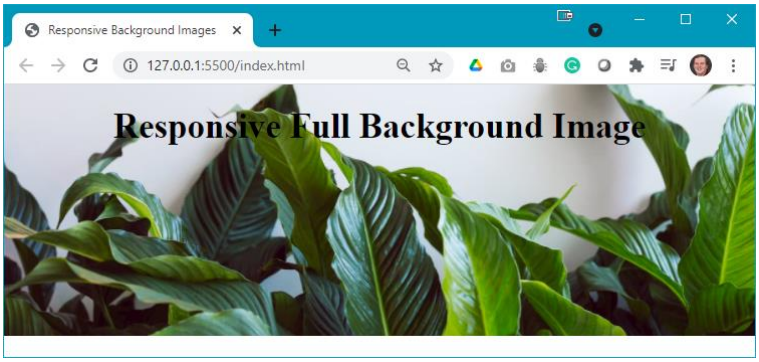
RWD- Images – Walkthrough Instructions

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.

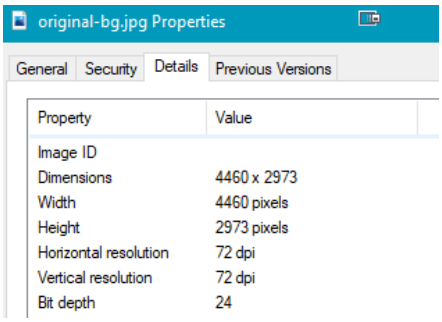
Steps

1. Download the **demo-responsive-bg.zip** file from Moodle and extract its contents to a folder named **demo-responsive-bg**.
2. The completed output should look like:



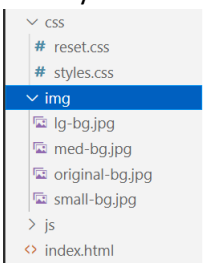
Note: When the browser is resized, the image will also resize.

3. The first thing to do is find the size of the original image; this is needed as a starting point for this demo:



4. Use Photoshop or Squoosh to resize the image (save each resized image but do not overwrite the original file):
    a. 600px x 400px: name this **small-bg.jpg**
    b. 1200px x 800px: name this file **med-bg.jpg**
    c. 1800px x 1200px: name this file **lg-bg.jpg**

Note: you should now have 4 images in your **img** folder:



5. You will need to add style rules to the **styles.css** file:
    a. Start with the smallest image:

```
/* Start with the smallest size first - serve a scaled-down version of
the background imgage to be conscious of page load speed and bandwidth  */
/* The original image is 4460px x 2973 px and 950KB. Use Photoshop (or
squoosh.app) to resize the image down to 600px x 400px and run it through
a compression tool to reduce the file size. */
header {
    height: 400px;
    padding: 2rem;
    background-color: #eff2f7;
    /* use shorthand for background-image, background-position (x & y),
    background-size (cover for full screen), background repeat. Note:
    background-size may be included in the shorthand but it must come
    immediately after background-position and separted with a /
    character  */
    background: url(../img/small-bg.jpg) center bottom / cover no-repeat;
}
```

    b. Style the <h1> element:

```
h1 {
    font-size: 2.5rem;
    text-align: center;
}
```

6. As the width of the browser increases, the quality of the image degrades. To fix this we need a media query:
    a. First break point at 850px:

```
/* As the screen gets wider, the background image stretches to fit and
the lower quality of the image no longer looks good on the screen. We can
use media queries to swap out the background image for a larger & higher
quality file size. */
@media only screen and (min-width: 850px){
    header {
        /* Using the same shorthand property as we did above, simply
        replace the image url */
        background: url(../img/med-bg.jpg) center bottom / cover
        no-repeat;
    }

    h1 {
        /* automatically increasing font size */
        font-size: calc(2.5rem + 2vw);
    }
}
```

b. Next break point at 1000px:

