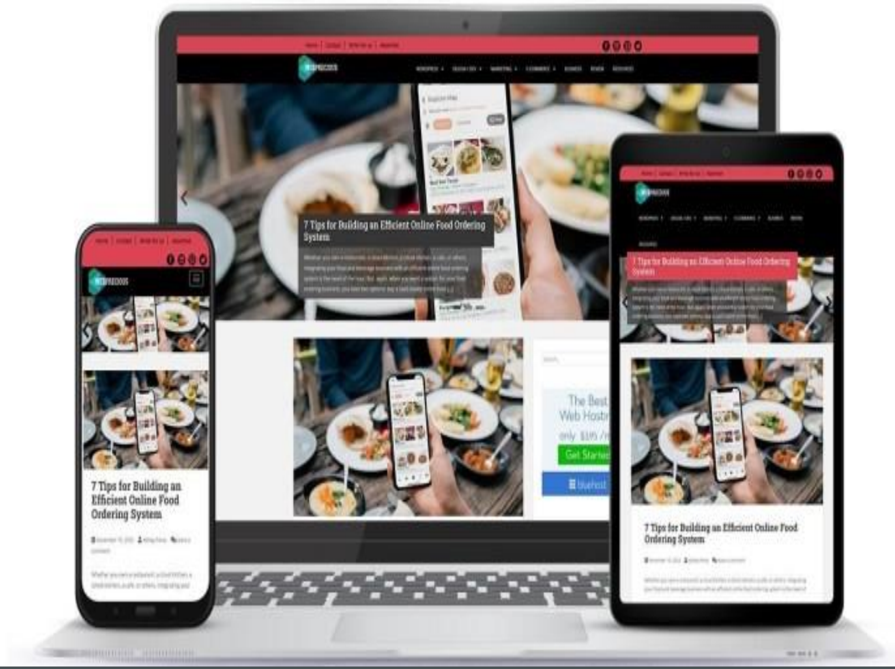
RESPONSIVE WEB DEISGN

Dr. Daniela Tsaneva


7 Tips for Building an Efficient Online Food Ordering System

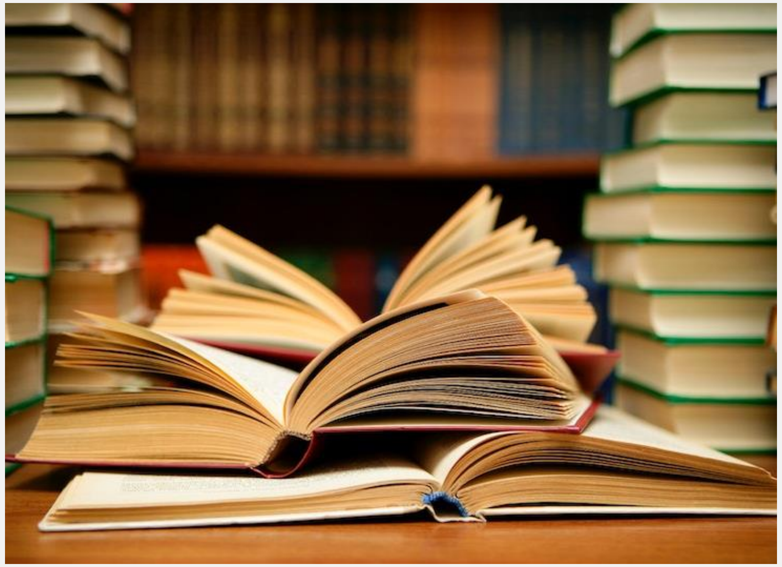
OUTLINE

- Responsive Web Design
- Code Examples
- Flexbox and Grid
- Code Examples
- Exercise 3

RESPONSIVE WEB DESIGN

Now we can annotate our content and documents so a web browser can understand their structure and can visually style that content and lay it out on a page, we will start to think about how our users access this content and what that means for our webpage design. We'll look at some of the simpler tools at our disposal to help us build pages that are accessible no matter what type of device our users are using.

READING LIST



Required Reading

- Responsive Web Design, https://alistapart.com/article/responsive-web-design/- Ethan Marcotte - although this is over ten years old it still holds up as a really great introduction to the whys and how of Responsive Web Design

