

# RWD Layout – Walkthrough Instructions

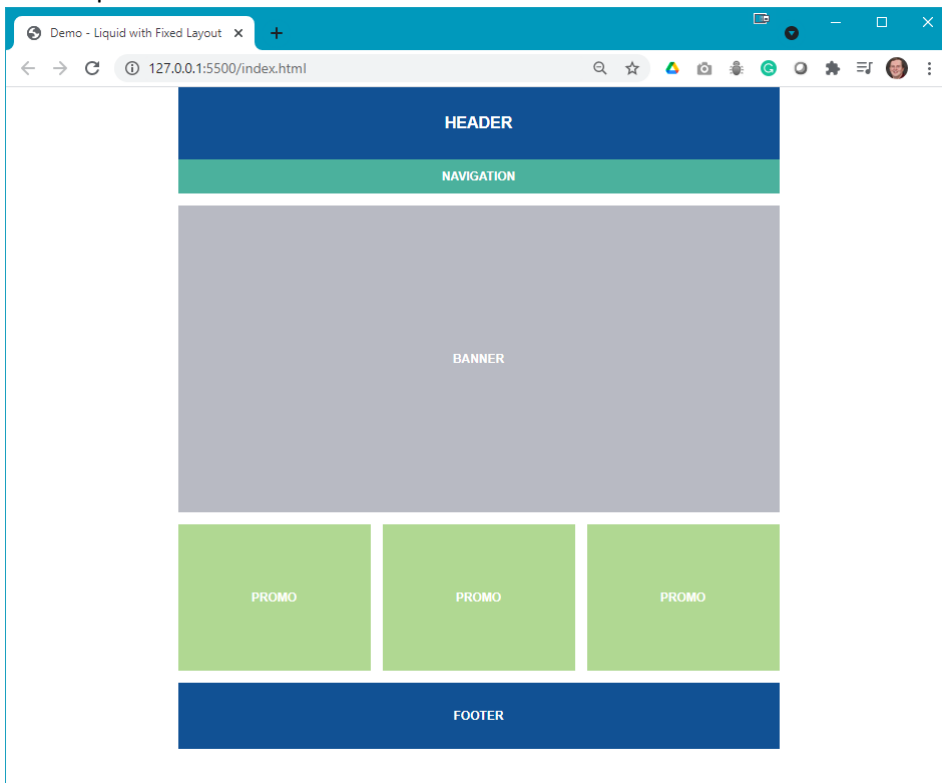
---

## Demo Instructions

You can follow along with your instructor to complete this build and/or you can use this document as a guide in completing the demo build.

## Steps

1. Download the **demo-liquid-fixed-layout.zip** file from Moodle and extract its contents to a folder named **demo-liquid-fixed-layout**.
2. The output of this demo should look like:



3. The first step is to code the layout in **index.html** (all classes shown in the code will need to be added to your **styles.css** file):
  - a. Create a container:

```
<body>
  <div class="container">
    </div> <!-- end container -->
</body>
```
  - b. Add a <header> block inside the container:

```
<header>
  <h1>Header</h1>
  <nav>
    <h2>Navigation</h2>
  </nav>
</header>
```

- c. Add the `<main>` with a `<section>`:

```
<main>
  <section class="banner">
    <h2>banner</h2>
  </section>
</main>
```

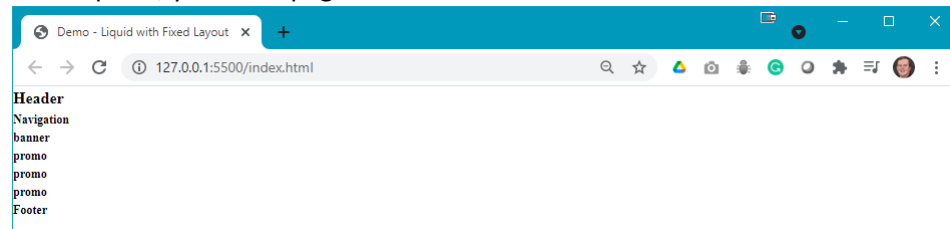
- d. Add a flex container below the `<section>` created above:

```
<div class="flex-container">
  <section class="promo">
    <h2>promo</h2>
  </section>
  <section class="promo">
    <h2>promo</h2>
  </section>
  <section class="promo">
    <h2>promo</h2>
  </section>
</div> <!-- end flex container -->
```

