

# VS Code

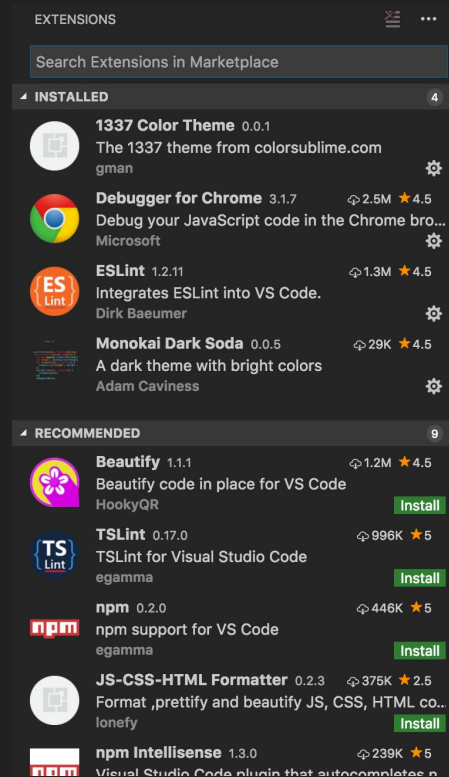


# Qué es?

VS Code o Visual Studio Code es un IDE (Integrated Development Environment) muy popular.

Cuenta con muchas extensiones que permiten y facilitan programar en casi cualquier lenguaje de programación.

Estas te sirven para correr código, lintear tu código, completar código, visualizar bases de datos, personalizar los colores de la interfaz y mucho más.

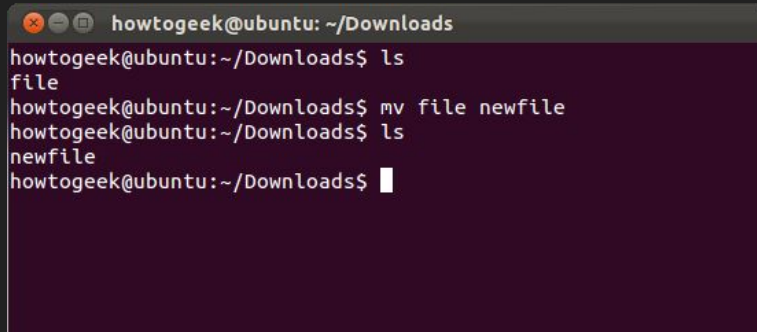


# Terminales

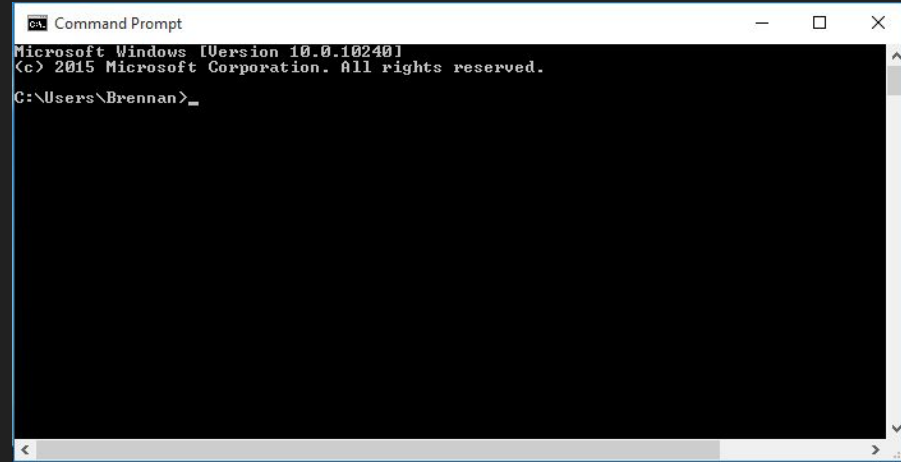
Command Prompt/ Terminal/ CMD, estos programas son “la verdadera computadora”.

En estos el usuario puede moverse por casi toda la computadora y hasta acceder a cosas que normalmente no son visibles.

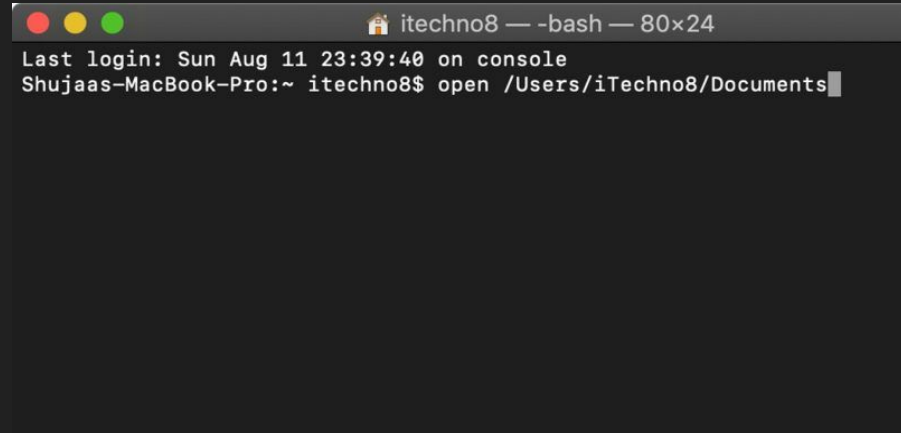
El único precio es la interfaz.



```
howtogeek@ubuntu: ~/Downloads
howtogeek@ubuntu:~/Downloads$ ls
file
howtogeek@ubuntu:~/Downloads$ mv file newfile
howtogeek@ubuntu:~/Downloads$ ls
newfile
howtogeek@ubuntu:~/Downloads$
```



```
Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Brennan>
```

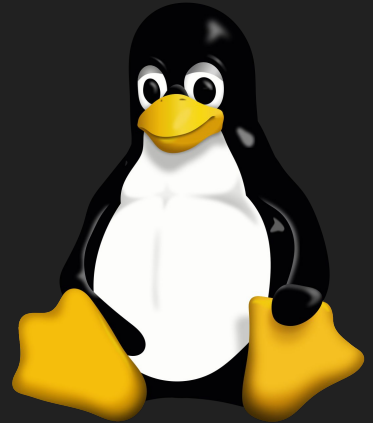


```
itechno8 — -bash — 80x24
Last login: Sun Aug 11 23:39:40 on console
Shujaas-MacBook-Pro:~ itechno8$ open /Users/iTechno8/Documents
```

# Ubuntu

Ubuntu es una distribución (o distro) de Linux.

