

Socrates

- [Tasks 2](#)
- [Tools](#)
 - [SQL Designer](#)
 - [SQL Tutor](#)
 -
 - [RubyDocs](#)
 - [Book a Counseling Session](#)
 - [Main Calendar](#)
- [Questions](#)
- [Challenges](#)
- [Labs](#)
- [Feedback](#)
 - [Feedback Dashboard](#)
 - [Give student feedback](#)
 -
 - [Give DBC Feedback](#)
- [Mark](#)
 - Mark Hagel
 - [My Profile](#)
 - [Island Foxes 2013](#)
 - [Logout](#)
 -
 - [Login Another](#)

1 Behavior Drill: Collapsible Side Menu Challenge



When designing user interfaces, there are often elements that you would be useful to include if they didn't take up so much screen real estate. It's a competitive market, and you have to position your elements wisely.

One way to get around this problem is to make elements collapsible, and then add event handlers to toggle the display of the element.

This is exactly what we're going to build in this challenge: a collapsible sidebar-style navigation area.

If you haven't already written the styles for a "sticky" sidebar-style navigation area, go and do the prerequisite challenge. You can reuse that code for this challenge.

The [source gist](#) contains a basic HTML file for a product site. To complete this challenge, you will need to edit the HTML, add some styles to the `main.css` stylesheet, and write your own JavaScript in a separate file.

You will need to brandish the powers of CSS's `position` property, and you will want to use jQuery to help you out with the showing/hiding part. Suggested readings:

- [CSS Positioning 101](#)
- [jQuery Click Event Binding](#)
- [jQuery Toggle Function](#)

Objectives

Make it Collapsible

The first step in making an element respond to user input is to ask the question "what should trigger this behavior?" In other words, what should your JavaScript "listen" for it to know what to do?

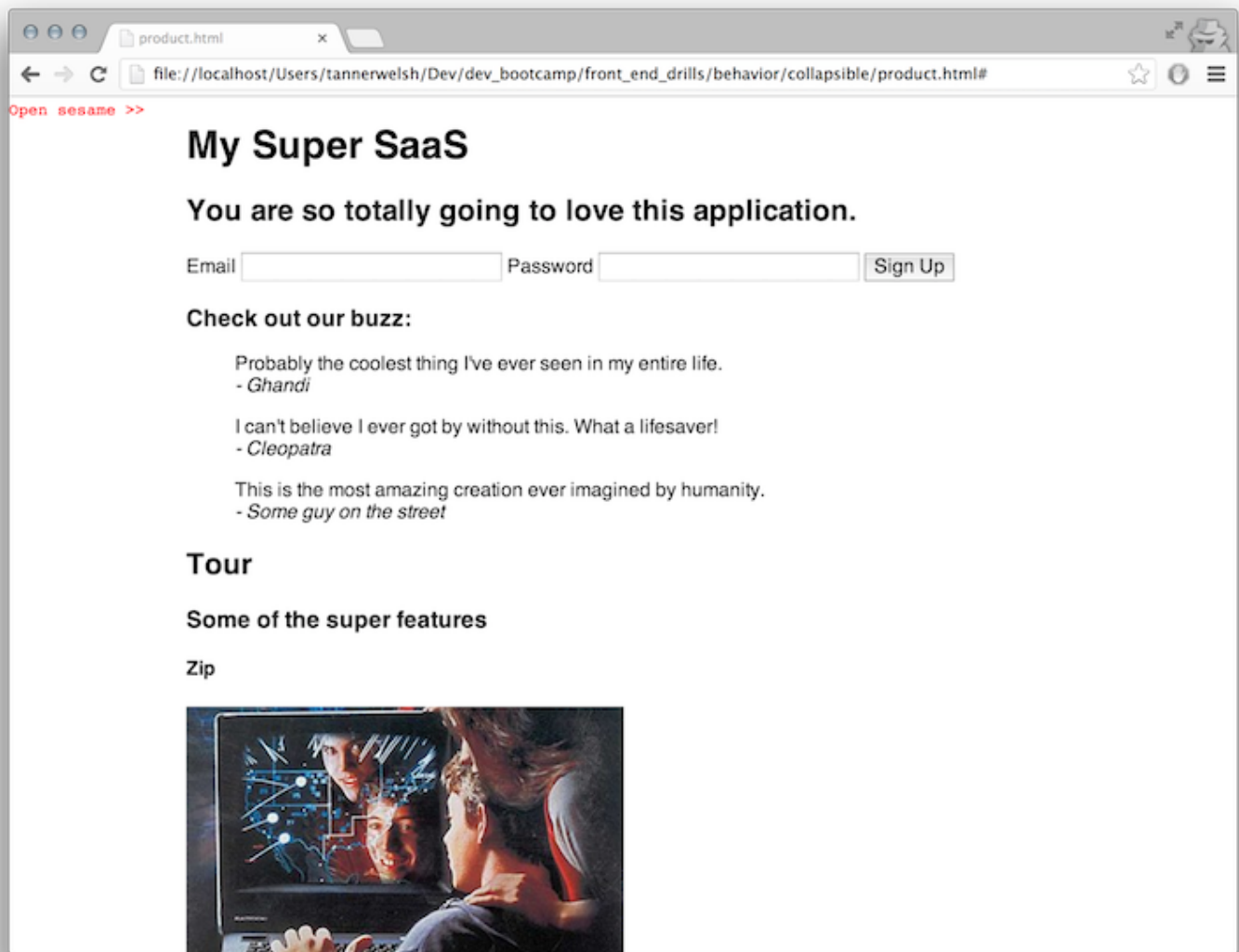
This is called *event binding* and it is a core component of designing rich client-side applications. In this challenge, we're going to stick with a tried-and-true event to trigger the behavior: the mouse click. Old, trusty, mouse click.