- e. Add a `<footer>`:

```
<footer>
  <h2>Footer</h2>
</footer>
```

At this point, your web page should look like:



4. Now you need to add style rules to your **styles.css** file:

- a. Style the `<body>` element with generic CSS; this will be mobile first:

```
/* Generic CSS, mobile first therefore elements fit edge to edge of the browser
screen and will grow as the screen increases.*/
body {
  /* add base body font properties here */
  text-align: center;
  color: #fff;
  font-weight: bold;
  font-family: sans-serif;
  text-transform: uppercase;
}
```

- b. Style the `<header>` and `<footer>` elements:

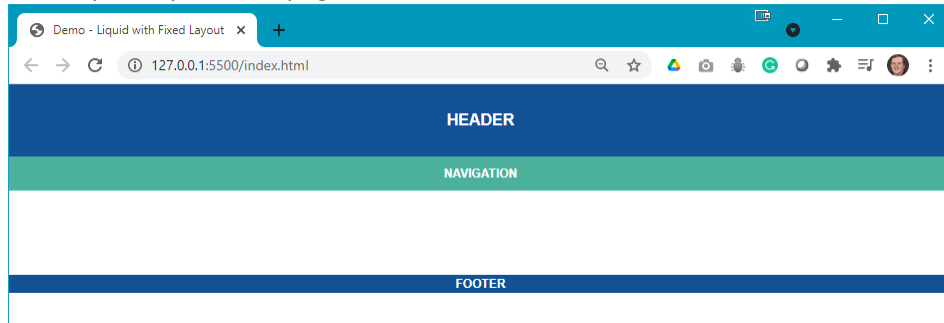
```
header,
footer {
  background-color: #115194;
  margin-bottom: 24px;
}

header > h1 {
  padding: 48px;
}
```

- c. Style the `<nav>` element:

```
nav {  
  background-color: #48B19D;  
  padding: 16px;  
}
```

At this point, your web page should look like:

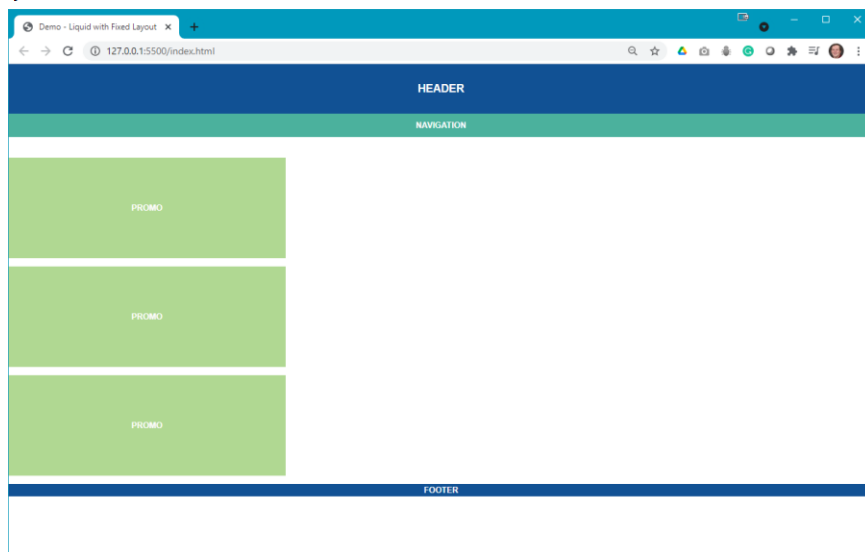


- d. As the `<section>` blocks are not yet being displayed, are they still there? The answer is yes:



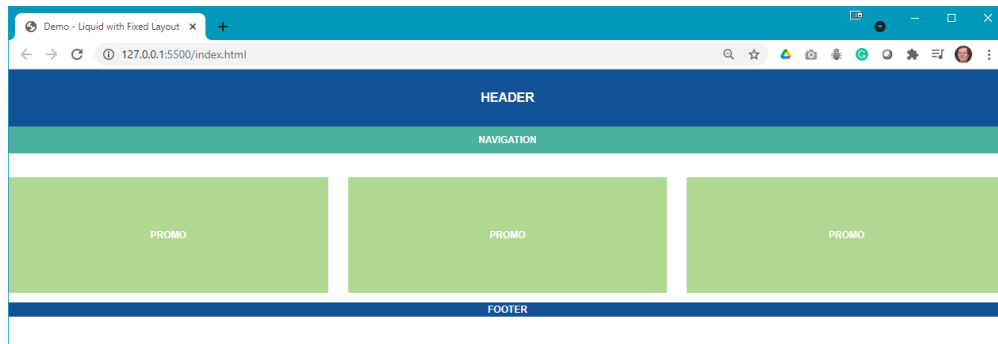
Now style the `.promo` class:

```
.promo {  
  background-color: #B0D892;  
  margin-bottom: 24px;  
  padding: 128px 16px;  
  /* liquid or fluid layout is done in % so that they grow with the screen  
  width*/  
  width: 32%;  
  /* three columns across, each would be 33.33% but if we make them 32% each,  
  there will be 4% of space left over for gutter spacing */  
}
```



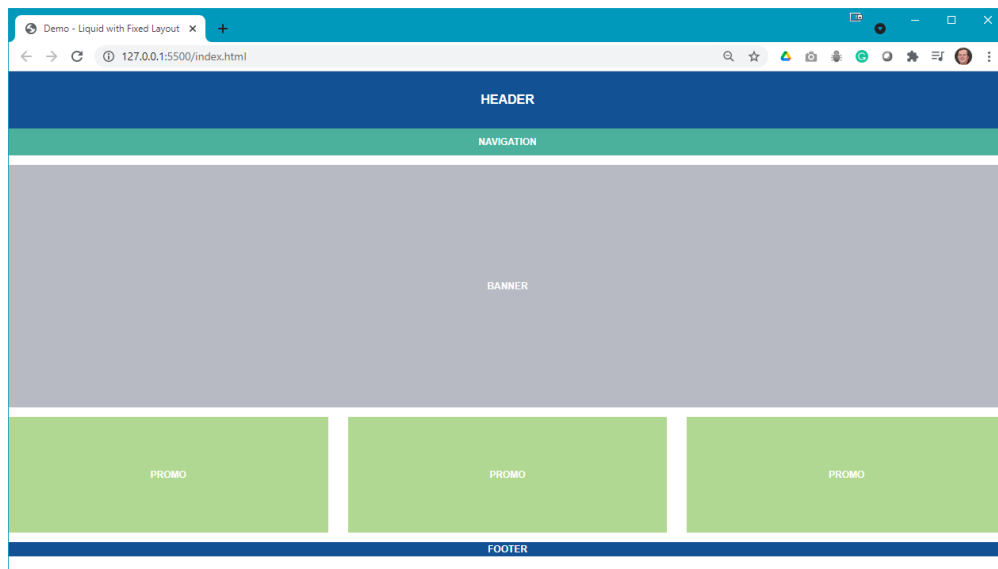
- e. You need each of the promo blocks to be side-by-side, equally spaced, thus add the **.flex-container** class:

```
.flex-container {  
  display: flex;  
  /* take the leftover 4% of available space and create gutters between the  
  columns */  
  justify-content: space-between;  
}
```



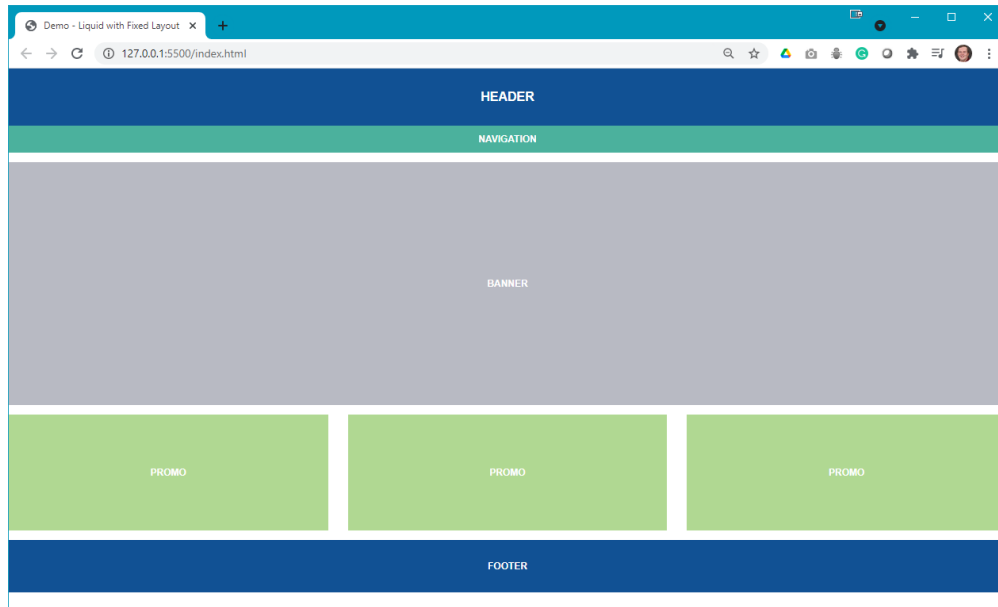
- f. You still need the **.banner** class, which is:

```
.banner {  
  background-color: #B8BAC3;  
  margin-bottom: 24px;  
  padding: 288px 16px;  
}
```



- g. Next add some padding for the <footer> element:

```
footer {  
  padding: 48px;  
}
```

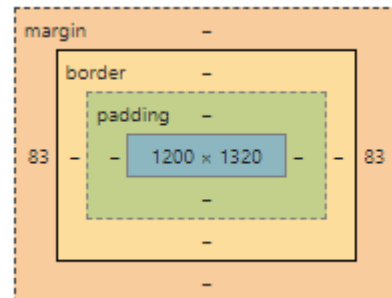
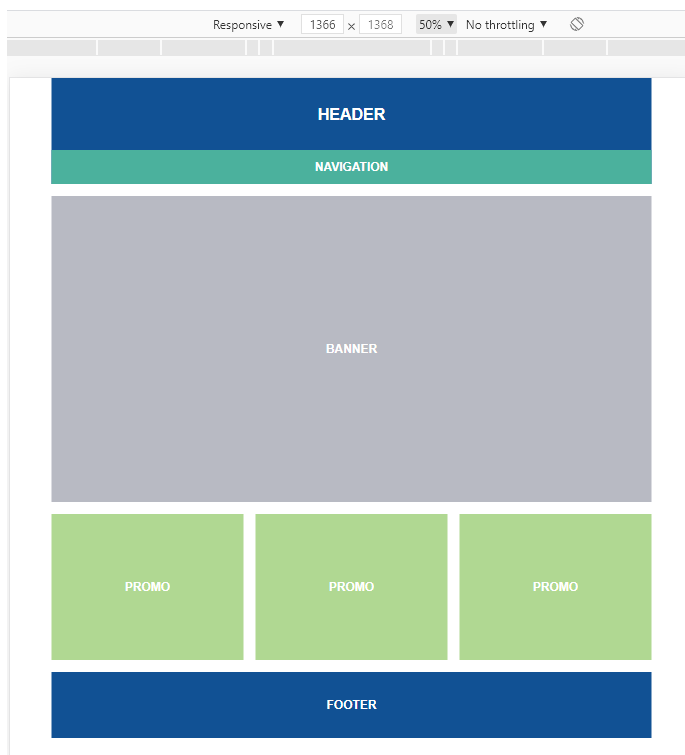
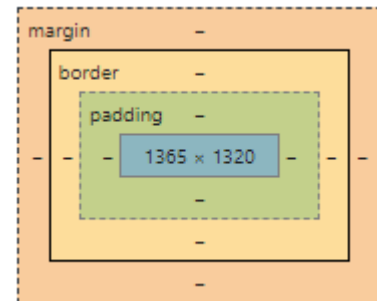
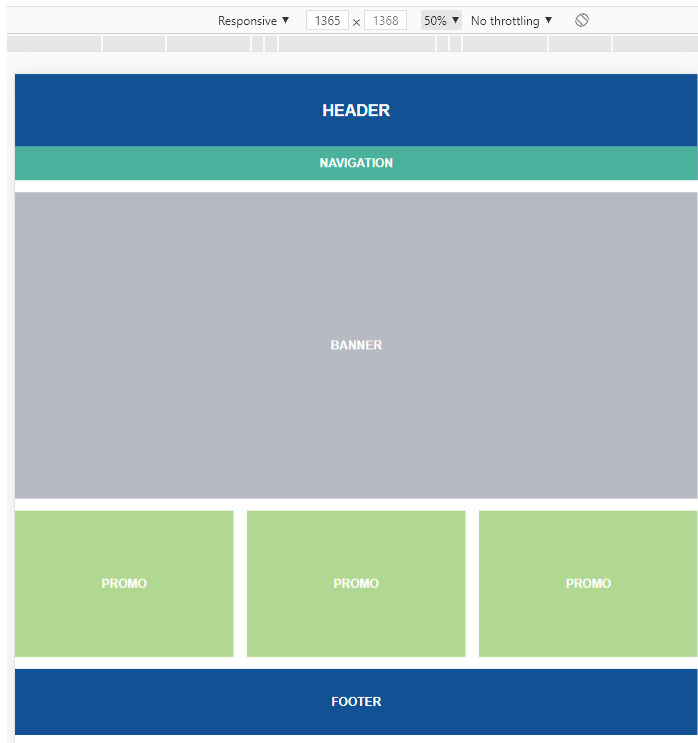


