

Exercise: Responsive Design with CSS Media Queries

Some “mobile-friendly” sites remove content and penalize mobile users. We want our “small screen” users to get the same content as the desktop people, but get an experience more appropriate for smaller screens.

1) Open up `darth.html` in a web browser



The above is what it looks like with no CSS.

Go ahead and open up `darth.html` in a text editor and look over the HTML structure. You'll notice that we have 3 main parts to the page: a `<header>` element, a `<section>` element with the `id = "main"`, and an `<aside>` element.

They contain the following child elements:

A) `<header>` contains:

- a "skip nav" section

- a `<h1>`

- the regular navigation (home, about, work, join me)

- a `<div>` of `id="banner"` that contains the Death Star image

B) `<section>` contains:

- "Facts about Darth Vader"

C) `<aside>` contains:

- "Empire Job Board"

2) Open up `main.css` in a text editor. You'll notice we have a standard style reset at the top of the page.

Add the following to the `/* Logo /* and /* Nav */` sections:

```
/* Logo H1 */
header h1 {
    height: 70px;
    width: 160px;
    float: left;
    display: block;
    background: url(../images/darth.png) 0 0 no-repeat;
    text-indent: -9999px;
}
```

The `h1` is getting a background image, getting floated left, and getting its width set to a low number so there is room for the nav system (and so the float works).

The negative `text-indent` is to hide the text from the user (so that the background image of "Darth" is visible). The purpose of having the text is so that the search engines can see it. See a discussion of this technique here:

<http://www.dennisplucinik.com/blog/2007/09/01/css-hack-text-indent-10000px/>

Here's the Nav CSS:

```
/* Nav */
header nav {
    float: right;
    margin-top: 40px;
}

header nav li {
    display: inline;
    margin-left: 15px;
}

#skipTo {
    display: none;
}

#skipTo li {
    background: #b1fffc;
}
```

The purpose of these rules is to get the navigation list to display all in one line, rather than in a row.

We are also hiding the “skip to” navigation, and will only reveal it once we are in “mobile mode”

Preview the page - it looks better - but if you re-size it you will see that it is fixed in size and not at all responsive.

3) Now add the banner CSS below:

```
/* Banner */
#banner {
    float: left;
    margin-bottom: 15px;
    width: 100%;
}

#banner img {
    width: 100%;
}
```

Now try scaling the page - notice that the image scales down nicely - but also scales up as big as you want to grow the browser window.

You might have also noticed that our “main” div - “Facts about Darth Vader” is at the top of the screen. We need to fix that.

4) Type the following into `main.css` for `#main` and `<aside>` - put it in the `/* Structure */` section:

```
#main {
    width: 60%;
    margin-right: 5%;
    float: left;
}

aside{
    width: 35%;
    float: right;
}
```

Reload the page. Note that we now have `#main` and `<aside>` where we want them, with a nice two column layout underneath the nav Death Star image.

5) To give our page a max-width so that the user can't infinitely scale our image, and to add a little padding around the content, add the following to the `/* Structure */` section:

```
#wrapper {  
    width: 96%;  
    max-width: 920px;  
    margin: auto;  
    padding: 2%;  
}
```

The “non-mobile” version of this site is done, now let's use a media query to give different CSS to a mobile device.

6) Type in the following on the bottom of `main.css`:

```
/* Media Queries */  
@media screen and (max-width: 480px) {  
  
    #skipTo {  
        display: block;  
    }  
  
    header nav, #main, aside {  
        float: left;  
        clear: left;  
        margin: 0 0 10px;  
        width: 100%;  
    }  
    header nav li {  
        margin: 0;  
        background: #efefef;  
        display: block;  
        margin-bottom: 3px;  
    }  
    header nav a {  
        display: block;  
        padding: 10px;  
        text-align: center;  
    }  
}
```

What should this do?