To make the sidebar collapsible, you will need to provide the user with two things:

1. A way to open the menu (if it is closed)
2. A way to close the menu (if it is open)

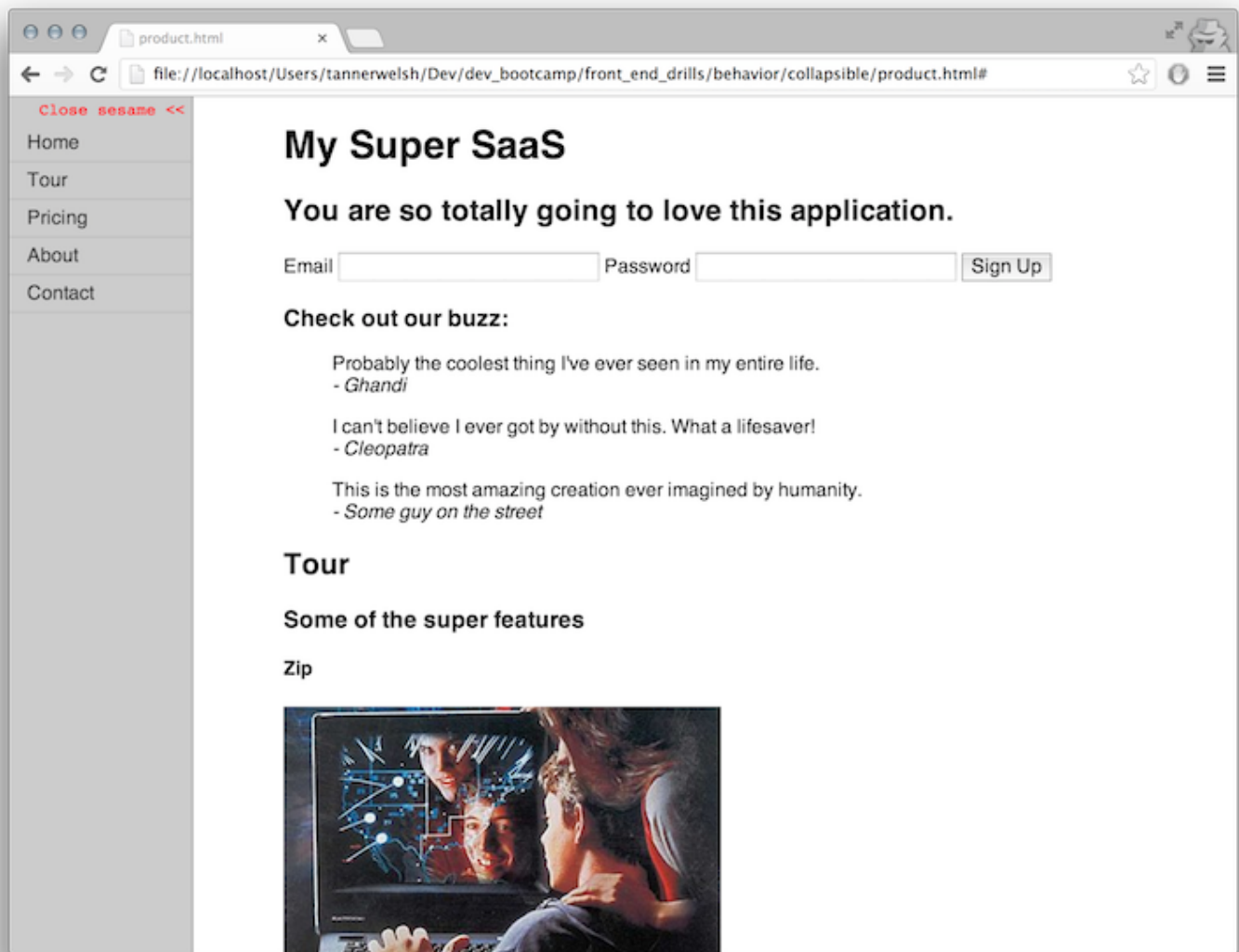
It's as simple as that. In the screenshots below, you can see how this works in action.