Related Reading

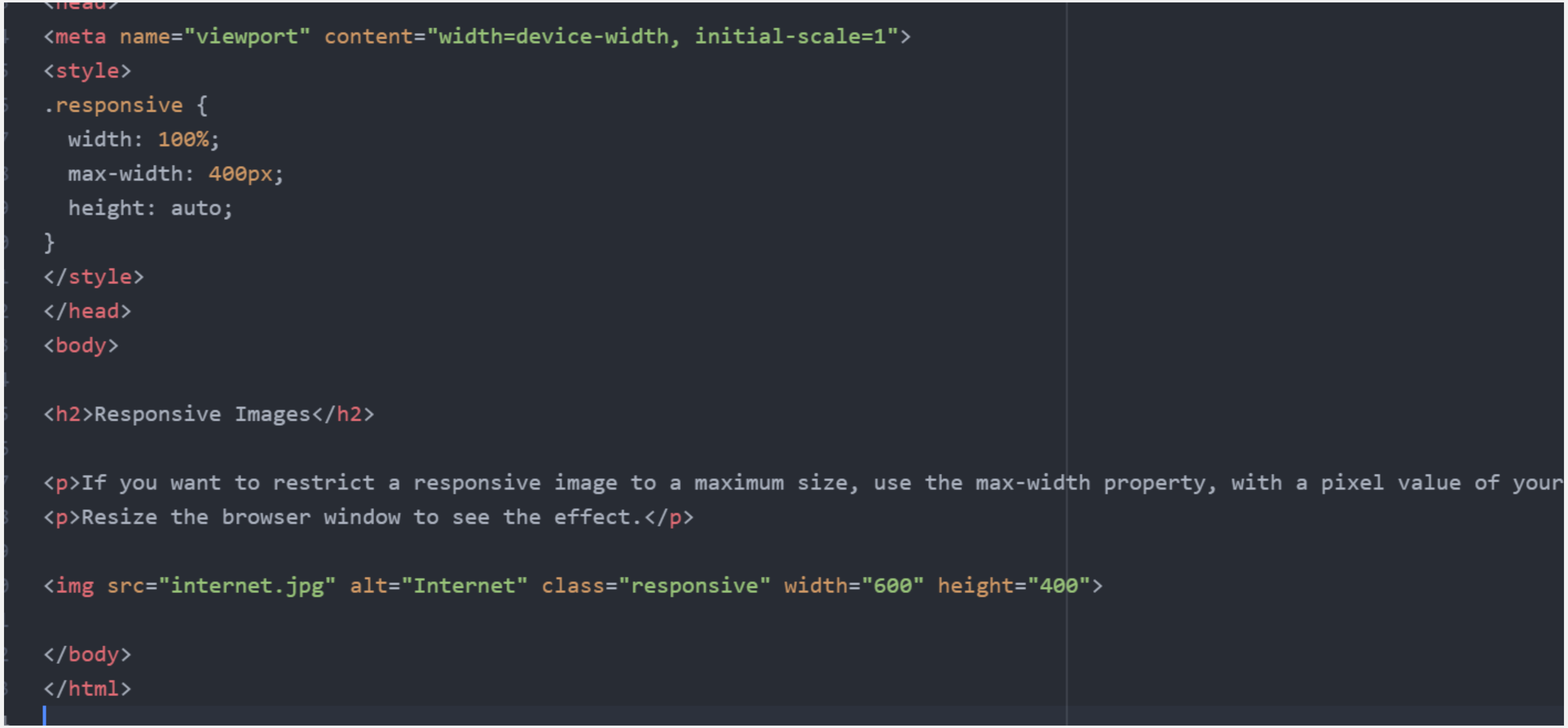
- Mobile First Design: Why It's Great and Why It Sucks, https://designshack.net/articles/css/mobilefirst/
- MDN: Media Queries, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_media_queries

CODE EXAMPLES

There are a set of code examples that accompany this course showing the use of the fundamental HTML tags and CSS styles/rules that we look at.

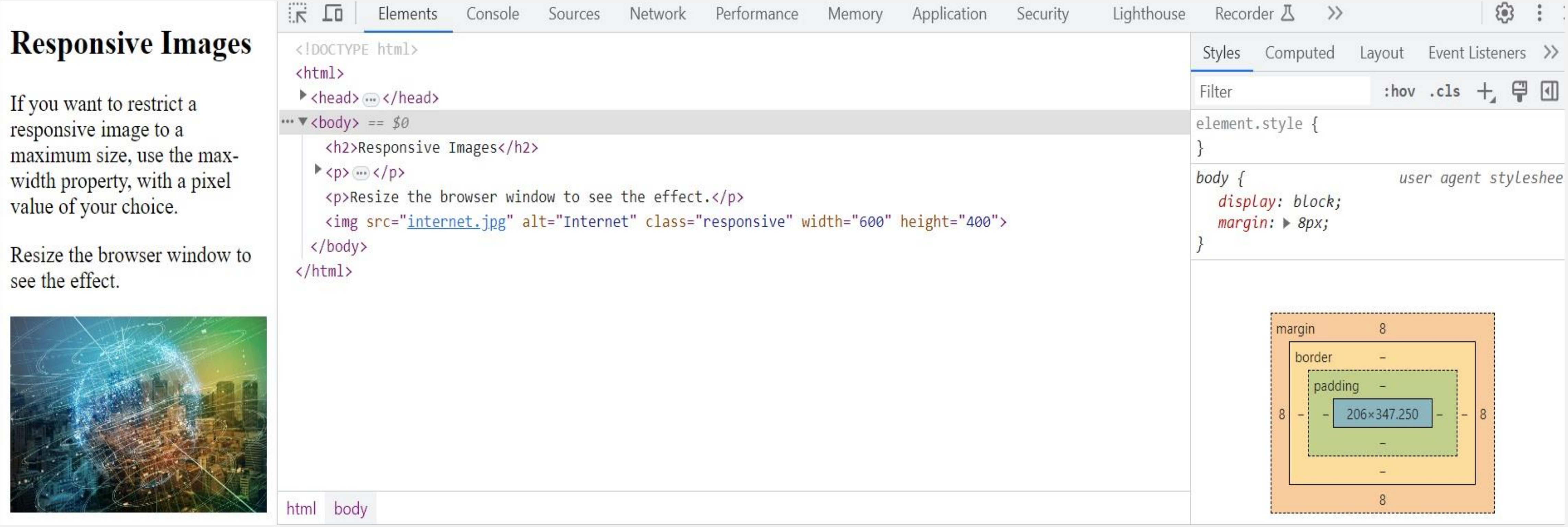
- ***Media Queries***
- ***Responsive Images***

