Submenu

Each menu item has a nested menu which is called the "submenu." This lesson will briefly explain how we are going to implement it in our code.

WE'LL COVER THE FOLLOWING

- Fixing hover behavior
- Skeleton for submenu

The submenu we will be implementing appears when we hover over a menu item. It looks as if it's extending out of the menu item, but it's actually just an overlay that stays in the same position regardless of the hovered item. The appearance of connectivity is given by the hovered menu item background matching that of the submenu.

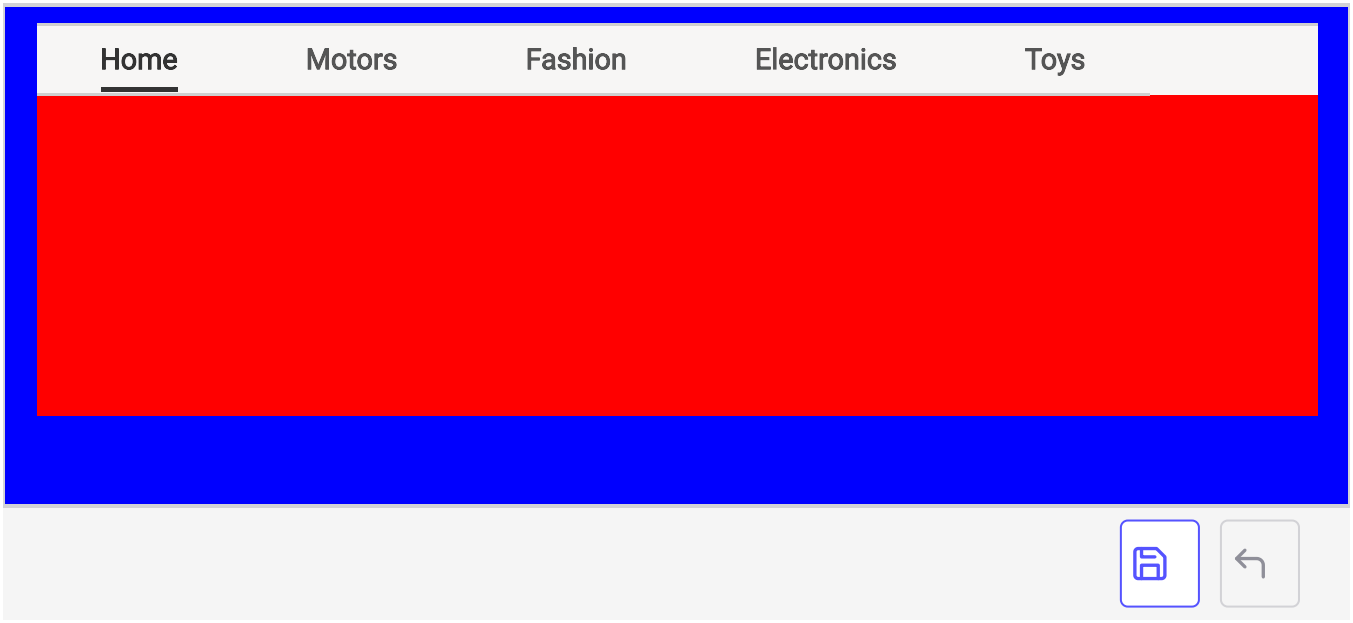
So with that, let's create this overlay to show up for now. Nothing fancy, just a `div` with some width and height.

Something that I like to do while tweaking CSS – especially positioning and sizing – is to liberally give color to things that take up space. For example, the default white background and the overlay are similar in color (along with the menu colors), so to avoid any confusion, we're just going to paint the background blue and the overlay red for now. This makes it abundantly clear the space that each element takes up.

