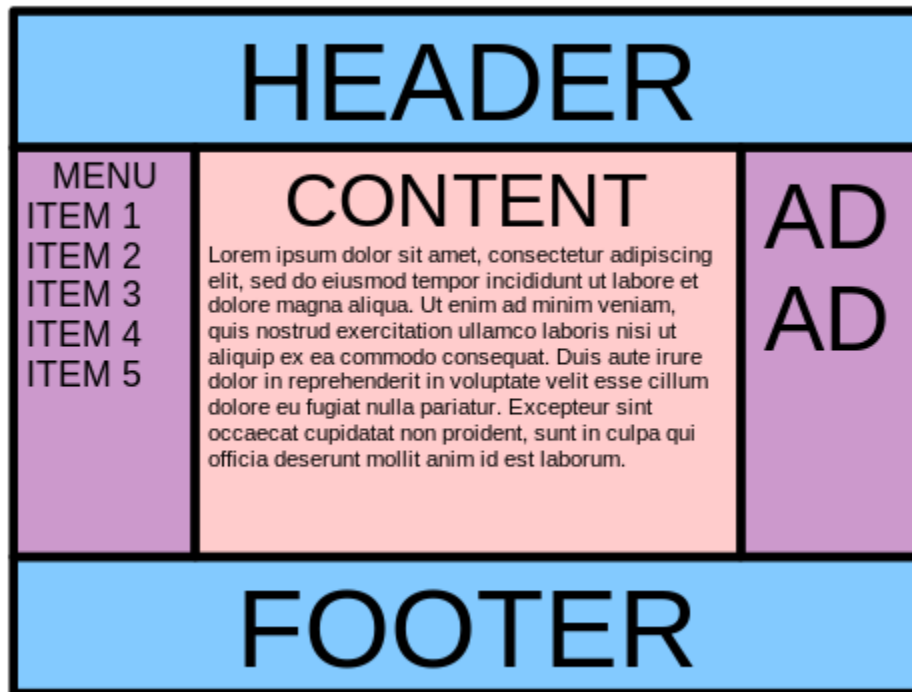## Flexbox Lab

In this second exercise, we'll create a prototype of a standard blog layout that has three columns with a header and a footer:

Follow these steps to produce your blog:

- ☐ Write all HTML in **index.html** and CSS in **css/styles.css**.
- ☐ Set all elements' box-sizing property to border-box.
- ☐ Create a **header** for the top, a **div** with an **id** of **middle** for the middle row, and **footer**. I suggest you temporarily give them distinctive background colors so that you can see the structure of your layout as you tweak the flex properties.
- ☐ Add a title in the header and a social media icon in the footer. Add padding as needed. But not margin. The background color doesn't appear behind the margin.
- ☐ Zero out the margin on body to remove the gap around your page.
- ☐ We want to distribute out these three elements across **body**, so make **body** a flex-container. Have it layout its flex-items in a column.
- ☐ To ensure that the footer always gets shoved to the bottom of the page, set the **min-height** of **body** to **100vh**. When you have little content, this will keep the footer from awkwardly rising up from the bottom of the page.
- ☐ Have **middle** be the only element that expands to fill the extra space of the flex-container. Set **flex-grow** to 1, while the header and footer remain at the default of 0.

- ☐ In **middle**, add a **div** with an **id** of **left**, a **main**, and a **div** with an **id** of **right**. Give them each distinct background colors so that you can see what's going on. Add some dummy content so the elements don't collapse.
- ☐ We want **middle** to distribute its child elements using Flexbox, so it too must be a flex-container. Set its **display** and **flex-direction** properties.
- ☐ Only **main** should expand. Set the **flex** properties accordingly.
- ☐ Add some more reasonable blog-like content to **left**, **main**, and **right**. Center what's in **left**. Box what's in **right**. Add fake posts in **main**.
- ☐ When the page shrinks below a certain size, we want the layout to be entirely vertical. We can do this with just one property tweak. Switch the **flex-direction** from **row** to **column** using a media query.
- ☐ Ensure that the layout is still aesthetically pleasing even when the media query kicks in. The contents of **left** should be centered cleanly. The boxes in **right** should expand to fill the page.
- ☐ You do not need to add a complete content, but try to complete it with your favorite contents, images, and links.