Linux es “el sistema operativo más sencillo para programar”. Tanto los distros de Linux como MacOS son base Unix, por lo que estos usan la misma terminal (bash), mientras que Windows está hecho en DOS, que cuenta con CMD y Windows Powershell.

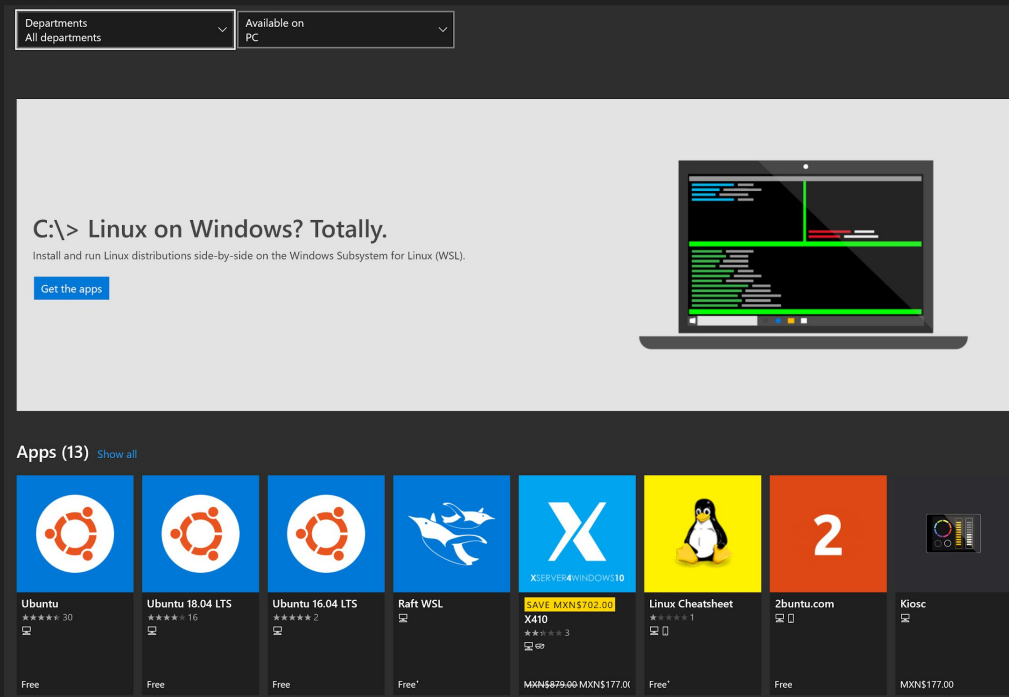


# Facilitemos las cosas

Como ya dije antes, Linux es “el mejor sistema para programar”, pero... Ninguna computadora viene con un distro de Linux, solo vienen con Windows o MacOS.

Así que debemos instalarlo por separado.

Esto nos ayuda a poder usar la terminal de bash, cosa con las que Mac ya cuenta.



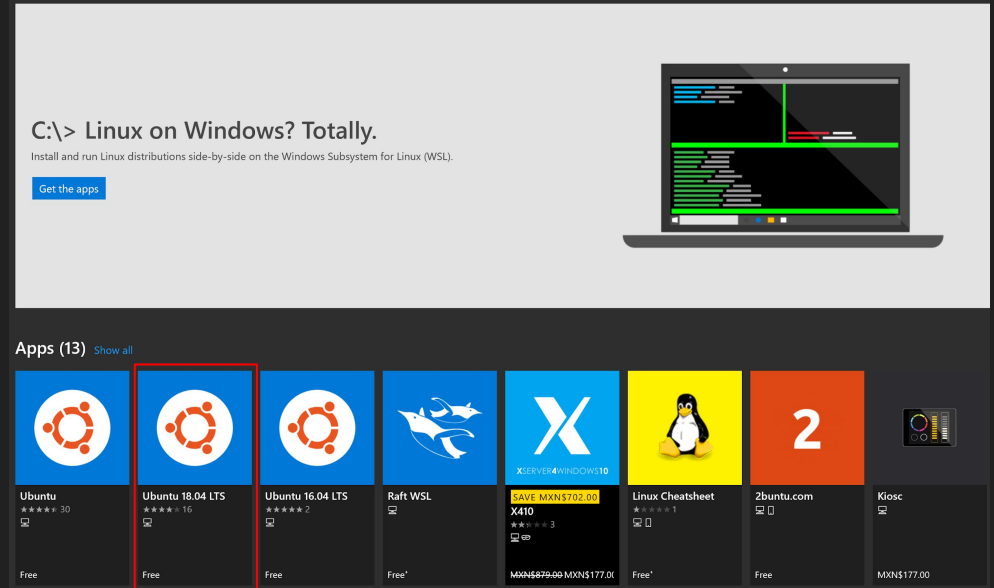
# Instalación Para Bash

Si tienes Mac... Tu ntp tu terminal ya es bash

Para windows:

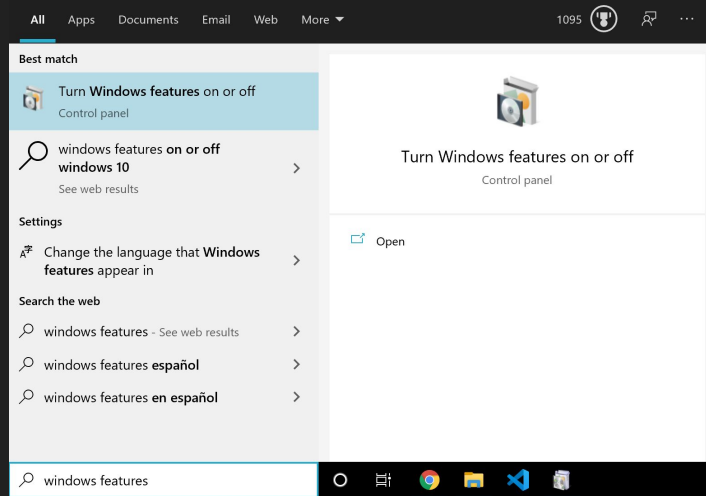
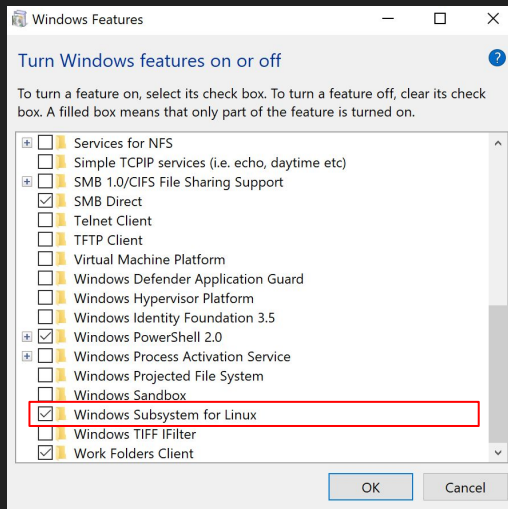
Entra a la Windows Store

