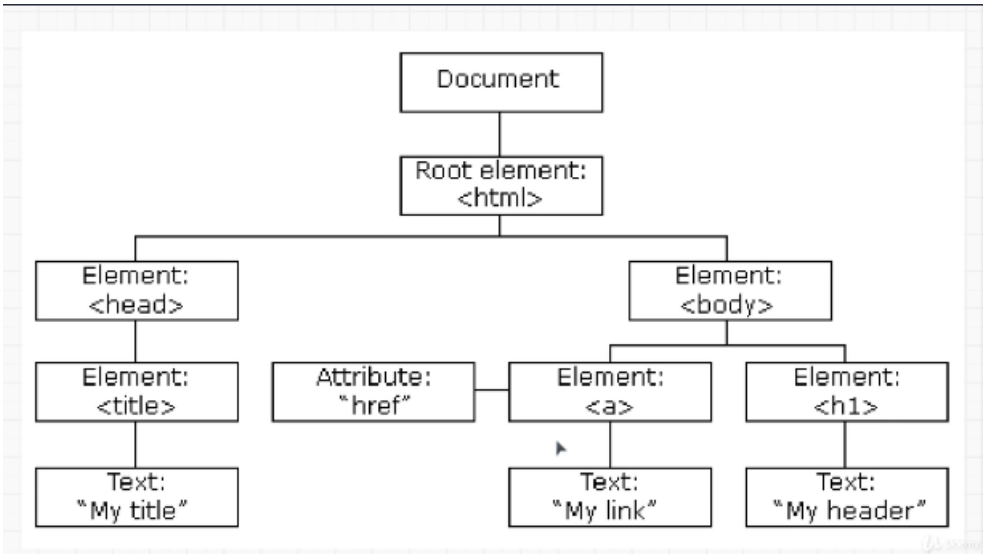
DOM & Performance

The Complete Web Developer in 2018

The Complete Web Developer in 2018
Zero to Mastery
Andrei Neagoie
Lecture Notes by Stephanie

Accessing DOM and Performance

Every time innerHTML set - HTML is parsed, DOM is constructed and inserted into the document.



Recreates this tree, adds it to the Web browser, and then has everything in here show up on the Web page.

This takes time!!

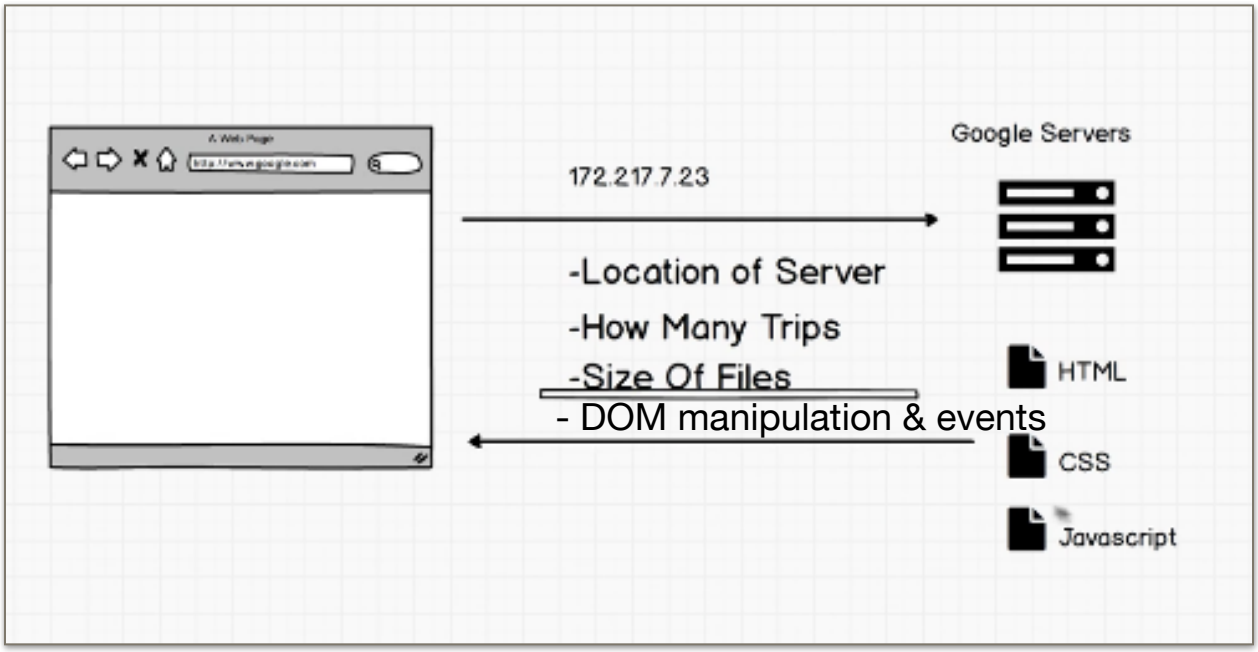
For example, if element has the tag ‘a', and there's thousands of these in the DOM tree, when you call this .innerHTML, it's going to cause the Parser to re-parse everything all over again.

