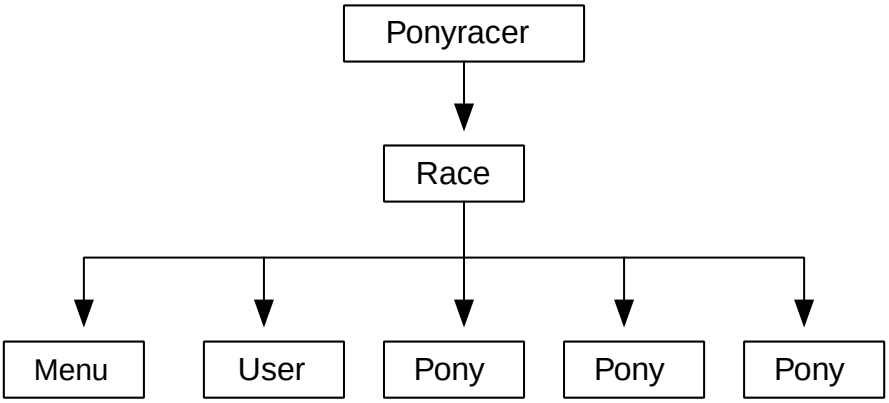
Angular is not alone in this, but it is among the first (it might actually be the first?) to really care about the integration of Web Components (the standard ones). But let's forget about this for now, as it is a more advanced topic.

Your components will be arranged in a hierarchical way, like the DOM is. A root component will have child components, each of them will also have children, etc. If you want to display a pony race (who wouldn't?), you'll have something like an app (Ponyracer), with a child view (Race), displaying a menu (Menu), the logged in user (User), and, of course, the ponies (Pony) in the races:



Writing components will be your everyday work, so let's see what it looks like. The Angular team wanted to harness another goodness of today's web development: ES6 (or ES2015, whatever you like to call it). So you can write your components in ES5 (but that's not very cool) or in ES6 (way cooler!). But that was not enough for them, they wanted to use a feature that is not a standard (yet): decorators. So they worked closely with the transpiler teams (Traceur and Babel) and the TypeScript team at Microsoft, to enable us to use decorators in our Angular apps. A few decorators are available, allowing to easily declare a component for example. I hope you already know all of that, as I just spent two chapters on these things!

For example, if we simplify, the Race component could look like this: