Set up your IDE and Development Environment

The IDE's of Programming

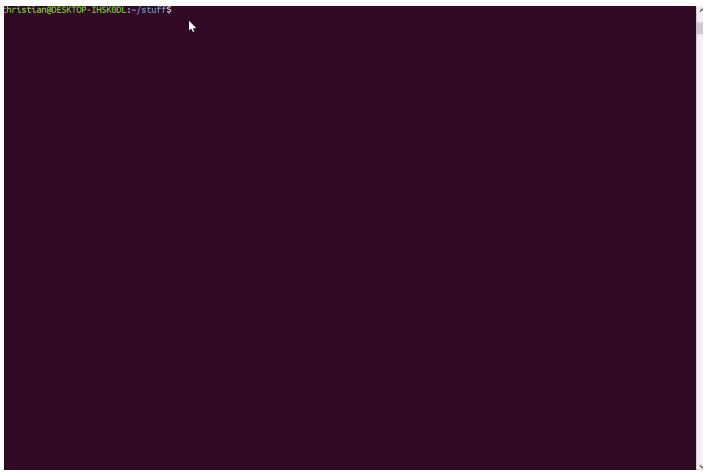
Christian Grewell Follow
Aug 13, 2018 · 4 min read

Now that we understand a bit about how to use our computer's terminal, let's get into the job of a programmer. Programming.

Before we jump in and start writing code, I want to say a few things about the tools that are used by programmers today. One in particular are quite important: the integrated development environment (or IDE).

But First...The Text Editor

Integrated development environment sounds pretty hi-tech. But the truth is it's entirely possible to write code in notepad.exe running on Windows 3.0, not efficient, but still totally possible. In fact, you don't even need a text editor to code, you can use consoles (such as the one that is included in your internet browser).



Here I am using VIM, which comes pre-installed on everything including toasters. For a take on why VIM is cool, check out this post: https://medium.com/@fay_jai/what-is-vim-and-why-use-vim-54c67ce3c18e

In our lab, we'll write a bit of code using VIM, a text-based text editor, just to familiarize you with using a text editor that has no mouse input 😱 . If you want to get a head start, download this cheat sheet: https://vim.rtorr.com/

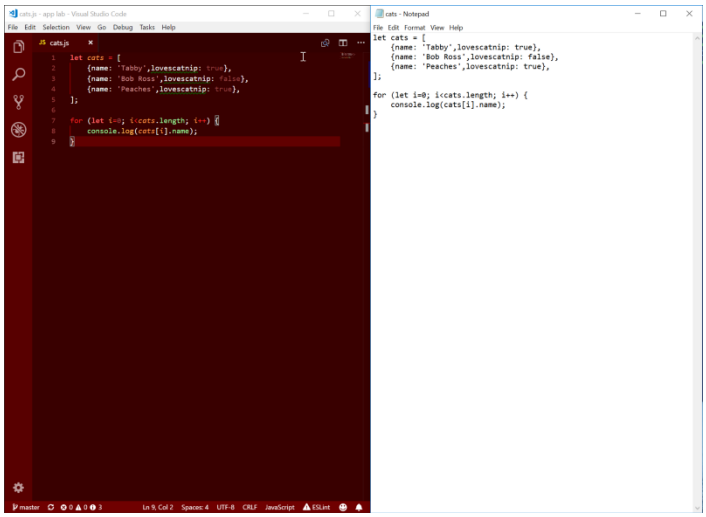
The Integrated Development Environment

For our programming labs, I'm going to be using VSCode, available for all operating systems here: https://code.visualstudio.com/.

VSCode is not a text editor (it can edit text). It's an IDE. You can think of an IDE as a performing many of the functions of a full-fledged word processor, it helps you write code by reducing keystrokes (intelligently

formatting code), suggesting function syntax, color-coding and a ton of other handy features.

On of the big benefits of an IDE is that many have a built-in code debugging environment (e.g., it can catch errors in your code before the compiling stage).

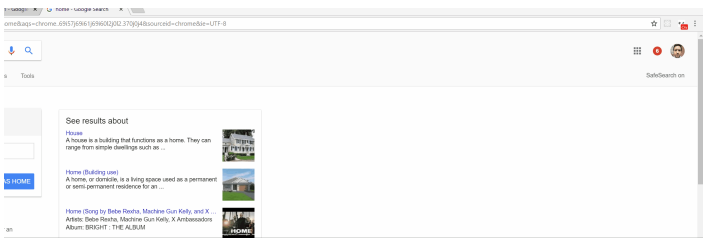


Can you guess which tool is more useful?

The Runtime Environment

The runtime environment is an integral part of the development environment. All of that text you type actually needs to be translated from something humans can read (higher order language) to something that machines can read.

Because we're going to be programming in the Javascript language (well, technically ECMAScript), we need a way to run the code that we write. Historically, Javascript was used for something called 'client-side' scripting, or small programs that ran on a client computer, as opposed to a server computer. Every modern web browser has a Javascript runtime environment within it.



You too can open your Develop Tools in Chrome, choose the Console and start programming!

The NodeJS Development Environment

For this class, we're going to use NodeJS, a Javascript runtime environment that runs outside the browser. This is good, because it means you can use it to build programs beyond just ones that run in the browser (although they'll likely work there too).

Here are two videos I recommend watching to learn more about the environment!

https://www.youtube.com/watch?v=pU9Q6oiQNd0

https://www.youtube.com/watch?v=RF5_MPSNAtU

Set up Your Development Environment

1. Download and install VSCode for your operation system (https://code.visualstudio.com/)
2. Install NodeJS for your operating system (see instructions below)

Installing Nodejs on Windows:

In order to get this version, we just have to use the `apt` package manager. We should refresh our local package index first, and then install from the repositories. Open your Ubuntu Terminal and run:

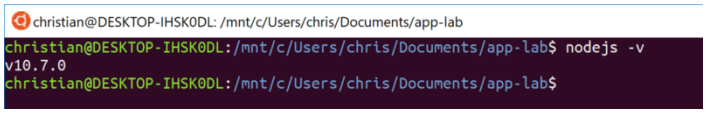
```
$sudo apt-get update
```

```
$sudo apt-get install nodejs
```

```
$sudo apt-get install npm
```

Next, make sure it's working by checking the version of nodejs:

```
$nodejs -v
```



Mac:

1. You need **XCode.** Apple's XCode development software is used to build Mac and iOS apps, but it also includes the tools you need to compile software for use on your Mac. XCode is free and you can find it in the Apple App Store.
2. **Homebrew**. Homebrew is a package manager for the Mac—it makes installing most open source sofware (like Node) as simple as writing `brew install node`. You can learn more about Homebrew at the Homebrew website. To install Homebrew just open Terminal and type: `ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"`. You'll see messages in the Terminal explaining what you need to do to complete the installation process.
3. **Open the Terminal app** and type `brew install node`.
4. **Sit back and wait.** Homebrew downloads some files and installs them. And that's it.

Next, make sure it's working by checking the version of nodejs:

```
$nodejs -v
```