RESPONSIVE IMAGES



```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.responsive {
  width: 100%;
  max-width: 400px;
  height: auto;
}
</style>
</head>
<body>

<h2>Responsive Images</h2>

<p>If you want to restrict a responsive image to a maximum size, use the max-width property, with a pixel value of your
<p>Resize the browser window to see the effect.</p>

<img src="internet.jpg" alt="Internet" class="responsive" width="600" height="400">

</body>
</html>
```

RESPONSIVE IMAGES DEMO



Responsive Images

If you want to restrict a responsive image to a maximum size, use the max-width property, with a pixel value of your choice.

Resize the browser window to see the effect.

```
<!DOCTYPE html>
<html>
 ▶<head>…</head>
…▼<body> == $0
    <h2>Responsive Images</h2>
   ▶<p>…</p>
    <p>Resize the browser window to see the effect.</p>
    <img src="internet.jpg" alt="Internet" class="responsive" width="600" height="400">
  </body>
</html>
```

```
element.style {
}
body {                          user agent styleshee
  display: block;
  margin: ▶ 8px;
}
```

FLEXBOX AND GRID

Now we are ready to move on from the relatively simple CSS layout methods we've looked at so far, and can look at Flexbox and Grid, two modern CSS layout methods that allow us to create complex, flexible layouts for our content.

Flexbox

- Basic concepts of flexbox - MDN tutorial on Flexbox,

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

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

CSS GRID

CSS Grid

- Basic concepts of grid - MDN tutorial on grids, https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Basic_Concepts_of_Grid_Layout
- A complete guide to CSS Grid

https://css-tricks.com/snippets/css/complete-guide-grid/

RELATED READING

- Using Media Queries For Responsive Design In 2018, https://www.smashingmagazine.com/2018/02/media-queries-responsive-design-2018/
- Flexbox the Animated Tutorial - a nice way of learning about different CSS-Flex properties,