Busca y descarga Ubuntu 18.04 TLS



# Y ya?

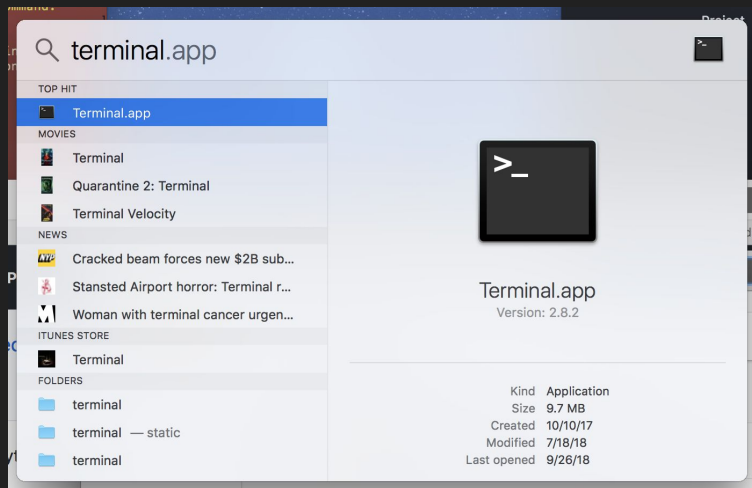
Para poder usar la terminal de bash en Windows, busca en tu computadora Windows Features.



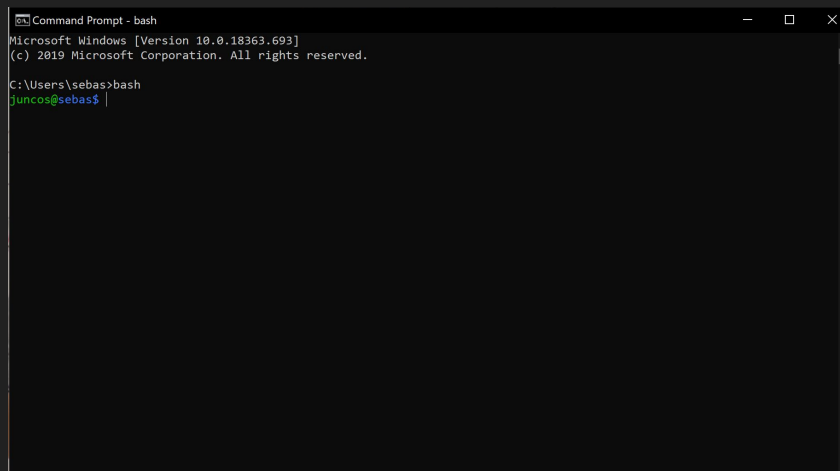
Ya que estés ahí, busca Windows Subsystem for Linux. Esto necesita reiniciar tu compu.

# Ahora si

Mac: teclea command + space bar y busca terminal.



Windows: en el buscador, busca CMD y cuando estés ahí, teclea bash.





# Bash Commands

ls → lista todos los contenidos del directorio en el que te encuentras

Mkdir → crea carpetas

pwd → printea el directorio completo donde te encuentras

cd Windows/Users/... → cambia de directorio

rm ... → borra un archivo

clear → Limpia la terminal para que no se vean los comandos de arriba

# Descarguen VS Code

Busquen en google  
vscode o entren a  
<https://code.visualstudio.com/> y descarguen  
vscode

## Code editing. Redefined.

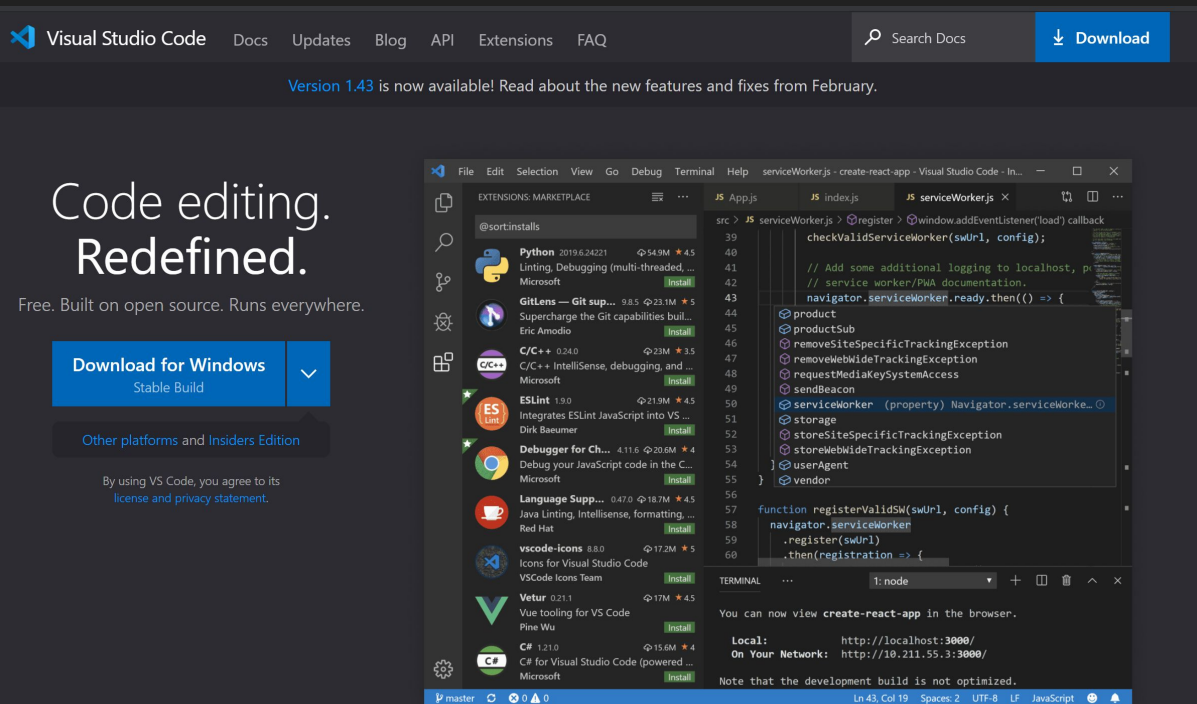
Free. Built on open source. Runs everywhere.

