

Bug 2: Tooltip Displays on Clicking

In this lesson, we will try to fix the behavior of our tooltip and making sure it doesn't display itself by merely clicking on the text.

WE'LL COVER THE FOLLOWING ^

- Clicking displays the Tooltip

Clicking displays the Tooltip

This is an artifact from when we added it to the click event for quick testing purposes. There's an event JavaScript allows to be listened for – `onselect`. However, [the docs](#) show that it only applies to text in `textarea` or `input` elements. We're going to have to rely on `onmouseup` and `onmousedown` then. How about if the user triggered `onmousedown` and `onmouseup` on an `article` div, and it wasn't the same position. That must imply some text was selected.

Output

JavaScript


HTML

CSS (SCSS)

we'll begin by scoping the
what what



This works, but play around with it, can you see if it introduces another bug?

 Show Hint

Since that doesn't work, we can just check whether there is a selection with every mouseup.

Output

JavaScript

HTML

CSS (SCSS)

we'll begin by scoping the
what what



I had expected `getSelection()` to give me null when I clicked on something, but instead, it just gives a `Selection` of a different type. Since I only care about the case where the text is highlighted, I'm filtering for only `Range` types.

This introduces yet another bug! If we single click on a selected text, the



selection is in range, but the text gets de-highlighted on the browser immediately after, but the tooltip stays.

When you've tried different solutions, and you're feeling stuck, be specific about what you need and go back to search engines and documentation. It seems the only way to capture the case where clicking on selected text de-highlights is if JavaScript lets us – as in if there's any way to listen for selection changes because using click events have shown not to be robust. Earlier, when we had dismissed `onselect`, I assumed there wasn't API for text selection changes. Digging further into the `Selection` API docs, however, I found an `event` called `selectionchange` that I can listen to. (A side effect of JavaScript iterating so rapidly is the lack of attention to address minor pain points, like this non-intuitive API name clash!)

Output
JavaScript
HTML
CSS (SCSS)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum eget sem pellentesque, ultricies dolor eget, malesuada mauris. Fusce lorem felis, eget tincidunt purus et, sodales luctus lectus. Praesent a congue eros. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris vitae malesuada urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut in mauris justo. Curabitur rutrum, est nec viverra rutrum, dolor est volutpat dolor, vitae auctor urna nulla nec nibh. Nunc convallis libero luctus purus auctor, sed aliquet lectus fermentum. Nunc a massa ac sapien viverra consequat.

Aliquam arcu mi, vehicula a sodales ut, dictum sit amet ex. Sed nec enim sed elit dapibus porta at eu felis. Quisque faucibus vel sem in bibendum. Nulla dictum, nunc in mattis tincidunt, nisi sapien blandit metus, ut bibendum tortor metus non nunc. Duis eu ultricies risus, quis mollis magna. Duis efficitur congue ante ut imperdiet. Vestibulum at sodales eros. Etiam id dictum ligula.





We add a bit more text for ease of testing.

The bug this time is much more apparent. Selection changes before you've finalized the selection, so the tooltip shows prematurely. Perhaps even less obvious than the name clash of `onselect` and `selectionchange`, there's the

obvious that the name chain of `onselectstart` and `selectionchange`, there's the existence of `onselectstart` event but not `onselectend`!

When you encounter some API that you feel like *should* exist – some tool that's missing in JavaScript – consider what it would take to implement it yourself. If the implementation is something other people might need, maybe clean it up, package it, and release it as an open-source project! The frontend world relies on open source contributions, and it's a great way to establish your expertise.

Anyway, `onselectend` to me feels like it *should* exist. Maybe the reason it doesn't is that the developers considered the `change` to be the terminal stage of selection, but in my interpretation and for the purposes of this component, selection doesn't end until your mouse is released. So `selectionchange` tells the tooltip what to do, but it doesn't happen until `mouseup`.


Output
JavaScript
HTML
CSS (SCSS)
<div><p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum eget sem pellentesque, ultricies dolor egetas, malesuada mauris. Fusce lorem felis, egetas tincidunt purus et, sodales luctus lectus. Praesent a congue eros. Interdum et malesuada fames ac ante ipsum primis in faucibus. Mauris vitae malesuada urna. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut in mauris justo. Curabitur rutrum, est nec viverra rutrum, dolor est volutpat dolor, vitae auctor urna nulla nec nibh. Nunc convallis libero luctus purus auctor, sed aliquet lectus fermentum. Nunc a massa ac sapien viverra consequat.</p><p>Aliquam arcu mi, vehicula a sodales ut, dictum sit amet ex. Sed nec enim sed elit dapibus porta at eu felis. Quisque faucibus vel sem in bibendum. Nulla dictum, nunc in mattis tincidunt, nisi sapien blandit metus, ut bibendum tortor metus non nunc. Duis eu ultricies risus, quis mollis magna. Duis efficitur congue ante ut imperdiet. Vestibulum at sodales eros. Etiam id dictum ligula.</p></div> <div></div>

The way I encode that logic is to use a flag. The tooltip can be in a queued state, which is set by the `selectionchange`, or not in a queued state, which is

the default and set by `onmouseup`. Since `selectionchange` always happens

before `onmouseup`, tooltips only show up when `selectionchange` says it should, and the user has “finished selecting”.

Can you see why we’ve commented out line 76?

 Show Hint

In the next lesson, we will take a look at our third bug.