https://ghost-together.medium.com/flexbox-the-animated-tutorial-8075cbe4c1b2

- Grid by Example - a useful site with examples of how to use CSS Grid

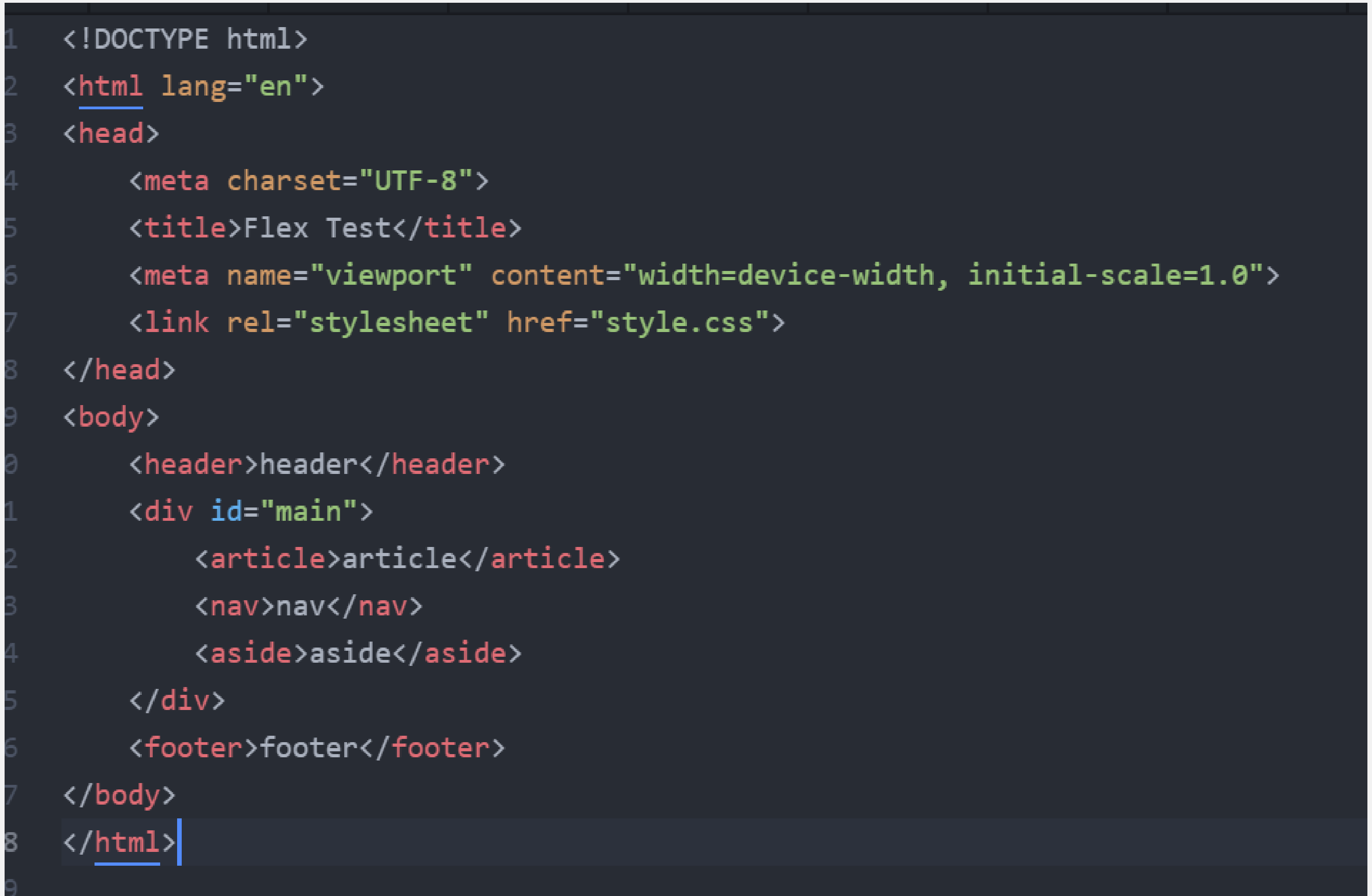
https://gridbyexample.com/examples/

CODE EXAMPLES

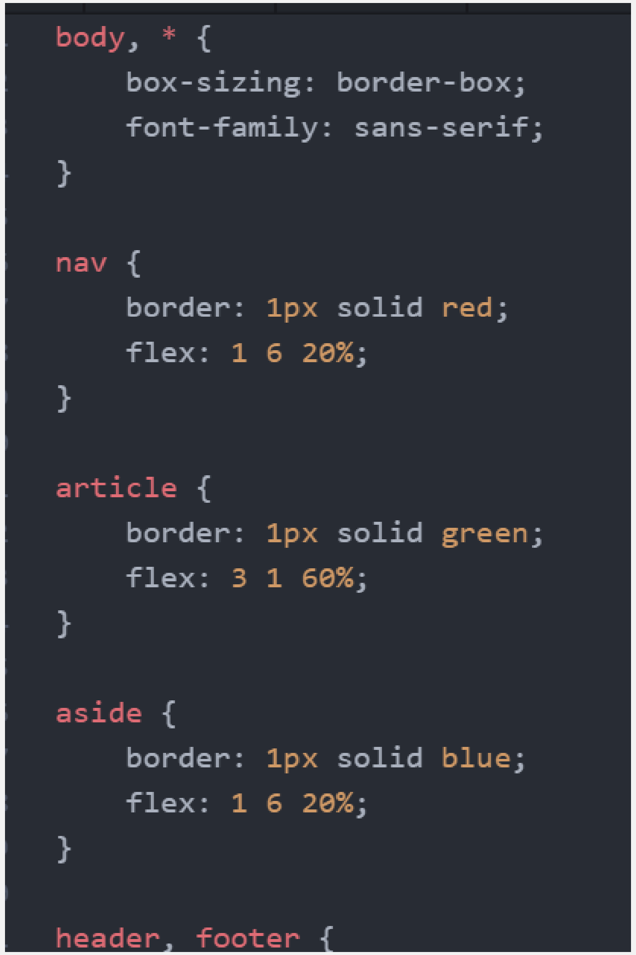
There are a set of code examples that accompany this course showing the use of the fundamental HTML tags and CSS styles/rules that we look at.

- ***Simple grid (without CSS-Grid)***
- ***Flexbox example***
- ***Responsive Flexbox example***
- ***Responsive CSS Grid***