Download for Windows

Stable Build

Other platforms and Insiders Edition

By using VS Code, you agree to its  
license and privacy statement.



IntelliSense



Debugging



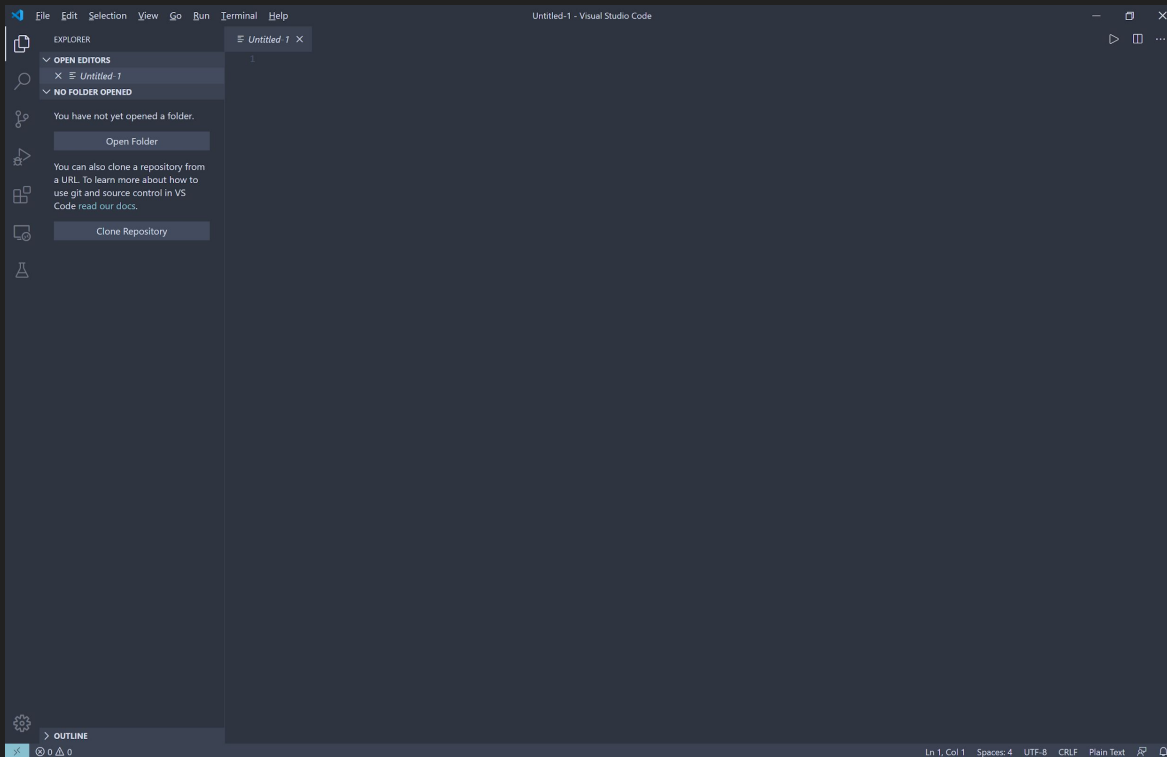
Built-in Git



Extensions

# Ahora se viene lo bueno

Abran Visual Studio Code:



# Primero que nada, pongamos bash:

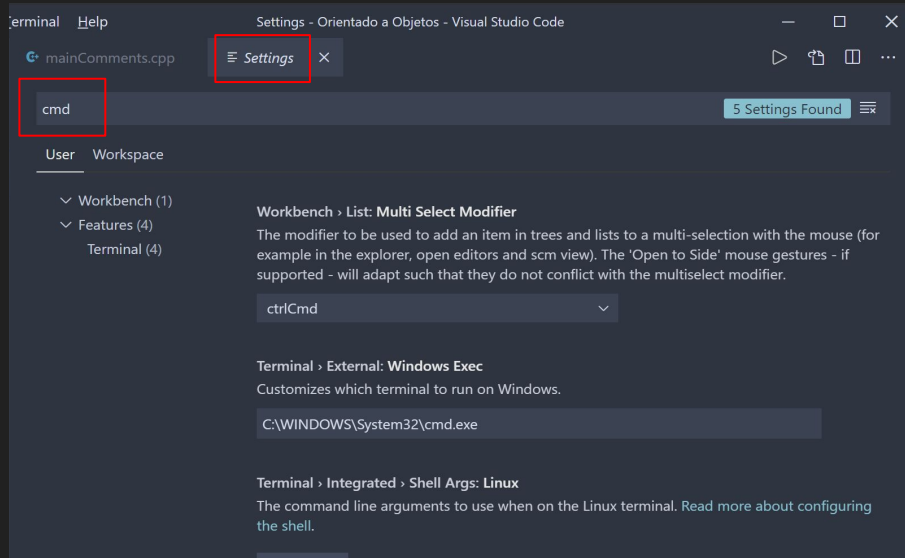
Si tienes Mac... Tu ntp tu terminal ya es bash

Windows:

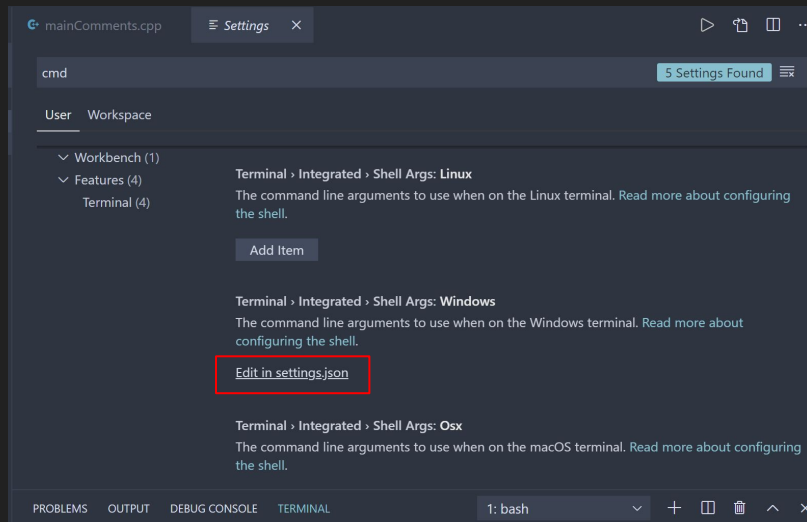
File → Preferences → Settings

(O solo usen ctrl + ,)

En settings busquen cmd



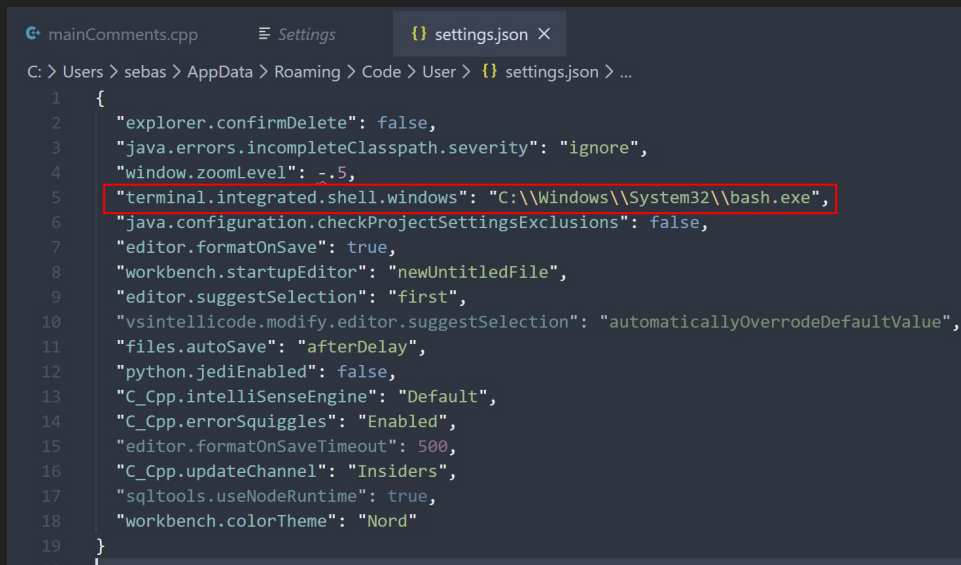
Scrollea hasta que encuentres Edit  
in settings.json



Ya ahí cambia o teclea lo siguiente:

```
"terminal.integrated.shell.windows":  
:"C:\\Windows\\System32\\bash.exe"
```

Tu settings.json no va a ser igual al mio, no te preocupes por los otros comandos.



# Para C++ en Windows

En la terminal de bash, ya sea en el programa dentro de tu compu o en la terminal de VS Code tecela:

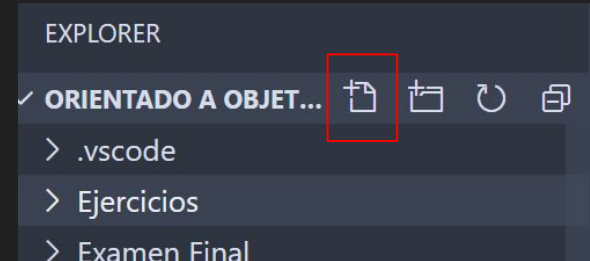
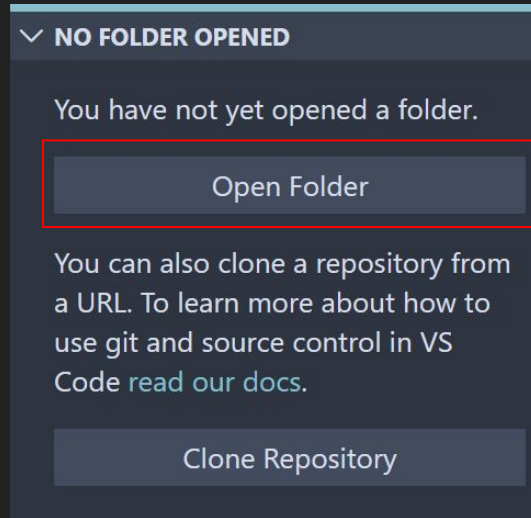
Primero actualizamos bash:  
`$sudo apt-get update`

Instalamos el compilador de c++:  
`$sudo apt-get install g++`

El "\$" es parte de la terminal, no lo tienen que agregar al comando, osea tienen que poner:

`Sudo apt-get update`

Ya tenemos todo, ahora abre un folder y crea un nuevo archivo



# Para C++ para mac

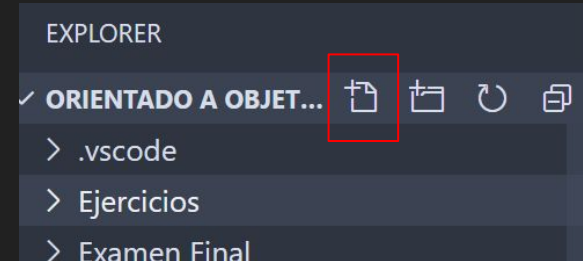
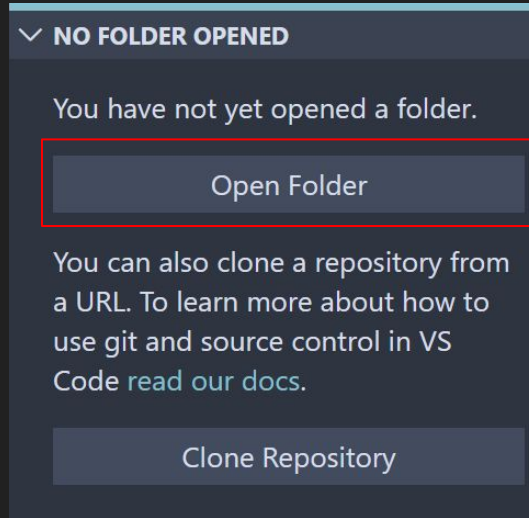
En la terminal de mac, ya sea en el programa dentro de tu compu o en la terminal de VS Code tecela:

