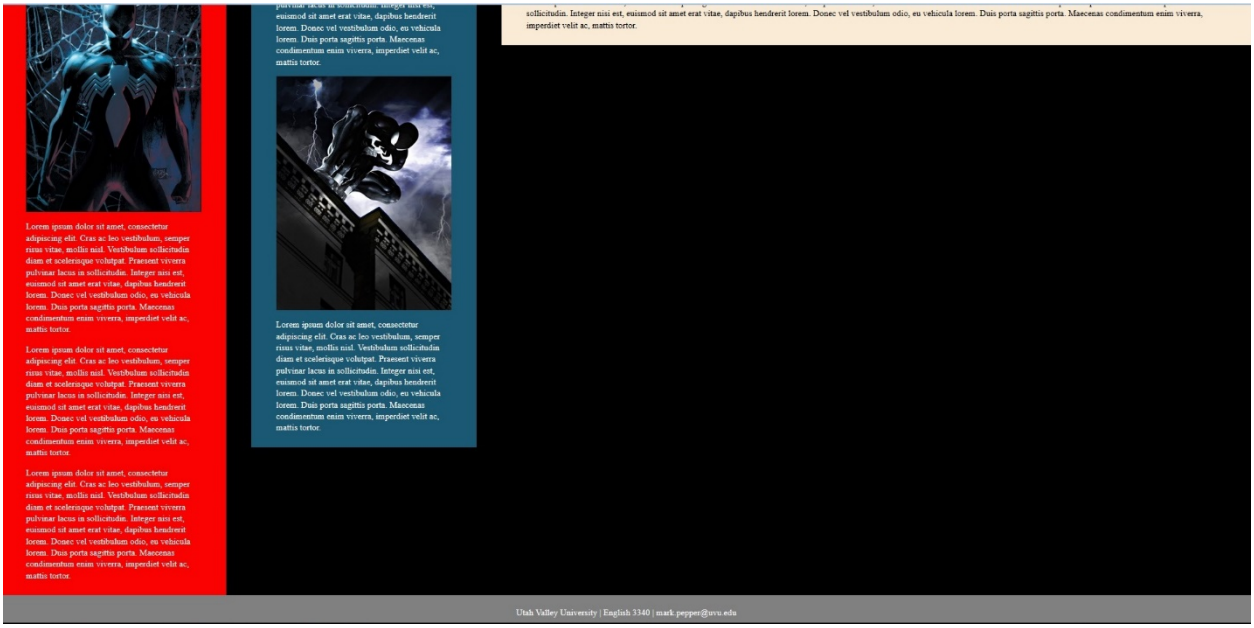
Equal Columns with Flexbox

Flexbox is a relatively new feature of CSS that is very powerful, but also not fully compatible yet with older browsers. Full layout with flexbox is possible (eliminating the need for all height, width, and positioning commands we've learned), but I don't recommend it. But, flexboxes do easily help a very old problem that had very complicated solutions in the past: equal column heights.

All Columns Equal Height

Let's start from here where we have three fluid columns. How do we easily make them all the same height?



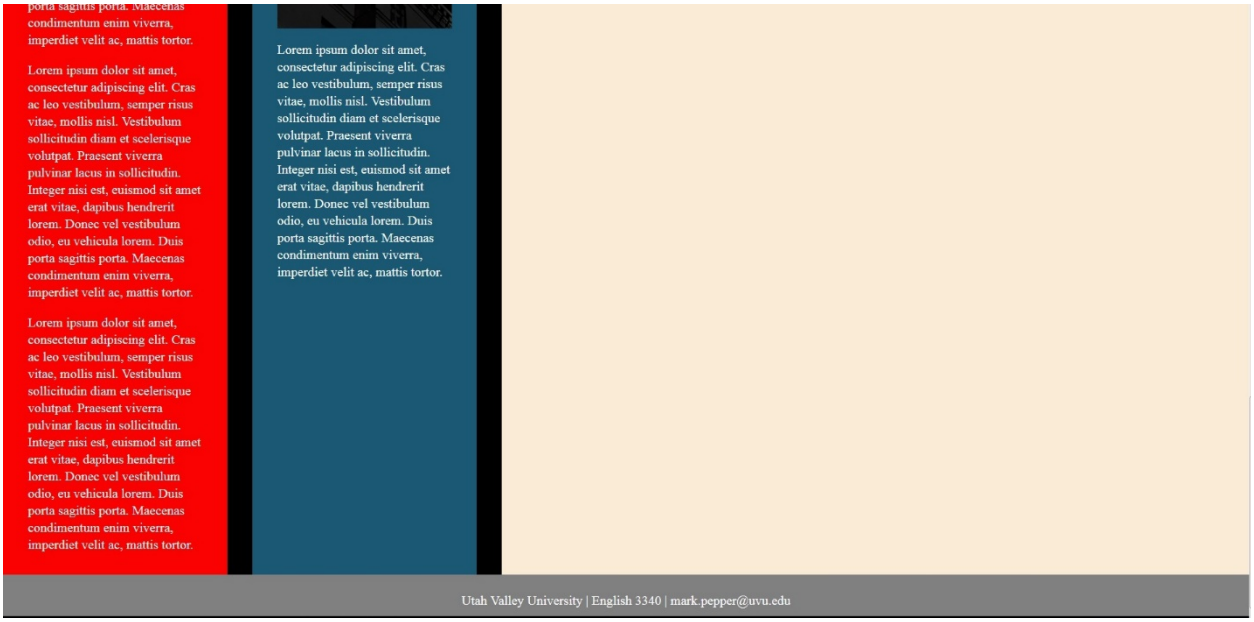
Step one is to wrap all of the columns in a new that we'll give an ID of "flexcontainer." When you wrap items in a flexcontainer, that container becomes the parent and all children (the columns) become flex-capable.

