

Lesson

Tuesday

Introduction to Programming

(/introduction-to-programming)

/ Git, HTML and CSS (/introduction-to-programming/git-html-and-css)

/ Debugging HTML and CSS

Text

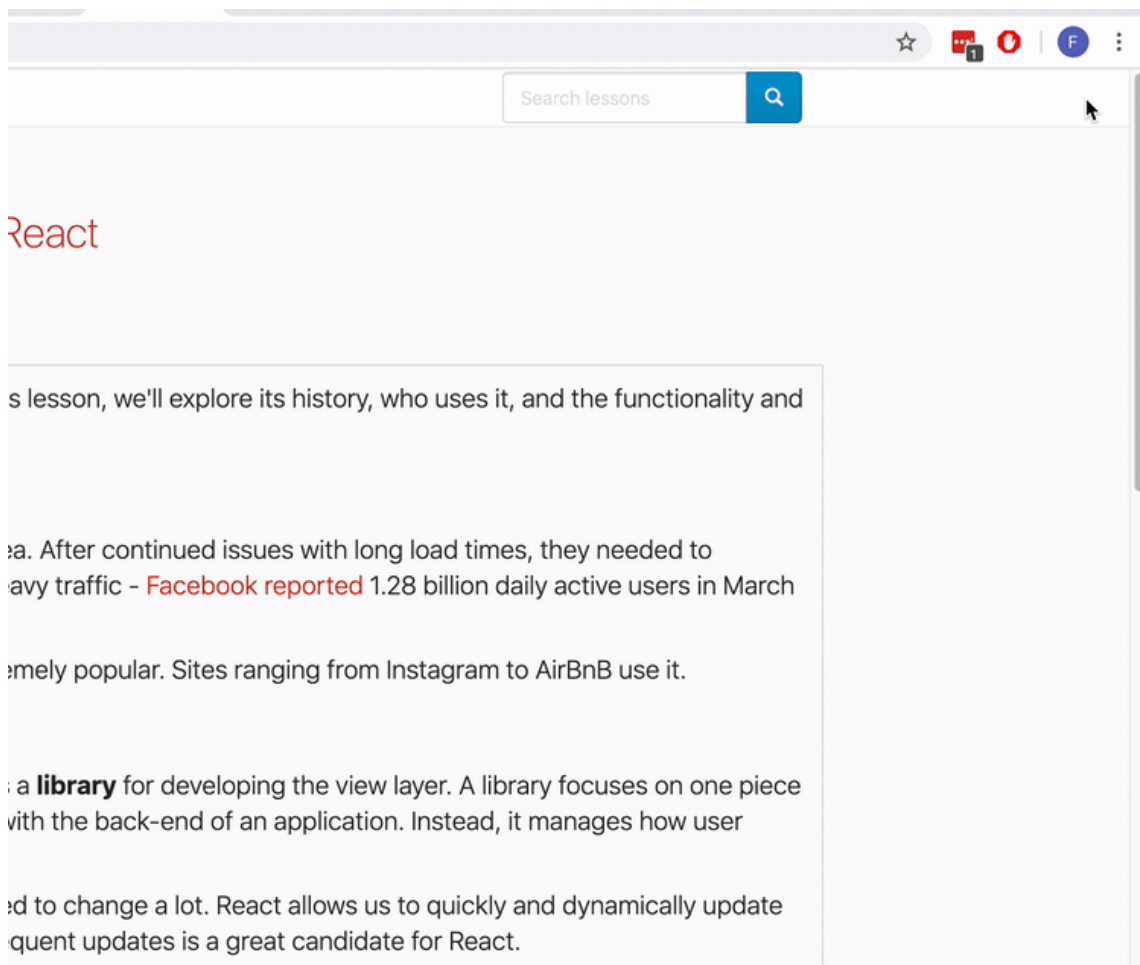
Cheat sheet

Learning to debug is an essential skill to pick up early on. There are many debugging techniques. Here are a few tools that are very helpful for debugging HTML & CSS.

We use Google Chrome (<https://www.google.com/chrome/browser/>) at Epicodus so we will focus on Chrome's Developer Tools, also known as **DevTools**. Other browsers like Firefox also have developer tools. Feel free to look into these on your own if you are interested.

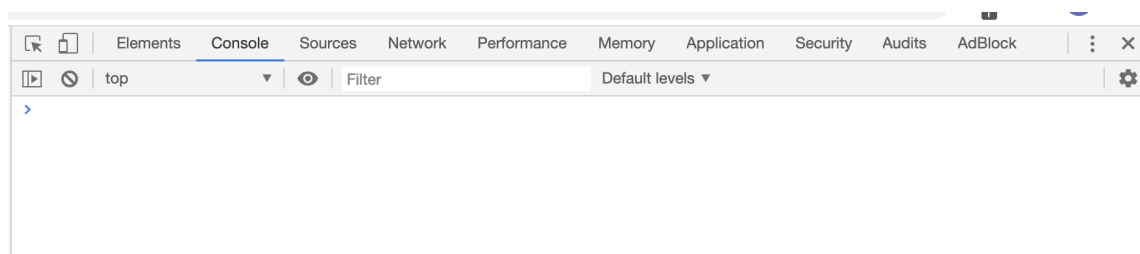
JavaScript Console

In Chrome, click on the menu button. It's on the right side of the address bar and looks like ☰ in older versions of Chrome or three vertical dots in newer versions. Click *More Tools* and then click *Developer Tools*.