\$g++

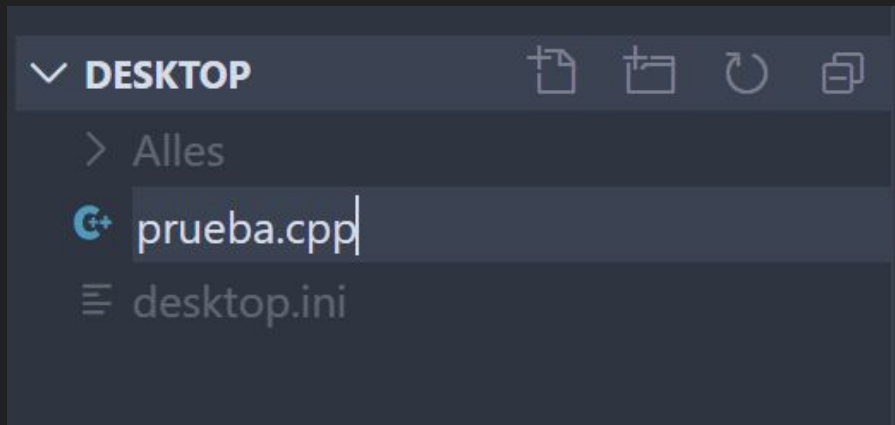
Y te va a salir esto:



Ya tenemos todo, ahora abre un folder y crea un nuevo archivo



Guarda el archivo con el nombre que quieras y .cpp



Escribe lo siguiente:

```
#include <iostream>
using namespace std;
int main()
{
    printf("Hello World"); //
para c
    cout << "Hello World" <<
endl; // para c++
    return 0;
}
```



Ahora Visual mostrará un pop up en la esquina inferior derecha, esta te sugiere instalar la extensión de C/C++, instala la extensión, además te recomiendo que descargues de las marcadas en la imagen

The image shows the Visual Studio Code interface. On the left, the 'Extensions Marketplace' sidebar is open, displaying a list of extensions. The 'C/C++' extension by Microsoft is highlighted with a red box. Below it, other extensions like 'C/C++ Compile Run', 'C/C++ Snippets', 'C++ Intellisense', 'Code Runner', 'Debugger for Java', 'Java Dependency Viewer', 'Java Extension Pack', 'Java Test Runner', and 'Language Support for Java/T' are visible. On the right, the 'C/C++ for Visual Studio Code' extension details page is shown. It features the extension's logo, name, publisher (Microsoft), version (0.27.0-insiders2), and a 'Preview' badge. The page also displays the number of downloads (10,006,058), a star rating (4.5 stars), and links to the repository and license. The extension is currently 'enabled globally'. Below this, there is a recommendation to install the extension based on recently opened files. The 'Details' tab is selected, showing an overview and getting started guide. The 'Quick links' section at the bottom lists links to the extension overview, getting started guides for various environments (WSL, Mingw-w64, macOS, MSVC), and editing features.

Search Extensions in Marketplace

ENABLED 15

- ★ **C/C++** 0.27.0-insiders2  
C/C++ IntelliSense, debugging, and c...  
Microsoft
- C/C++ Compile Run** 1.0.5  
Compile & Run single c/c++ files easily  
danielpinto8zz6
- C/C++ Snippets** 0.0.14  
Code snippets for C/C++  
Harsh
- ★ **C++ Intellisense** 0.2.2  
C/C++ Intellisense with the help of G...  
austin
- Code Runner** 0.9.16  
Run C, C++, Java, JS, PHP, Python, Per...  
Jun Han
- Debugger for Java** 0.25.1  
A lightweight Java debugger for Visua...  
Microsoft
- Java Dependency Viewer** 0.9.0  
Manage Java Dependencies in VSCode  
Microsoft
- Java Extension Pack** 0.8.1  
Popular extensions for Java developm...  
Microsoft
- Java Test Runner** 0.22.2  
Run and debug JUnit or TestNG test c...  
Microsoft
- Language Support for Java/T** 0.58.0  
RECOMMENDED 0
- DISABLED 0

**C/C++** ms-vscode.cpptools Preview

Microsoft | 10,006,058 | ★★★★★ | Repository | License

C/C++ IntelliSense, debugging, and code browsing.

Disable ▾ | Uninstall | This extension is enabled globally.

This extension is recommended based on the files you recently opened.

Ignore Recommendation

Details | Feature Contributions | Changelog

## C/C++ for Visual Studio Code

Repository | Issues | Documentation | Code Samples | Offline Installers

Live Share enabled

This preview release of the C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.

### Overview and getting started

- C/C++ extension overview
- Get Started with C++ and Windows Subsystem for Linux (WSL)
- Get Started with C++ and Mingw-w64
- Get Started with C++ and Clang/LLVM on macOS
- Get Started with C++ and Microsoft C++ compiler (MSVC)

### Quick links



- Editing features (IntelliSense)

# Compilar y correr en C++




Ya que tengas tu archivo:

```
#include <iostream>
using namespace std;
int main()
{
    cout << "Hello World" << endl;
    return 0;
}
```

Abre la terminal, ahora para compilar:

```
juncos@Desktop$ g++ *.cpp  V1
juncos@Desktop$ g++ -o prueba1 *.cpp V2
juncos@Desktop$ 
```

Para correr:

```
juncos@Desktop$ ./a.out  V1
Hello World
juncos@Desktop$ ./prueba1  V2
Hello World
juncos@Desktop$ 
```