RESPONSIVE FLEXBOX EXAMPLE

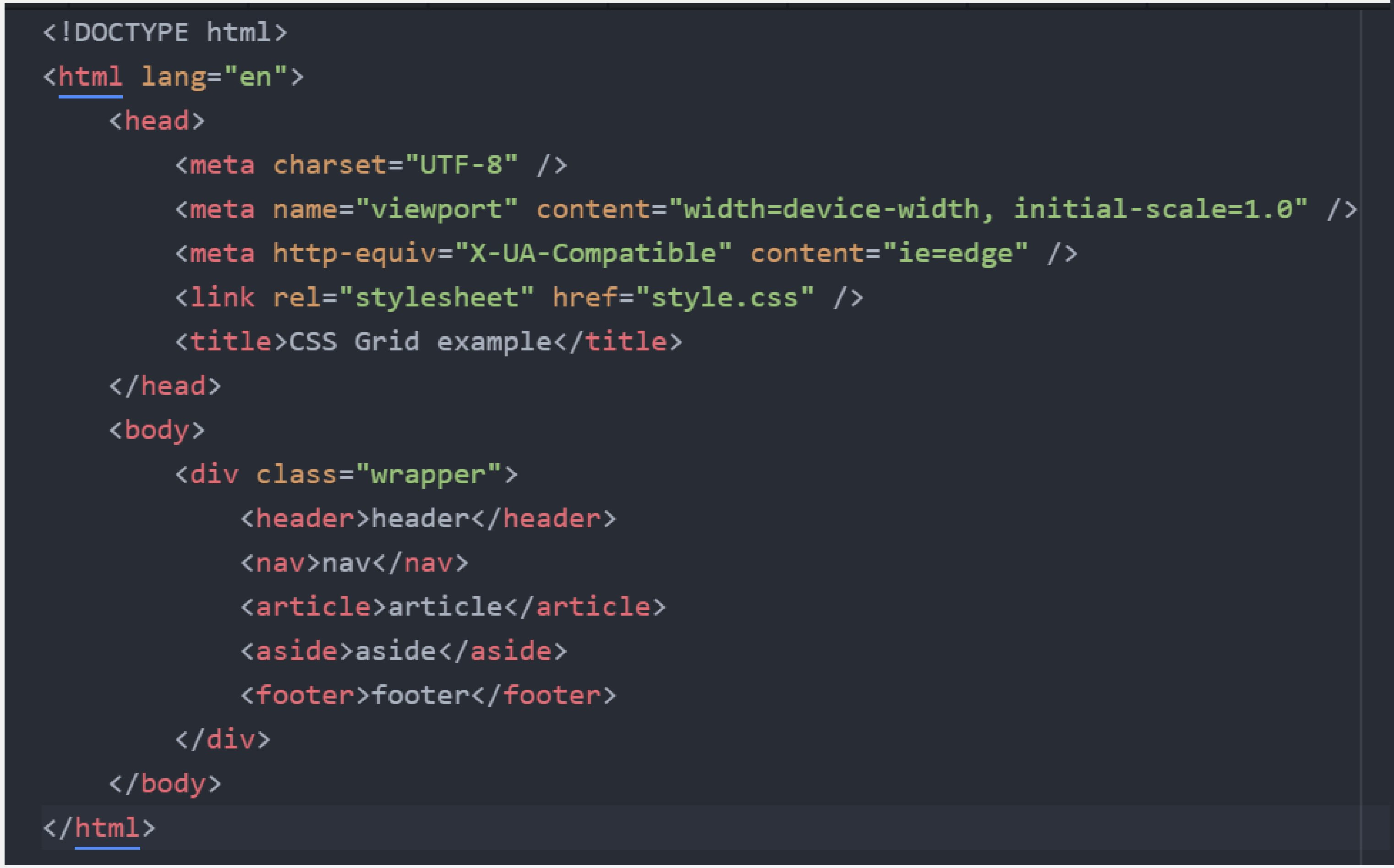


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Flex Test</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>header</header>
    <div id="main">
        <article>article</article>
        <nav>nav</nav>
        <aside>aside</aside>
    </div>
    <footer>footer</footer>
</body>
</html>
```

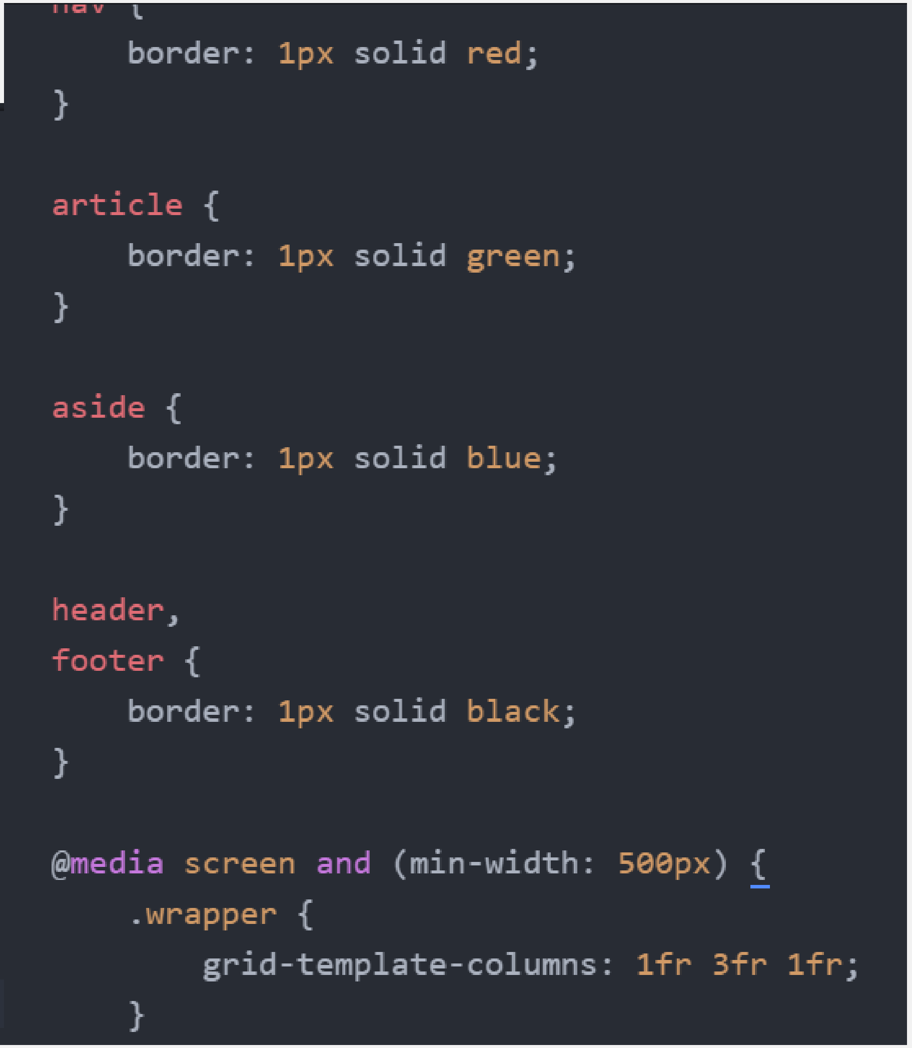