This could break references to click events and cause other chaos. In reality, all you want to do is attach a single new element to the end.

innerHTML is also susceptible to attacks called cross-site scripting which is a security problem.

For example, look at the Udemy website: a lot more going on than just text and style.

They access the DOM and change things quite a lot but this is a huge performance problem.



Make performant and fast websites by <u>minimizing the amount of backend requests we make.</u>

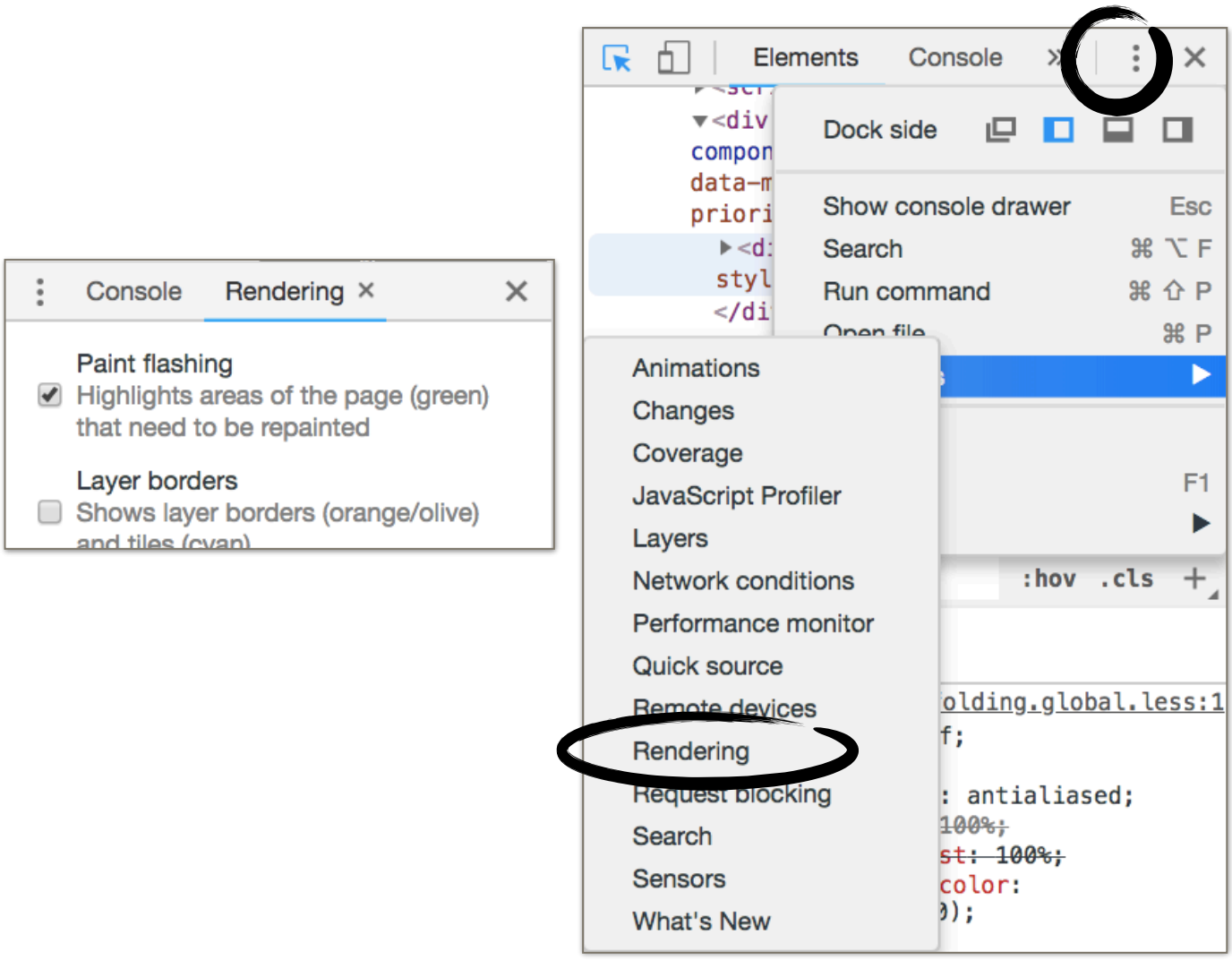
Performance factors to consider:

- location of servers
- how many trips to grab all the files
- size of the files
- minimize amount of DOM manipulation & events

If we change one drop down, we don't want the whole page to re-render.