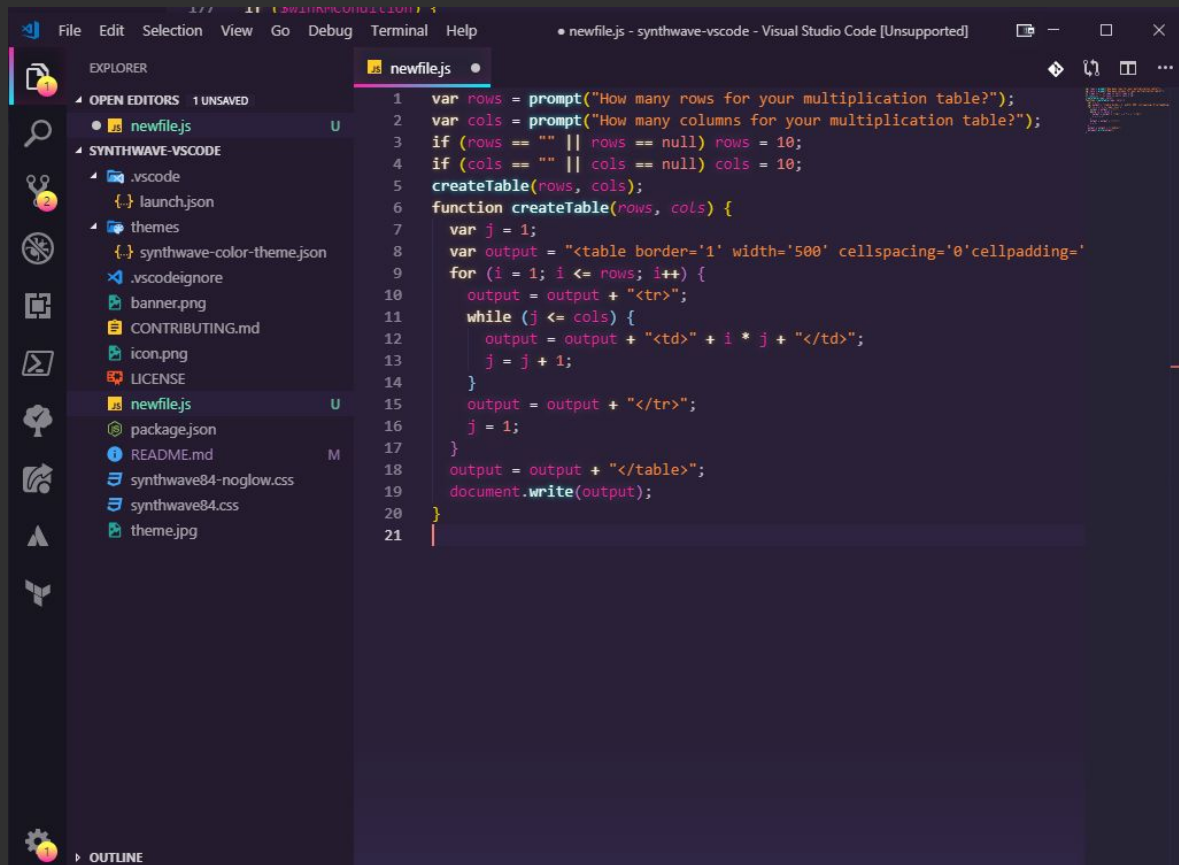
# Temas Recomendados

Pa que quede chula la  
cosa



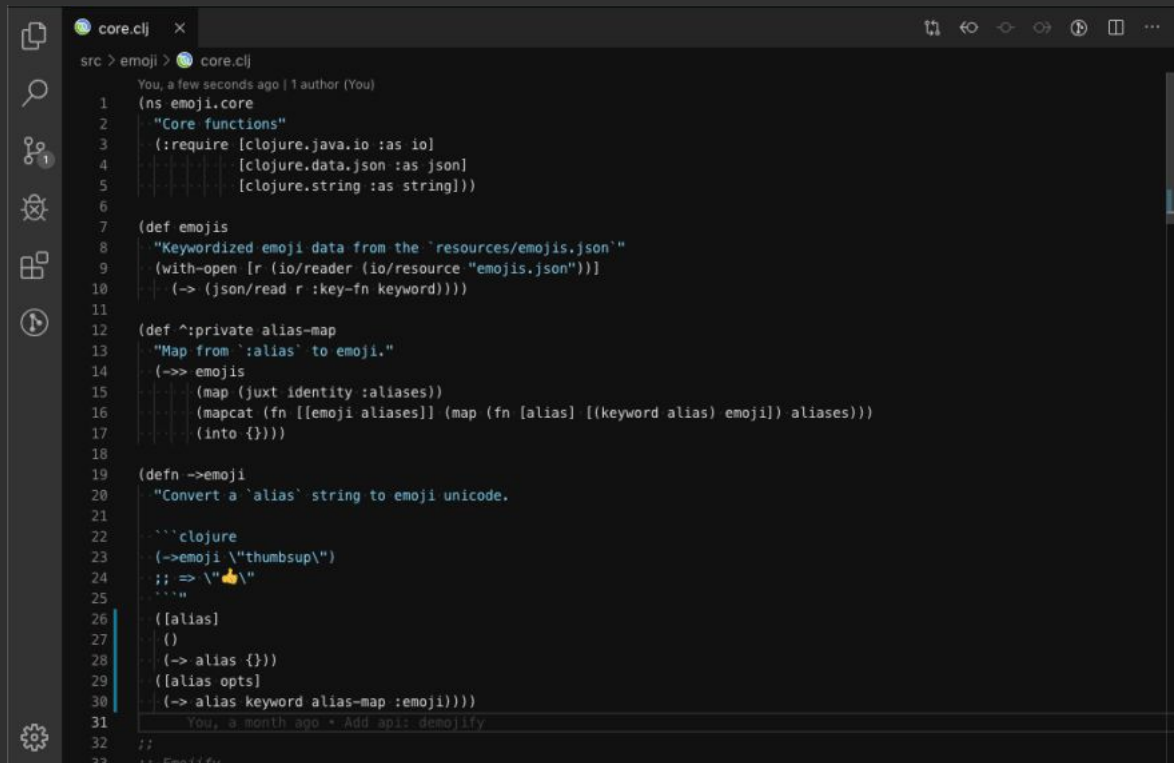
# Synth Wave '84

Si quieren que las palabras brillen, mandenme mensaje por wa, necesitas hacer unos cuantos pasos extra