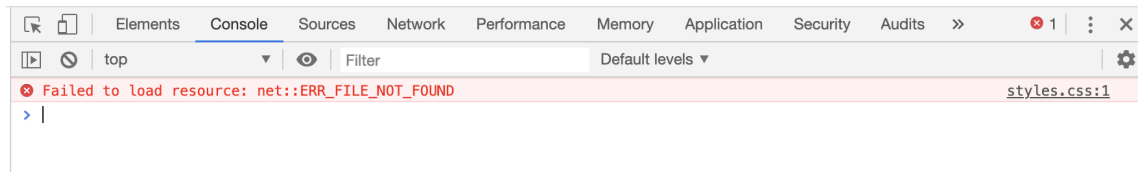
As a shortcut, you can also press `Cmd + Option + J` (`Windows + Alt + J` on Epicodus keyboards) to open DevTools.

Next, let's take a look at the DevTools window.



As the image above shows, there are a number of different tabs in the DevTools window. We will regularly use the Elements and Console tabs throughout the program. While the other tabs (such as Sources and Network) also contain useful information and can be helpful for debugging, we won't use them as much.

The Console tab is selected in the image above. This is the JavaScript console where all errors in our web page are logged. We aren't using JavaScript yet but the console will also show us descriptive errors that can be helpful now. For instance, if your CSS file isn't properly linked, you'll see an error message that looks like this one:



The error reads `Failed to load resource: net::ERR_FILE_NOT_FOUND`. Then, on the right side of the error message, it reads `styles.css:1`.

This is a very common error, especially for beginners. It simply states that a file can't be found. The name of the file is on the right side of the error message: `styles.css`. There is also a line number to the right of the file name (`1` in this case). Sometimes this is helpful and correctly points to a line number in a file where an error can be found. Sometimes, it's not accurate because of the way Chrome compiles JavaScript. In this case, Chrome can't find a `styles.css` file so the line number isn't relevant.

The error above can mean one of several things:

- The file doesn't exist.
- The file exists but there's a typo in either the file name or in the code referencing the file (for instance, if we'd accidentally named the file `style.css` instead of `styles.css`).
- The file exists but it's in a different directory than the one specified.

You will likely see this error many times — not just as a student but even as an experienced developer. The exact same error will come up with other file types, too, not just CSS files. You can follow the

Finally, below the elements pane, there is another pane in the bottom right corner that show us exactly which CSS styles apply. These are applied from highest specificity to lowest. In the image above, the most specific CSS rule is `u1.breadcrumb`, which has a property of `padding-left: 0;`.

The least specific rule is `u1, o1`. Note that the `margin-bottom` property here is crossed out. This is because the rule has been overridden by the `margin-bottom` property in the more specific CSS rule just above `u1, o1` called `.breadcrumb`.

As you can imagine, this is a great tool to debug CSS styles. If a style isn't showing up, it's possible that another rule is overriding it. The Styles pane will show this. It's also possible that a CSS property isn't being applied even though you think it is. The Styles pane is a great place to find out.

Editing CSS in the Browser

Here's another really cool thing we can do — we can actively edit our CSS in the browser without changing it in our code. This allows us to experiment with changing styles. Check out the GIF below.

Tracks Next

Lesson Weekend

Introduction to Programming

/ Getting Started at Epicodus / Learn How to Program

Text

Hi! Welcome to *LearnHowToProgram.com*. This site is a step-by-step guide to take you from zero experience to web programmer in a few short months. Learn How to Program contains both lessons and practice exercises. This content is the curriculum for *Epicodus*, a school for people who want to change careers and learn how to code. You are also welcome to go through the site on your own.

Before we get started, let's discuss our general philosophy at Epicodus. If you talk to an experienced developer, they'll likely agree that the more you learn about programming, the more you realize just how little you know. It's like pointing a telescope out at the universe. There are more and more stars and galaxies and solar systems the further you go. The same is true with coding.

Being a developer is not about learning a fixed set of skills that you can apply for the rest of your career. A tool you learn today may be replaced by a tool you learn a few years from now. Even if it's not replaced, it will likely be updated and modified, perhaps to the point where it no longer looks like the tool you use today.

That understanding fundamentally shapes how we structure Epicodus. We believe that the languages, tools, and approaches you'll learn here are much less important than the general skills of **solving problems**. Successful programmers embrace the limitations of their knowledge and get good at figuring out what they don't know. They develop a mindset in which *not knowing* the answer isn't a source of anxiety, but rather an opportunity to learn and explore.