#1 - Show the “Skip Nav” link

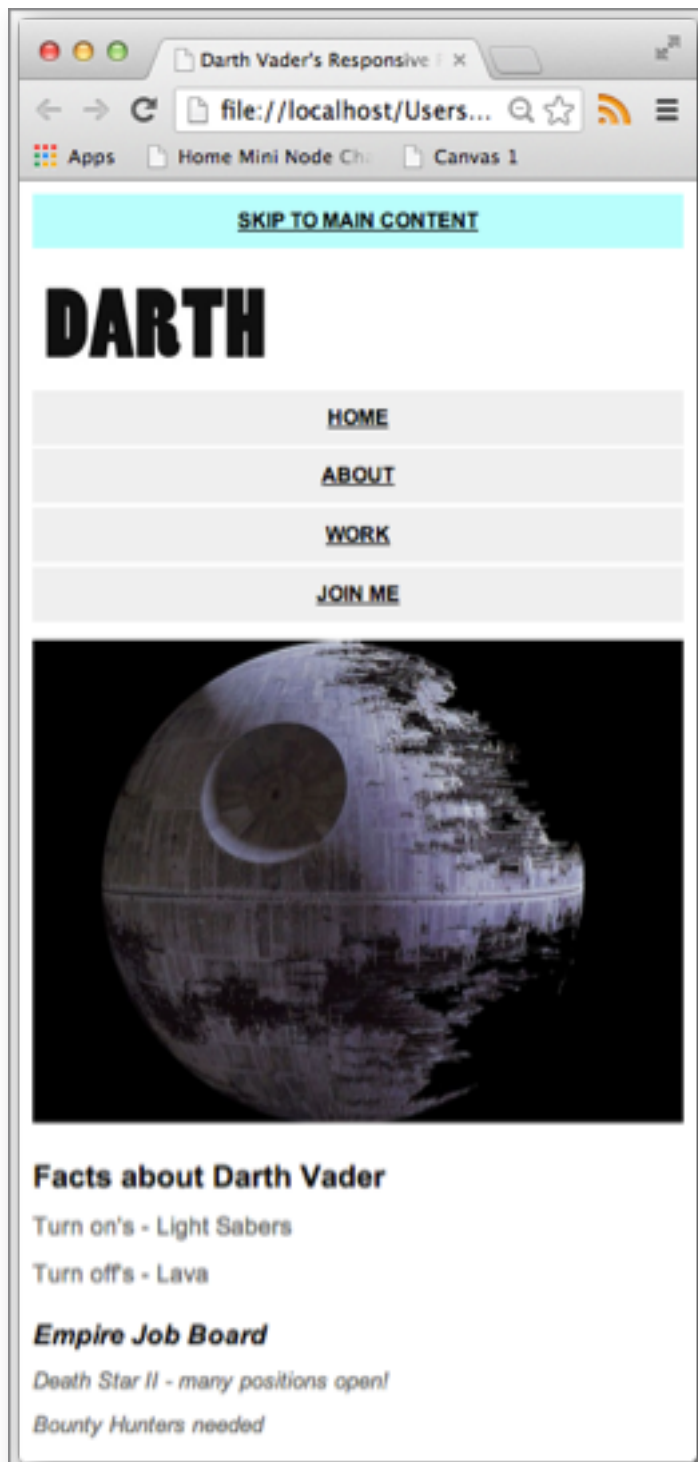
#2 - go to a single column layout - with nav, #main, and aside on top of one another

#3 - display the navigation links in a row again, and give them a lot of padding so that they look like “table cells” from iOS or Android.

Reload the page - no change.



Resize the page to mobile size - change!



Now, if you try this on a smartphone, depending on the model, you may notice that some text is smaller than you want or other issues pop up. This is because the smartphone scales everything on the page, which you can correct by adding the following in the <head> of your HTML:

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

Submission: Make sure your final version is uploaded and tested on Banjo, and linked from your main 230 page. (Don't use the existing folder with a name full of spaces!)