So, in the HTML, place immediately above the opening of the left column and place the closing immediately after the that closes the right column.

In the CSS, do this:

```
#flexcontainer {
        display: flex;
        }
```

This tells that container to let its children flex. The default of flex is to stretch to the bottom of the container. It really is that easy! Make sure the columns still have width percentages and margin for gutters. **You don't even need height: auto anymore, because the flex is taking care of that.**



Only 2 Equal Columns

What if the left and middle columns are done, but we know the right column is going to get a lot more content and grow far past them? In that case, we might only want the left and middle to be equal—but not the right. It's a bit trickier.

- Move the closing for the flexcontainer to come after the closing of the middle column (you're only wrapping the ones you want to be equal)

Since a flexcontainer is 100% width by default (and that worked fine in the previous example), now we need to give the flexcontainer the exact width of the two columns we're working with. In this case, that's 40% (remember to include margin and

padding). Finally, we will float the flexcontainer left. Make sure the final column (the right one) is floating right.

```
#flexcontainer {
        display: flex;
        float: left;
        width: 40%;
}
```

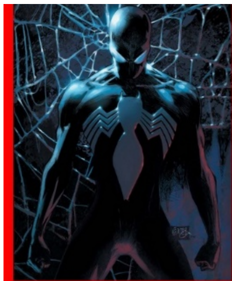
However, this is not enough. Flexed elements aren't affected by floats, and this is one of those cases where having them will throw off our design.

Remember, height: auto is no longer needed. So remove it and remove the floats.

Finally, we have to remove the column widths. Why? Because now they're in a 40% container. So if we kept the width on the column (say 16%, for example), then the column would fill only 16% of that 40% (leading to a tiny column). If we remove the widths on the columns, the natural properties of flexboxes will fill the space perfectly.

```
#leftcolumn {
        padding: 10px 2% 10px 2%;
        background-color: #fa0202;
        color: white;
        margin-right: 2%;
        }
#middlecolumn {
        padding: 10px 2% 10px 2%;
        background-color: #1b5874;
        color: white;
        margin-right: 2%;
        }
```

Remember, flexed elements will not float. But the columns are the flexed elements. The parent flexcontainer can float, because it is a container—it's not actually flexed. This is the only way you'll get the other column to its side. See next page for result.

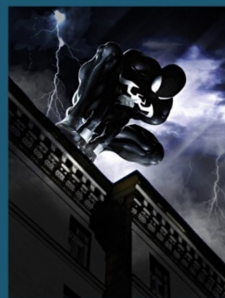


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac leo vestibulum, semper risus vitae, mollis nisl. Vestibulum sollicitudin diam et scelerisque volutpat. Praesent viverra pulvinar lacus in sollicitudin. Integer nisi est, euismod sit amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac leo vestibulum, semper risus vitae, mollis nisl. Vestibulum sollicitudin diam et scelerisque volutpat. Praesent viverra pulvinar lacus in sollicitudin. Integer nisi est, euismod sit amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac leo vestibulum, semper risus vitae, mollis nisl. Vestibulum sollicitudin diam et scelerisque volutpat. Praesent viverra pulvinar lacus in sollicitudin. Integer nisi est, euismod sit amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.

amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac leo vestibulum, semper risus vitae, mollis nisl. Vestibulum sollicitudin diam et scelerisque volutpat. Praesent viverra pulvinar lacus in sollicitudin. Integer nisi est, euismod sit amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ac leo vestibulum, semper risus vitae, mollis nisl. Vestibulum sollicitudin diam et scelerisque volutpat. Praesent viverra pulvinar lacus in sollicitudin. Integer nisi est, euismod sit amet erat vitae, dapibus hendrerit lorem. Donec vel vestibulum odio, eu vehicula lorem. Duis porta sagittis porta. Maecenas condimentum enim viverra, imperdiet velit ac, mattis tortor.