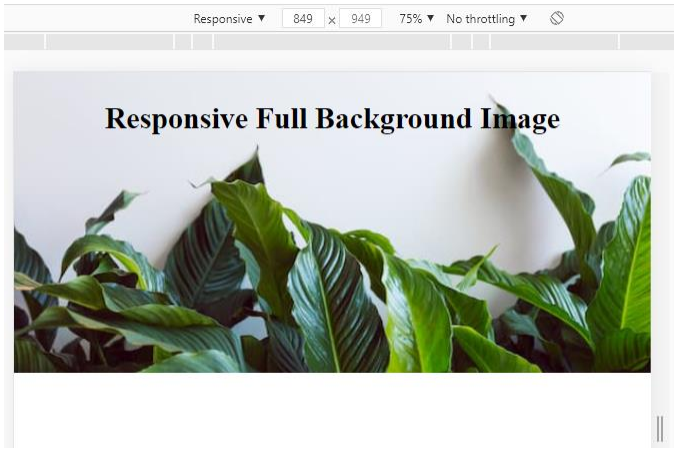
```
/* Adjust the background image (height or positioning) as needed */
@media only screen and (min-width: 1000px){
    header {
        height: 500px;
    }
}
```

c. Finally, a full-sized image for wide screens:

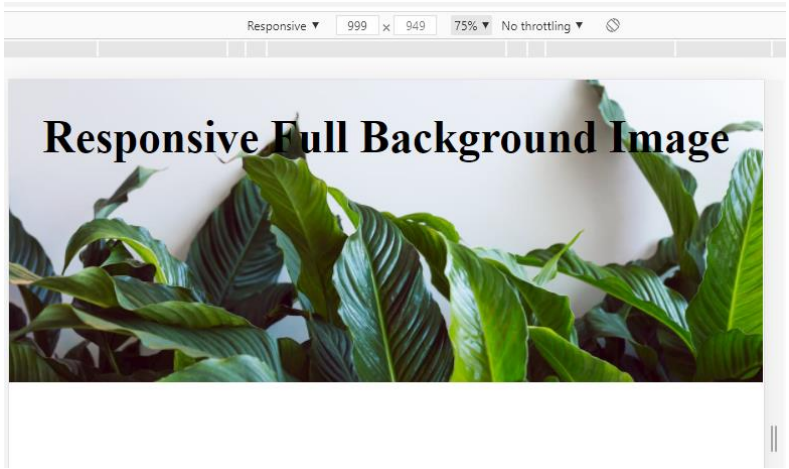
```
/* Current reccommendations are to keep full screen hero and background
images under 250KB */
@media only screen and (min-width: 1500px){
    header {
        background: url(../img/lg-bg.jpg) center bottom / cover no-repeat;
        height: 700px;
    }
}
```

d. Open the browser's developer tools to see:

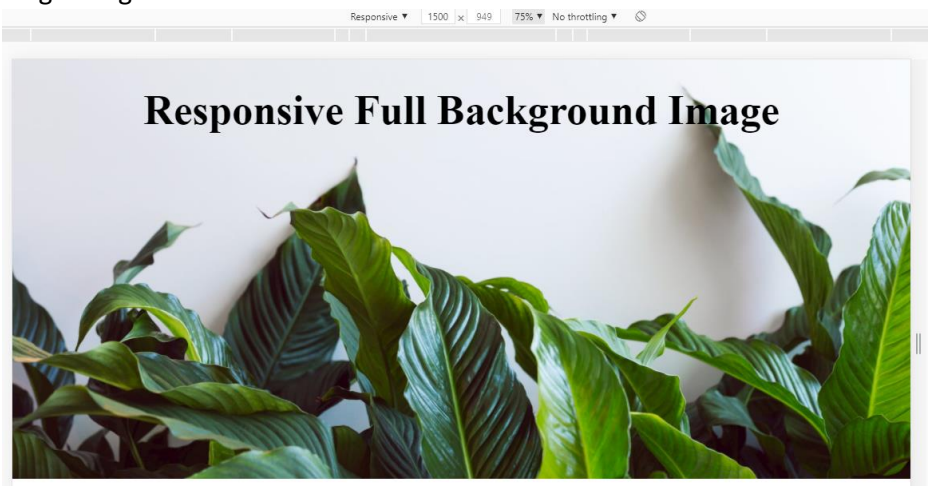
i. Smallest resolution:



ii. Medium image:

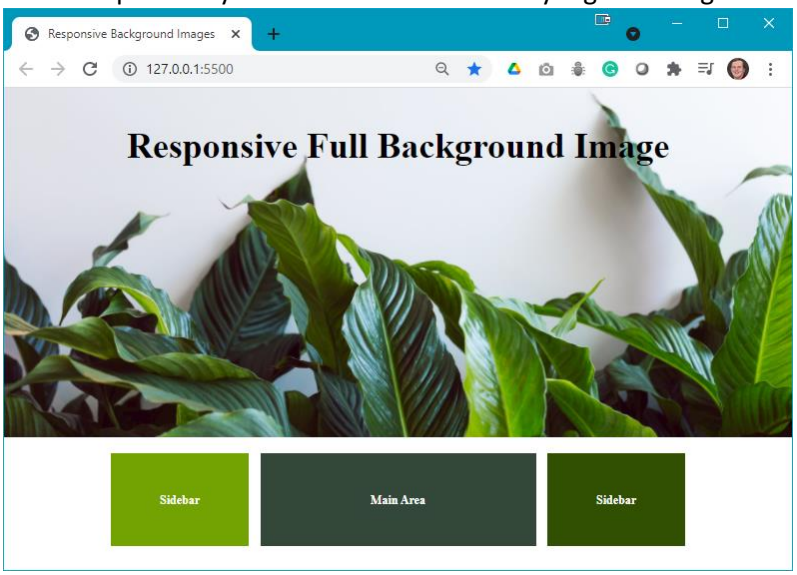


iii. Large image:



Additional Layout

For extra practice you will add content and styling rules to get the output shown below:



1. The **index.html** file needs the following code added below the <header> block:

```
<div class="flex-container">
    <section class="left-sidebar">
        <h2>Sidebar</h2>
    </section>
    <section class="main-content">
        <h2>Main Area</h2>
    </section>
    <section class="right-sidebar">
        <h2>Sidebar</h2>
    </section>
</div>
```

2. The following code needs to be added to the **styles.css** file:
    a. Add a `.flex-container` class:

```
/* additional responsive column layout */
.flex-container {
    display: flex;
    flex-flow: column;
}
```

    b. Style the `<section>` element:

```
section {
    padding: 4.7rem 0;
    color: #fff;
    text-align: center;
}
```

    c. Style the first `<section>` element:

```
section:first-of-type {
    background-color: #73a302;
}
```

    d. Style the second `<section>` element:

```
section:nth-of-type(2) {
    background-color: #334839;
    /* on small mobile screens we will want the main content area to come
    first as it is more important, let us change that order for this
    screen */
    order: -1;
}
```

    e. Style the last `<section>` element:

```
section:last-of-type {
    background-color: #315002;
}
```

    f. Add a media query so you can switch the order back to the original order:

```
/* on wider screens we can switch back to the same order the items are in
the HTML (left sidebar, main, right sidebar) and place them in a row*/
/* have the sidebars each take up 24% of the available row space, the
main content take up 48%, and the leftover space (4%) can be used to
create 2% gutter spacing between the columns */
@media only screen and (min-width: 850px){
    .flex-container {
        margin-top: 2rem;
        flex-flow: row nowrap;
        justify-content: space-between;
    }

    section:nth-of-type(2) {
        order: 0; /* default order position */
        width: 48%;
    }

    section {
        width: 24%;
    }
}
```

g. Add a media query for a fixed-width, edge-to-edge, background image:

```
/* add a fixed width container for very wide screens but allow background
image to go edge-to-edge */
@media only screen and (min-width: 1200px) {
    h1,
    .flex-container {
        max-width: 71.875rem; /* 1150px */
        margin: 2rem auto;
    }
}
```

3. Refresh your browser to see the final output. Additionally, use your browser's developer tools to see the results at different screen widths.