>
    <header>
        Header Section
    </header>
    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="#">Our Services</a></li>
            <li><a href="#">Contact Us</a></li>
        </ul>
    </nav>
    <section>
        <div class="main">
            Main section content Main section conte
            content Main section content Main sect
            section content Main section content M
        </div>
        <div class="sidebar">
            Sidebar section
        </div>
    </section>
    <footer>
        Footer section
    </footer>
</body>
```



```css
body{ margin: 0; padding: 0; font-size: 20px; }
header, footer{
    text-align: center; background: lightblue;
    padding: 3%; }
nav{ background: green; }
nav ul{
    margin:0;display:flex;justify-content:space-around; }
nav ul li{
    list-style: none; margin: 0.5%; padding: 1%; }
nav ul li a{ text-decoration: none; color: red; }
section{ display: flex; }
.main{
    background: #16A2C5; width: 71%; text-align: center;
    padding: 2%; }
.sidebar{
    background: #5275AD; width: 21%; text-align: center;
    padding: 2%; }
@media screen and (max-width: 600px){
    nav ul{ flex-direction: column;}
    nav ul li{ width: 96%; margin: 1%;padding: 1%; }
    section{ flex-direction: column;}
    .main{ width: 96%; order: 2;}
    .sidebar{width: 96%; order: 1;}
}
```

More Examples

- https://www.w3schools.com/css/css3_mediaqueries_ex.asp
- https://www.w3schools.com/css/tryit.asp?filename=trycss_mediaqueries_website