```css
body, * {
    box-sizing: border-box;
    font-family: sans-serif;
}

nav {
    border: 1px solid red;
    flex: 1 6 20%;
}

article {
    border: 1px solid green;
    flex: 3 1 60%;
}

aside {
    border: 1px solid blue;
    flex: 1 6 20%;
}

header, footer {
```

RESPONSIVE CSS GRID EXAMPLE



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <meta http-equiv="X-UA-Compatible" content="ie=edge" />
        <link rel="stylesheet" href="style.css" />
        <title>CSS Grid example</title>
    </head>
    <body>
        <div class="wrapper">
            <header>header</header>
            <nav>nav</nav>
            <article>article</article>
            <aside>aside</aside>
            <footer>footer</footer>
        </div>
    </body>
</html>
```

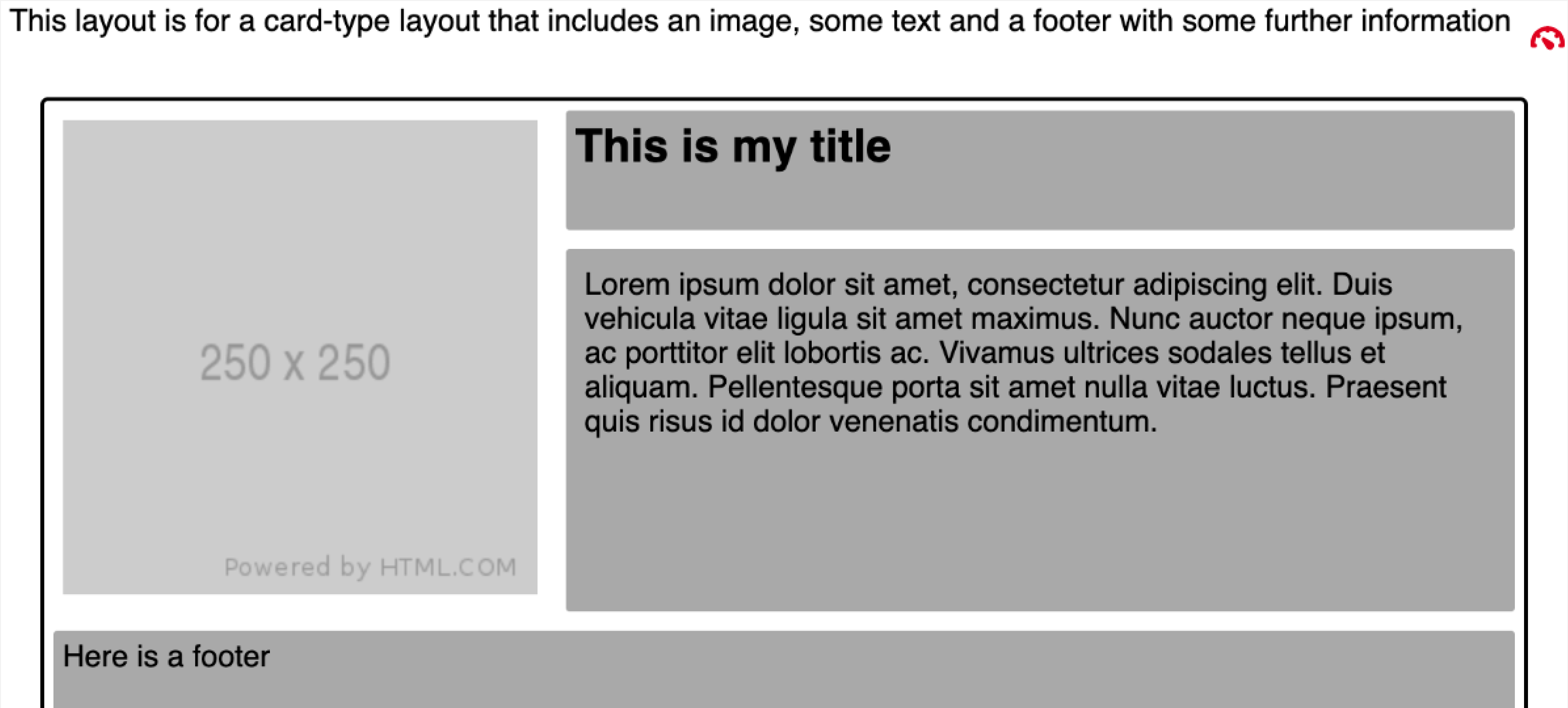


```
nav {
    border: 1px solid red;
}

article {
    border: 1px solid green;
}

aside {
    border: 1px solid blue;
}

header,
footer {
    border: 1px solid black;
}

@media screen and (min-width: 500px) {
    .wrapper {
        grid-template-columns: 1fr 3fr 1fr;
    }
```