1. Open the page, sidebar is closed, and a link is provided to trigger the "open" action

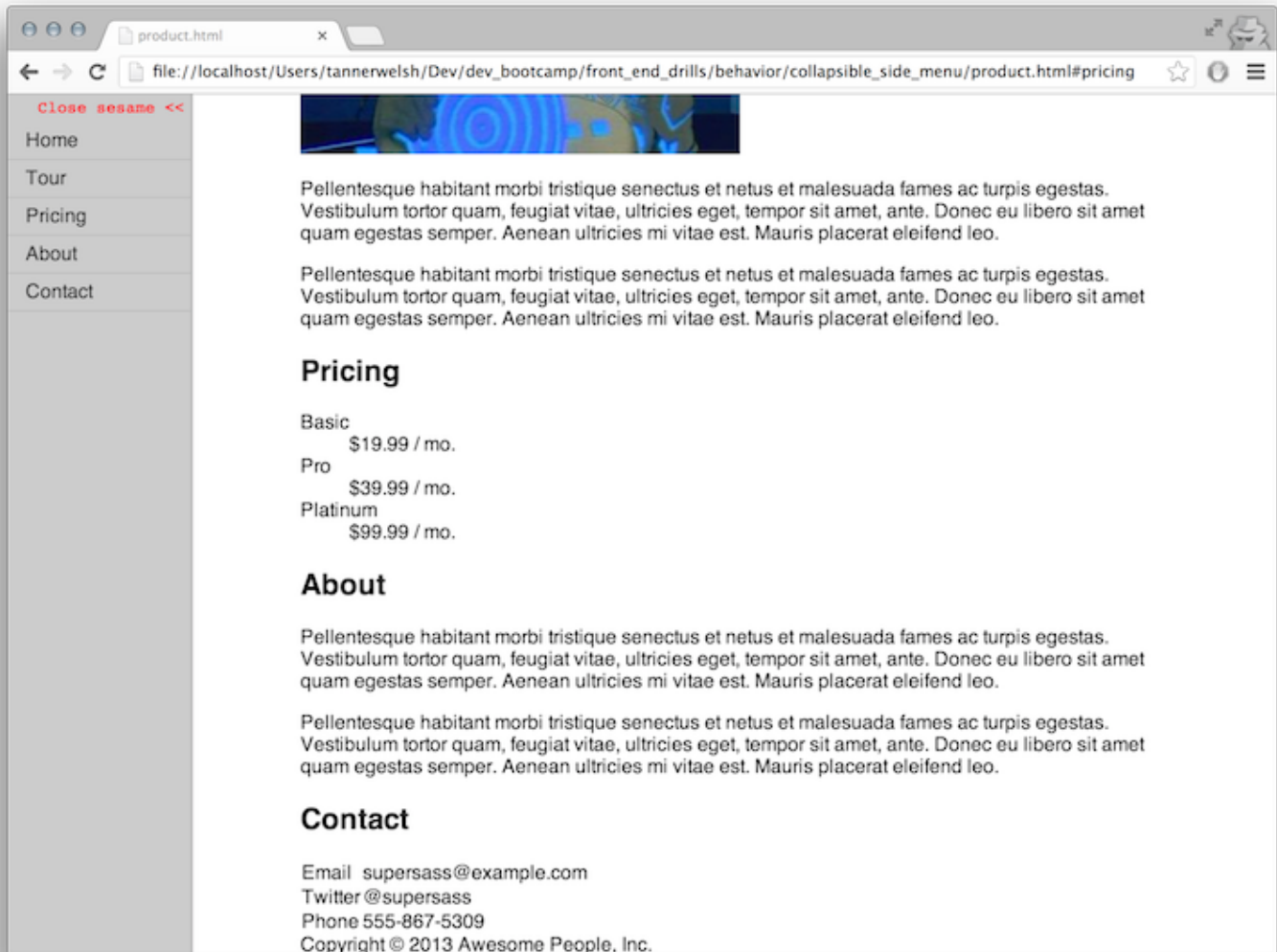


2. When the `open sesame` link is clicked, the sidebar appears. Ta-da!

Notice that the content area is still centered in the white area of the page, *not* centered on the browser window itself.



3. The sidebar is sticky and can be opened or closed from any scroll position on the page



Write whatever HTML, CSS, and JavaScript is needed to achieve the above functionality.

For extra bonus points, you can explore even further:

- Instead of an ugly link, make a nicely-styled button toggle the sidebar. Use the same button to open and close the sidebar.
- Redesign the sidebar so that when it is closed, a small section of it (~ 20px) is still showing, and when the user hovers over that section, the full sidebar is shown.
- Animate the show/hide behavior with jQuery's `animate()` function.

Comments

Please share: relevant blog posts, useful resources, thoughtful questions

You can use GitHub-flavored Markdown here.

Can't be blank

Comment

x

New Comment

You can use GitHub-flavored Markdown here.

Can't be blank

[Cancel](#)

Comment as Mark

x

Code submissions