Output

HTML

CSS (SCSS)



Fixing hover behavior

This is a good time to bound the width of our menu and center it. Adding `margin: 0 auto` often does the trick for horizontal centering and will keep it centered as you increase or decrease the number of menu items. I've put the submenu as an element within the `menu` div, so there's cohesion between elements that share similar positioning properties. If the layout requirements change in a way that we want the element to be closer to the left, I can add properties to the parent `menu` element instead of two different ones if I had separated them. But I don't want it at the same level as the menu items since they're not the same category of things, so I'll wrap that in its own `div` as well and call it `menu__main`. Of course, this means I have to change all the previous `menu__item` to `menu__main__item` (in both the HTML and CSS).

Now that the background color is blue, it helps me notice something I hadn't before. Upon hover, the first menu item shouldn't be given a left border, and the last menu item shouldn't be given a right. The border appears out of the bounds of the menu. It's much easier to catch things like that when you have this much contrast between your elements in development.

Let's fix that. In CSS, you can chain multiple pseudo-classes so that we can target the first with `:first-child:hover` and the last one with `:last-`

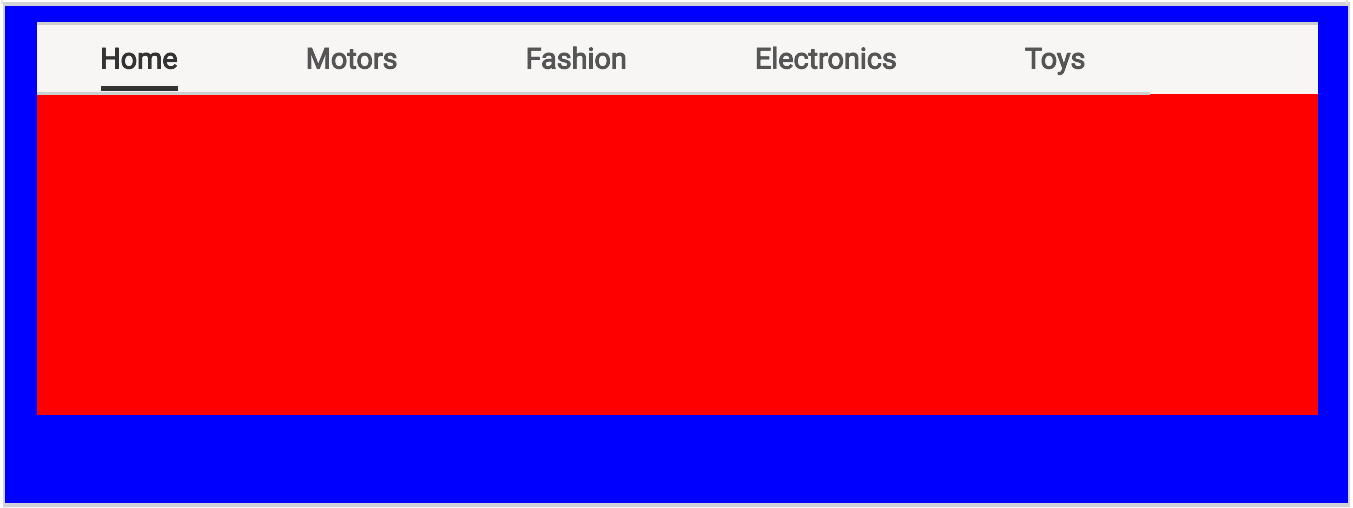
`child:hover`. With that, we can remove the overflow behavior easily by

overriding properties. In CSS, in the event of conflicting properties, the specificity of the targeting determines precedence.

