Lab 3: CSS Layouts

Description
This lab takes you through the stages of a website layout's design to its implementation in CSS and HTML. You will be asked to logically decompose an image of a website layout into divs, create an HTML document that represents those divs, and then to position the divs into their final places with CSS.

Fixed Layouts
Part 1: Identifying divs

1. Download today's lab files under Assignments > Lab 4.
2. For `site1.jpg`, `site2.jpg`, and `site3.jpg`, look at each image and identify the location of the divs which make up the layout for each site. You don't need to turn in anything for this part.

Part 2: From Image to HTML

1. Now that we've broken our layout into divs, it's time to create the corresponding HTML file.
2. Open `index.html` and code up the divs with the appropriate `id` and `class` values depicted in `site2_divs.jpg`.

Part 3: Layout styling with CSS

1. Now that we have the HTML representation of our layout, it's time to position the divs so they match the final configuration of our layout.
2. Link your css and html files together by adding the following line between the `<head></head>` tags in `index.html`:
   `<link href="style.css" rel="stylesheet" type="text/css"/>`
3. Open `style.css` and fill out the stylesheet so that your divs match the layout shown in `site2_divs_info.jpg`. Note that we're using colored divs in this lab (so you can easily tell where you're positioning them), but in a real website you'd be using a combination of images and content. We've given you the exact pixel dimensions for each of the divs, but feel free to approximate the positions (since some of them aren't clear from the screenshot).

Extra for Experts

1. We can only horizontally or vertically "stack" things using positioning if we know the exact height or width of the objects. What do we do if we don't know the dimensions of our HTML elements (if they're dynamically sized to fit their contents)?
2. Oops, it looks like the coffee cup div in the lab today is hidden behind some of the other divs (depending in which order your created the divs). There's a special CSS property called z-index that controls the order in which elements appear "on top" of each other. Apply this property to make the coffee cup div "above" the other divs in the layout.
3. While viewing your `index.html`, resize your browser. You'll notice that all of the elements remain the same size and may trigger a scrollbar to appear in your browser. Sometimes this is the behavior we want, when we have a very precise layout that needs to fit down to the last pixel. However, it may also be advantageous (and more accessible) to have a flexible layout. In `extra/extra1.css`, style the three colored columns in `extra/extra1.html` so that they resize automatically when the browser size changes. Note that all three of them should take up exactly 100% of the screen width (you do the math!).
4. Try adding another colored div inside the center one in extra1.html, and make it *vertically centered*. How did you do it? HTML5 actually introduces a new way to do this, called the *flexible box model* – look it up and play around with it. You might want to use it in the future!

Submission
Remember how to upload files? Post your submission online to `inst.eecs.berkeley.edu/~cs198-xx/lab3`, where xx is your login, and make sure it's viewable from your browser.