Now let's explore how this site works. Each course here lines up with the different courses Epicodus offers. Our "Introduction To Programming" course is designed to take you from zero experience to being able to create the content of basic web pages, style those pages so that they look nice, and add basic interactivity. This course also covers how to use the common tools of the programming trade including the command line and source control.

Elements Console Sources Network Performance Memory

```
<!doctype html>
<html>
  <head>
    <title>Introduction to Programming</title>
    <meta charset="utf-8">
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col-xs-12">
          <div class="label label-info">Lesson</div>
          <div class="label label-primary">Weekend</div>
          <div class="breadcrumb">
            <li>Introduction to Programming</li>
            <li>Getting Started at Epicodus</li>
            <li>Learn How to Program</li>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

HTML body div.container div.row div.col-xs-12 ul.breadcrumb li a

Filter

element.style {

.breadcrumb>li, .breadcrumb>li>a {

font-size: 3rem;

body, h1, h2, h3, h4, h5, h6, div, p, a, input, button {

font-family: -apple-system, BlinkMacSystemFont, Helvetica Neue, open sans, sans-serif; font-weight: normal;

a {

color: #d9230f;

text-decoration: none;

a {

background-color: transparent;

We've already highlighted the element that holds the tile "Introduction to Programming / Getting Started at Epicodus / Learn How to Program." Let's say we want to see how this heading looks with a different color.

We can click the checkbox to the left of the property in the Styles pane. When the checkbox is clicked, the property is applied. When it's unclicked, the property is removed.

We can also update a property. In this case, we click on the color and change it to green.

We could even add a new property if we like — just start a new line in the CSS rule and add the property! (This isn't shown in the GIF above.)

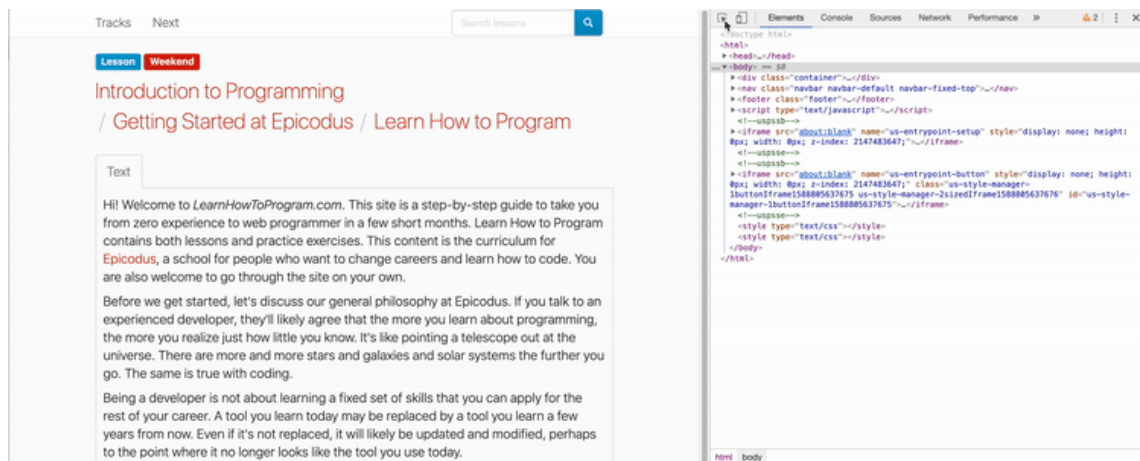
These changes aren't permanent — they are just revisions to the DOM we see. If we refresh the page, we'll see any changes we make are undone.

We can do this to *any* webpage. Pick out a site you really like (or just do this lesson) and try changing some rules on your own. It's a great way to experiment!

Inspecting Elements

We can also inspect individual elements on a page. This trick is especially useful for debugging HTML and CSS. Let's take a look at the DevTools menu again. Start by clicking on the Elements tab (if you're not already on it).

On the far left, we'll see a square with an arrow icon on it. The icon will generally be gray, but we can click on it to activate the inspect element tool. (It will turn blue when it's activated.) The GIF below shows the inspect element tool in action.



Once we activate the tool, we can hover over any element on the page to see the CSS rules that apply to it. If you look at the graphic above, you'll see how the information changes as we hover over different elements. We see the new CSS rules on the left and we see the specific element on the right (in the Elements panel).

As we can see, we can target specific elements either by clicking them in the Elements pane *or* by clicking the inspect tool and then clicking on them in the DOM.

[Previous \(/introduction-to-programming/git-html-and-css/css-styling-text-and-best-practices\)](#)

[Next \(/introduction-to-programming/git-html-and-css/practice-styling-text\)](#)

Lesson 25 of 64

Last updated March 24, 2023

[disable dark mode](#)



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.