Flexbox menu

Start by showing what we have in the the starting files – index.html and style.css

It’s much of the same content as the clamp/var demo, but now with some added start for a menu. It’s just a bunch of <a> elements instead of a <nav>. We could add some margins manually, but this gets really messy really quickly, and can lead to weird wrapping issues

nav {

background-color: var(--background-secondary);

}

nav a {

color: var(--text);

}

nav a:hover {

color: var(--text-accent);

}

Another option is to use flexbox!

We set display:flex on the container element, and css will automatically work out what to do with the children elements

\*\*be aware, align-items is for vertical alignment (for a row)

We can add justify-content: center to move them to the middle

And add gap: (some value); and let CSS work out how much margin on each side

See style2.css

nav {

background-color: var(--background-secondary);

display: flex;

justify-content: center;

gap: 1em;

}

nav a {

color: var(--text);

}

nav a:hover {

color: var(--text-accent);

}

But there’s a problem – if we add a bunch of children elements, the navbar will overflow! That’s not good. This is because the default wrapping behavior is to use “no-wrap” – we can change this by specifying flex-wrap:wrap

nav {

background-color: var(--background-secondary);

display: flex;

justify-content: center;

gap: 1em;

flex-wrap: wrap;

}

See style3.css

And notice how it automatically works out the gap in *all* directions

Flexbox direction

By default, flexbox assumes you want all your content to be arranged in a row. But we can use flex to space out columns as well

We just have to set flex-direction: column (there are also reverse-row and reverse-column)

See style4.csside

On its own this won’t look very good – because the nav is still taking up 100% of the width.

But we can use width: fit-content and height: 100vh; to size it based on its children and the window size – and then use float:left; to move it to the side.

And finally change justify-items: start; to put things back to the top

nav {

background-color: var(--background-secondary);

display: flex;

flex-direction: column;

justify-content: start;

gap: 1em;

flex-wrap: wrap;

width: fit-content;

height: 100vh;

float: left;

}

There’s still a lot of work to be done, but hopefully this gives an overview of what’s possible

See style5.css