Examining the Important Features

Medium has a pretty interesting tooltip. Let's analyze what we're building and how we can achieve it.

WE'LL COVER THE FOLLOWING

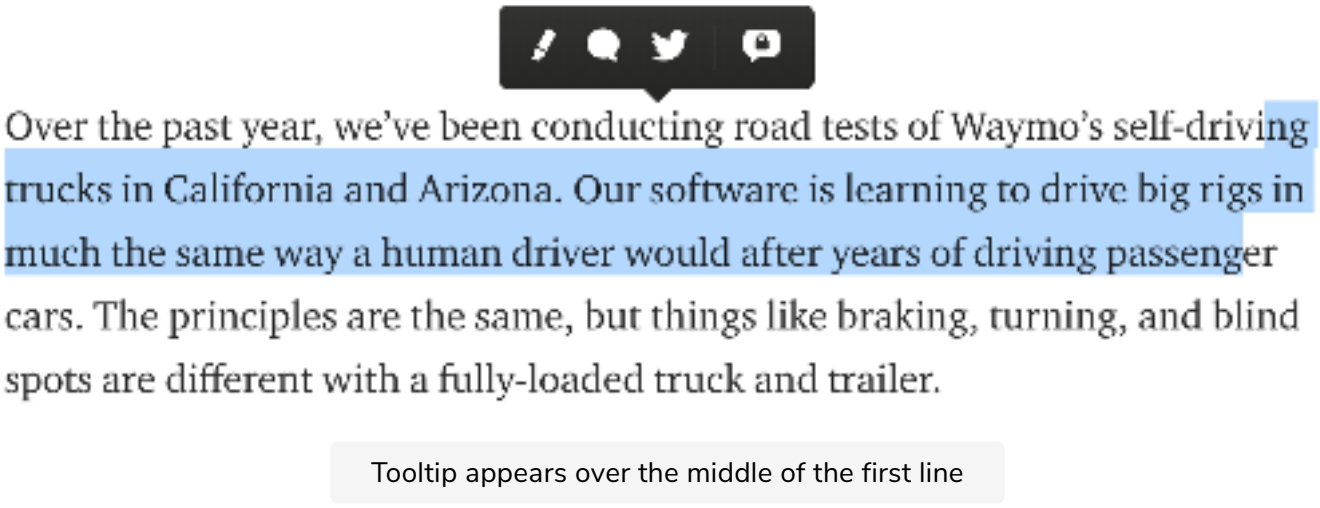
Determining scope

As always, we'll begin by scoping the work ahead. It's an imperfect process, and something will almost always be missed. It's important that as we're designing our solution, we can ask ourselves whether it'll work for x, y, and z. If you have an example you're building off of, play around with it and try to break it or see how it behaves given a variety of user interactions. To move past this stage, you shouldn't have an unanswered, "how do they do that?" You don't have to know how *it* is implemented, but you should know *a way* to implement it.

If you're stuck, being more specific about what's happening can often help.

- When text is highlighted, a tooltip shows up.
- When text is highlighted, an HTML element is displayed in the middle of the highlighted text.
- When I click down on some text, drag my mouse over some text, and release the mouse, an HTML element is added to the page positioned over the middle of the text that was dragged over.
- When I click down on text in the div of the article, drag my mouse over

some text, and release the mouse, an HTML element is added to the page centered slightly above the middle of the selected text. The exception is when the selection spans multiple lines, in which case it's positioned over the middle of the first line.



Tooltip appears over the middle of the first line

Searching for possible solutions

At this point, we should ask ourselves how can we reliably get the text? Tracking the position of the mouse when it was first clicked to where it was released and figuring out which text was selected based on that path seems tricky. As a web developer, you should liberally rely on searching documentation and online resources like Stack Overflow. Googling "getting selected text JavaScript" shows that there's a native way to do so. https://developer.mozilla.org/en-US/docs/Web/API/Selection

The next question is, how do we get the middle? Do characters in fonts have the same spacing between each one such that I can multiply the number of characters selected with some constant and divide by two?

Googling "get middle of selected text JavaScript" yields nothing useful this time around, unfortunately (or maybe fortunately, since we get a teachable moment from it). Okay, let's see. Googling "character width of fonts", it appears only "monospaced" fonts have the attribute that each character is the same width. Since these are articles with aesthetically pleasing typography, we're out of luck there.

Testing and exploring

Let's see what the `selection` api gives us. The nice thing about web

development is that trying out some code on a whim is extremely easy to do.

Open up the console, and there's your REPL (interface to evaluate expressions).

Hmm, the value doesn't seem to be quite what we selected. I wonder if it's thrown off at all by the fact that we defocus the selection when we focus on the console. We can write a simple script to test that.

```
setInterval(() => {
  console.log(document.getSelection());
}, 1000);
```

Putting this in an HTML will have the current selection printed to the console every second. After running the test, I see that the act of running the earlier test didn't interfere with the result – `getSelection` gives us all the text in the element that the selection took place. Of course, the documentation tells us the same thing, but you shouldn't hesitate to run quick and dirty tests to confirm or deny assumptions about some JavaScript code you're unfamiliar with!

So the library gives us the two HTML nodes (node has roughly the same meaning as an element) that the selection begins and ends at, and the character offset into each node. I'm not convinced that helps us, since again, we don't know the exact spacing of each character selected.

We'll continue exploring this problem in the next lesson.