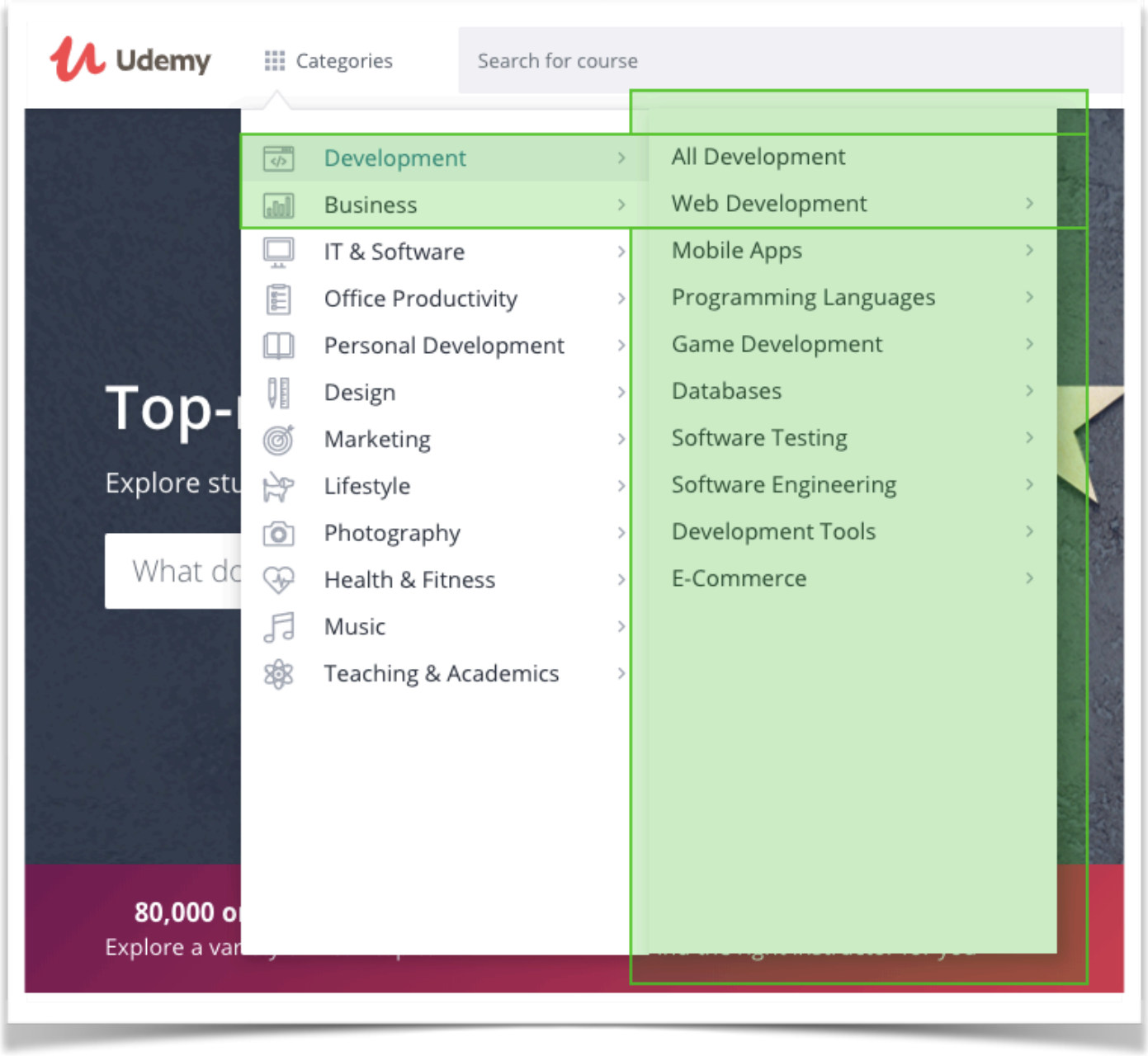
We want an interactive website without overworking the Web browser by constantly changing things to the document object.

Chrome > Developer tools > … > Rendering > Paint Flashing



Let's use this to see how websites are re-rendered as we scroll or change drop-downs, hover over items, etc.

Good pages: Udemy does a good job of making sure that when I do this only this section that I'm actually
touching gets...gets painted.



Bad: Some websites you'll notice when you scroll… it re-renders the whole thing instead of just the bar as you can see on Udemy. The React library will help us with this.