5. The next step is to add a media query that will make the web page change at a specific width.

Add the following to your **styles.css** file:

```
/* liquid or fluid layouts are great for mobile devices, but on desktops we often  
want to switch to a fixed width container */  
/* we will use a media query to change our layout for devices with a screen size  
wider than 1365px */  
@media screen and (min-width: 1366px) {  
  /* the CSS rules in here will only apply to screens 1366px and wider */  
  .container {  
    /* the width of the container should be smaller than the min-width  
    conditon to allow for margins on the left & right */  
    width: 1200px;  
    margin: 0 auto;  
  }  
}
```

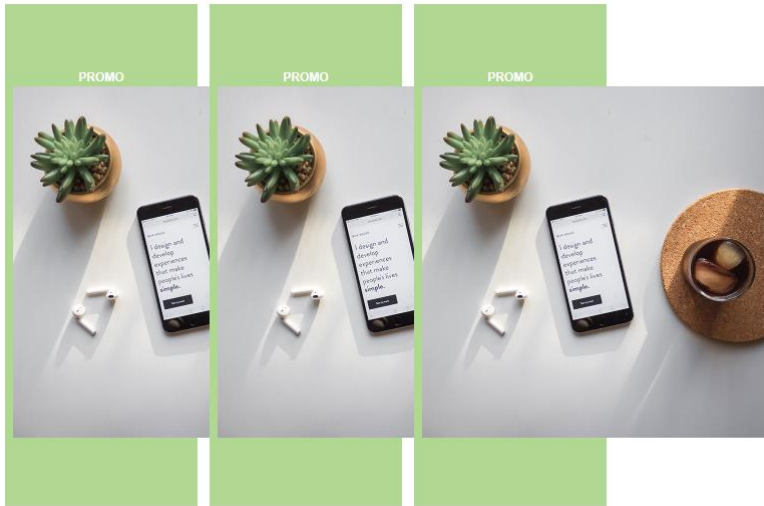
6. You need to test this using the browser's developer tools. Remember the min-width: 1366 px; thus, once the page width exceeds the minimum width (larger screen resolution) the web page will change:



- The next step is to see how images behave in this layout. Add the following code below the `<h2>` in each `<section class="promo">`:

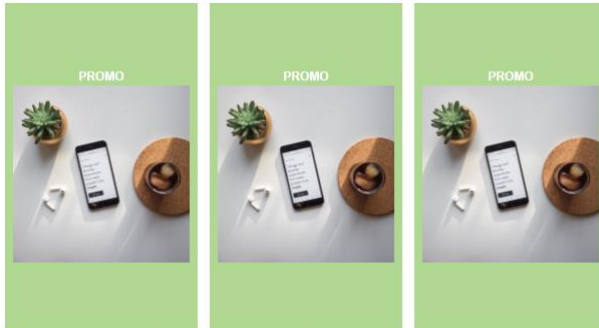
```
<section class="promo">
  <h2>promo</h2>
  
</section>
```

8. Notice how the images have broken the layout:



9. To fix this there needs to be a style rule to set the maximum width of the `<img>` element to 100%. Open the **reset.css** file and scroll down to line 52. Replace the `÷` symbol with the following code:

```
/* Make images easier to work with */
img{
  max-width: 100%;
}
```



10. Using the browser's developer tools, change the width of the web page and notice how the images scale.

## Exercise

Open the **flexbox-exercise-instructions.docx** file and follow the instructions to complete the exercise for this lesson.