EXERCISE 3: RESPONSIVE WEB DESIGN

- The task in this exercise is to practice using the CSS layout techniques we have been looking at for the last few weeks
- Working individually, you should look at the CSS layouts below and attempt to build each one using as many different layout techniques as possible. You can use floats, flexbox, CSS-Grid, absolute/relative positioning, whatever methods you like. You should try to minimise the amount of CSS code required.
- As you build each version of the layout, pay attention to how difficult it is, how much code it needs, and how responsive the resulting layout is.

LAYOUT 1 – MEDIA CARD



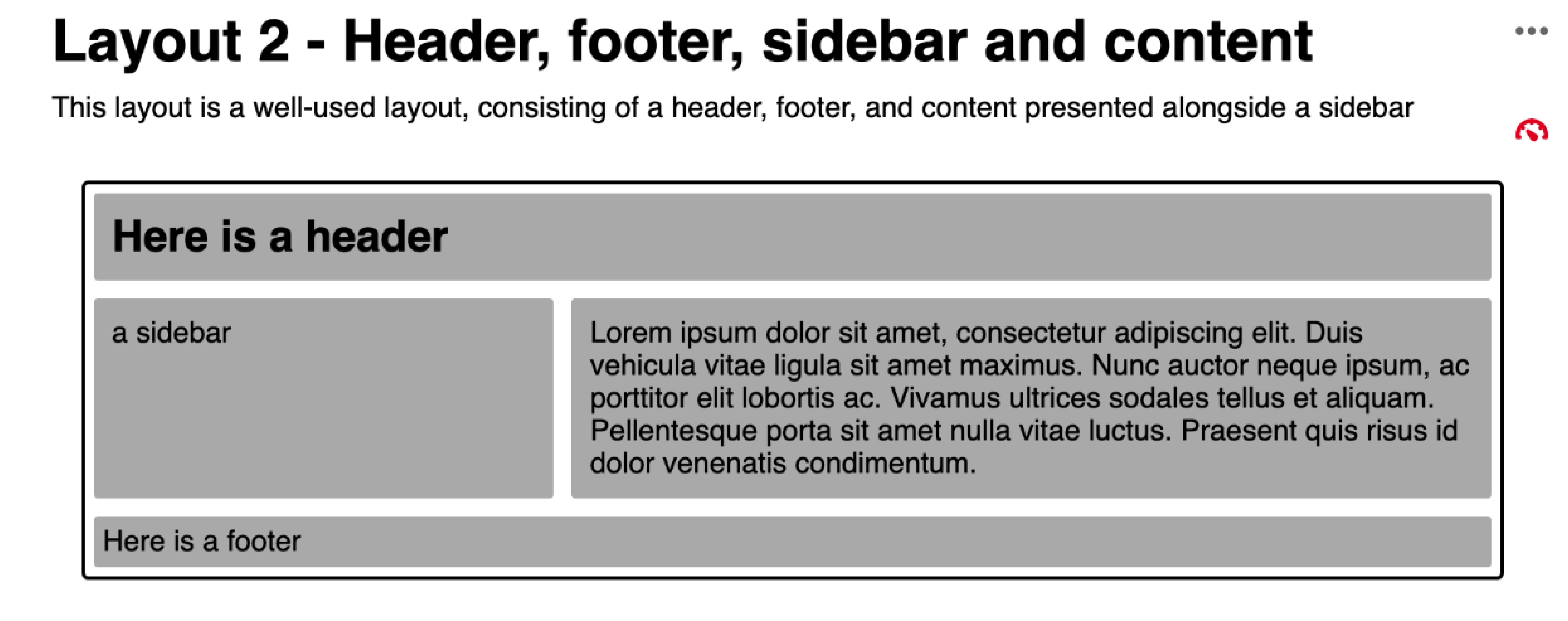
This layout is for a card-type layout that includes an image, some text and a footer with some further information

This is my title

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.