Output

HTML

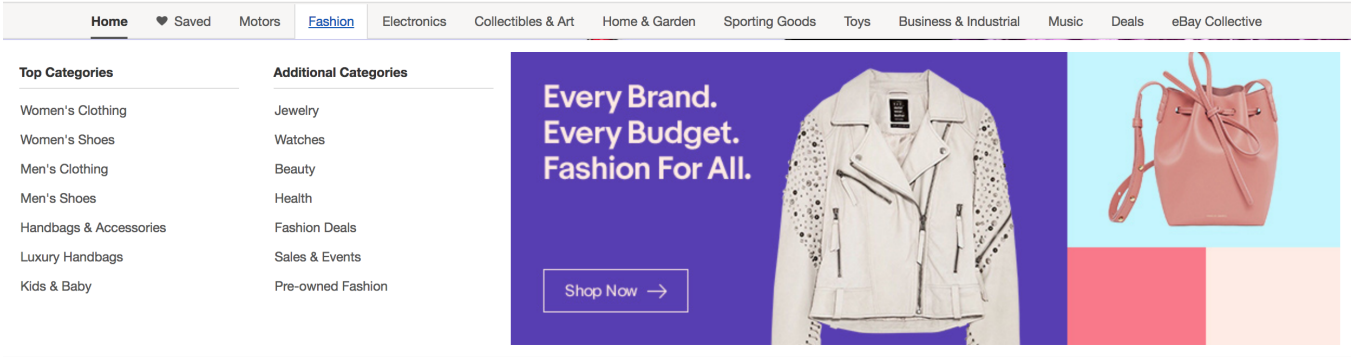
CSS (SCSS)



Skeleton for submenu

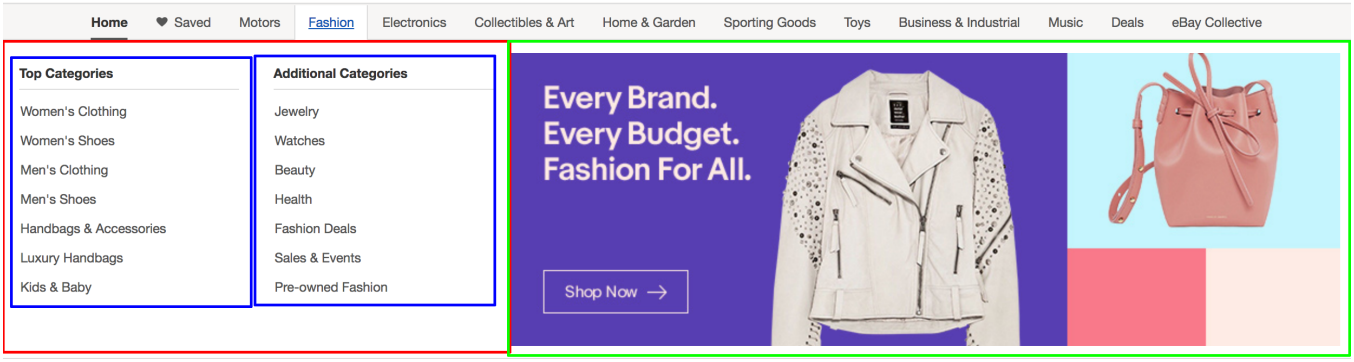
Next, let's give some stubbing for text so that we can visually work with content to add styles too.

Take another look at the original model we're trying to get to figure out how we want to do the groupings.



Q Which division of groups is best suited for the submenu?

One way to determine groupings is to look at how things are splitting the parent element in one orientation. In the image below, I've marked up how I think of these. The red and green boxes look like a firm ratio on the horizontal space they take for the parent element, maybe 40/60. And since we always have just two columns, these split the space of the red div 50/50.



Let's go ahead and take the approach we did implementing the top-level menu. In general, I almost always find myself taking these steps when creating new components:

1. Figure out groupings
2. Name classes and fill divs with text
3. Switch the colors temporarily to give distinct contrast for easier styling
4. Apply spacing styles in the order of outer to the inner (e.g., child divs will

be styled after their parent ones)

5. Switch off ugly temporary colors, use real colors, and style the rest.

Output

HTML

CSS (SCSS)

Home Motors Fashion Electronics Toys

Top Categories

- Apple
- Banana
- Pear
- Orange
- Watermelon

Additional Categories

- Circle
- Square
- Oval
- Diamond

We'll use this stock image of a dog for our image, and I've stubbed the categories with fruits and shapes. As is, things are overflowing. Take a moment now to think about what we have to do to get these in the general layout we want.

When you're ready, click through to the next lesson.