Here is a footer

LAYOUT 2 – HEADER, FOOTER, SIDEBAR AND CONTENT



Layout 2 - Header, footer, sidebar and content

This layout is a well-used layout, consisting of a header, footer, and content presented alongside a sidebar

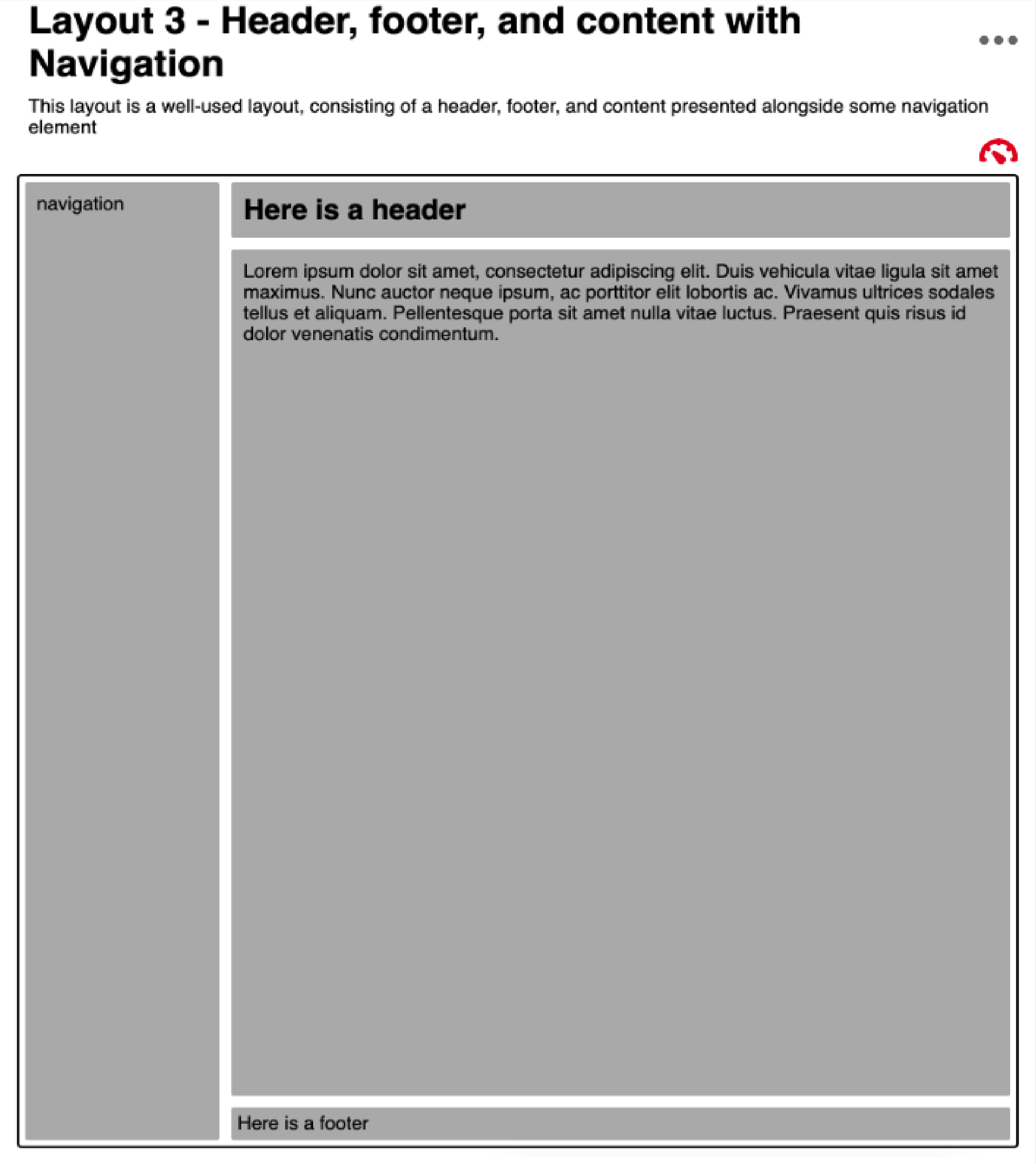
Here is a header

a sidebar

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.

Here is a footer

LAYOUT 3 – HEADER, FOOTER AND CONTENT WITH NAVIGATION



Layout 3 - Header, footer, and content with Navigation

This layout is a well-used layout, consisting of a header, footer, and content presented alongside some navigation element

navigation

Here is a header

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis vehicula vitae ligula sit amet maximus. Nunc auctor neque ipsum, ac porttitor elit lobortis ac. Vivamus ultrices sodales tellus et aliquam. Pellentesque porta sit amet nulla vitae luctus. Praesent quis risus id dolor venenatis condimentum.

Here is a footer

THANK YOU VERY MUCH!

Dr. Daniela Tsaneva
Email: TsanevaD1@cardiff.ac.uk