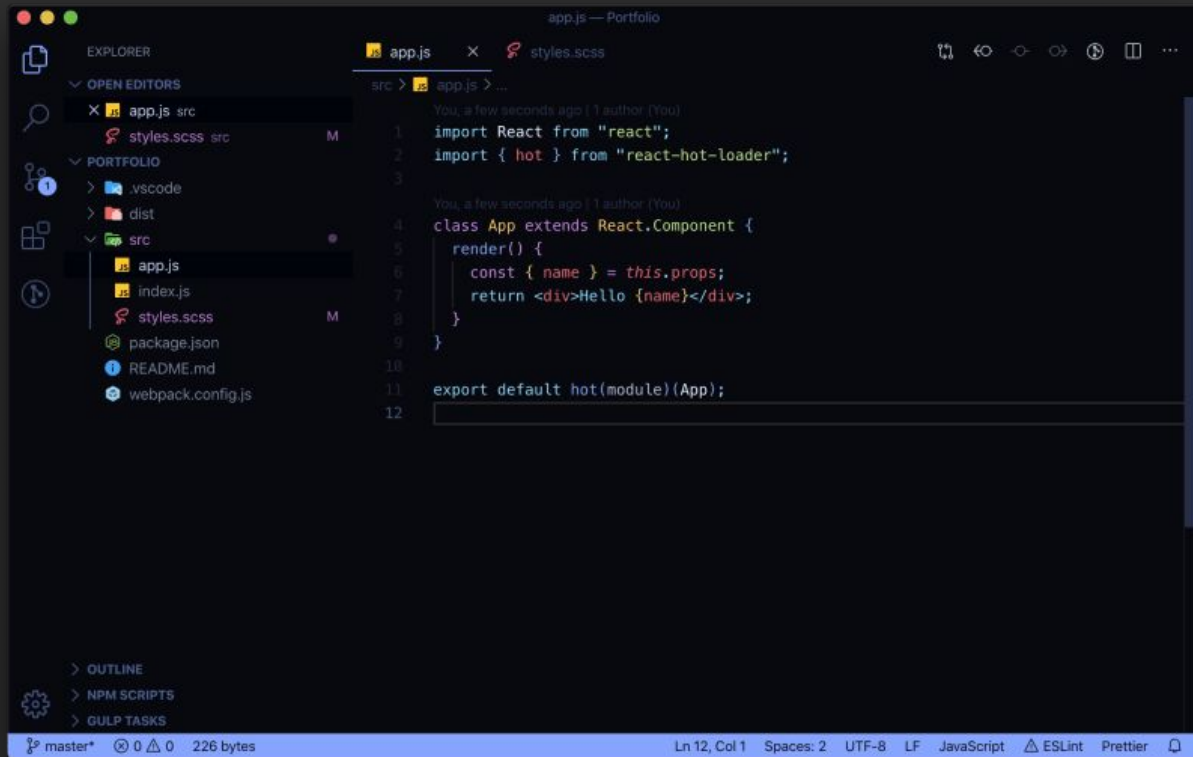
```
1 var rows = prompt("How many rows for your multiplication table?");
2 var cols = prompt("How many columns for your multiplication table?");
3 if (rows == "" || rows == null) rows = 10;
4 if (cols == "" || cols == null) cols = 10;
5 createTable(rows, cols);
6 function createTable(rows, cols) {
7     var j = 1;
8     var output = "<table border='1' width='500' cellpadding='0' cellspacing='0'>";
9     for (i = 1; i <= rows; i++) {
10         output = output + "<tr>";
11         while (j <= cols) {
12             output = output + "<td>" + i * j + "</td>";
13             j = j + 1;
14         }
15         output = output + "</tr>";
16         j = 1;
17     }
18     output = output + "</table>";
19     document.write(output);
20 }
21
```

# minimal

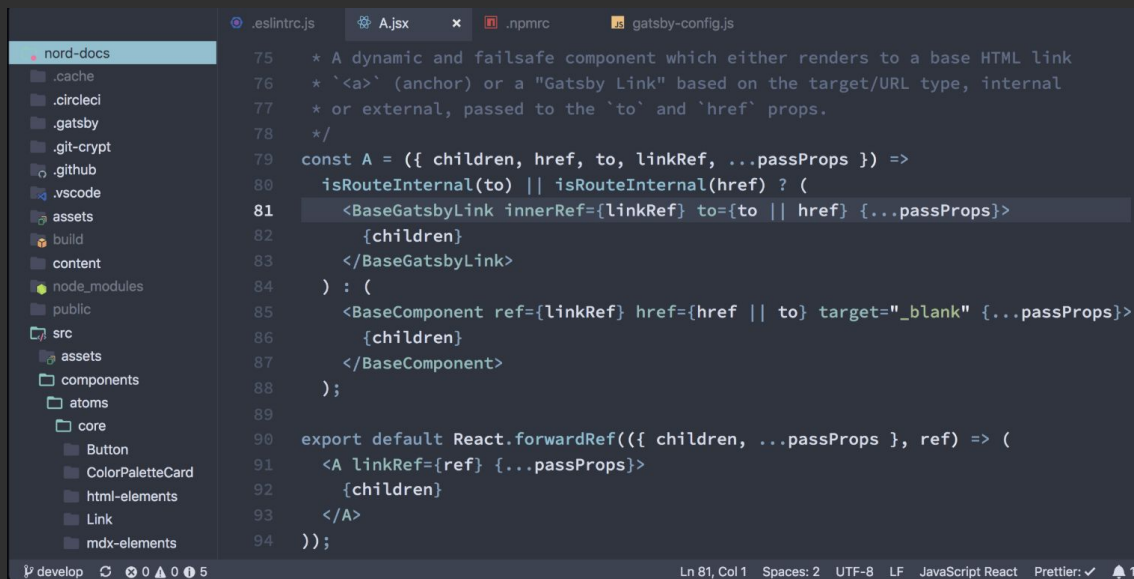


```
core.clj x
src > emoji > core.clj
You, a few seconds ago | 1 author (You)
1 (ns emoji.core
2   "Core functions"
3   (:require [clojure.java.io :as io]
4             [clojure.data.json :as json]
5             [clojure.string :as string]))
6
7 (def emojis
8   "Keywordized emoji data from the `resources/emojis.json`"
9   (with-open [r (io/reader (io/resource "emojis.json"))]
10     (-> (json/read r :key-fn keyword))))
11
12 (def ^:private alias-map
13   "Map from `:alias` to emoji."
14   (-> emojis
15       (map (juxt identity :aliases))
16       (mapcat (fn [[emoji aliases]] (map (fn [alias] [(keyword alias) emoji]) aliases))
17       (into {})))
18
19 (defn ->emoji
20   "Convert a `alias` string to emoji unicode."
21
22   ```clojure
23   (->emoji \"thumbsup\")
24   ;; => \"👍\"
25   ```
26
27   ([alias]
28    ()
29    (-> alias {}))
30   ([alias opts]
31    (-> alias keyword alias-map :emoji)))
32
33 You, a month ago • Add api: demoify
34 ;;
35 ;; Emojify
```

# Aurora X



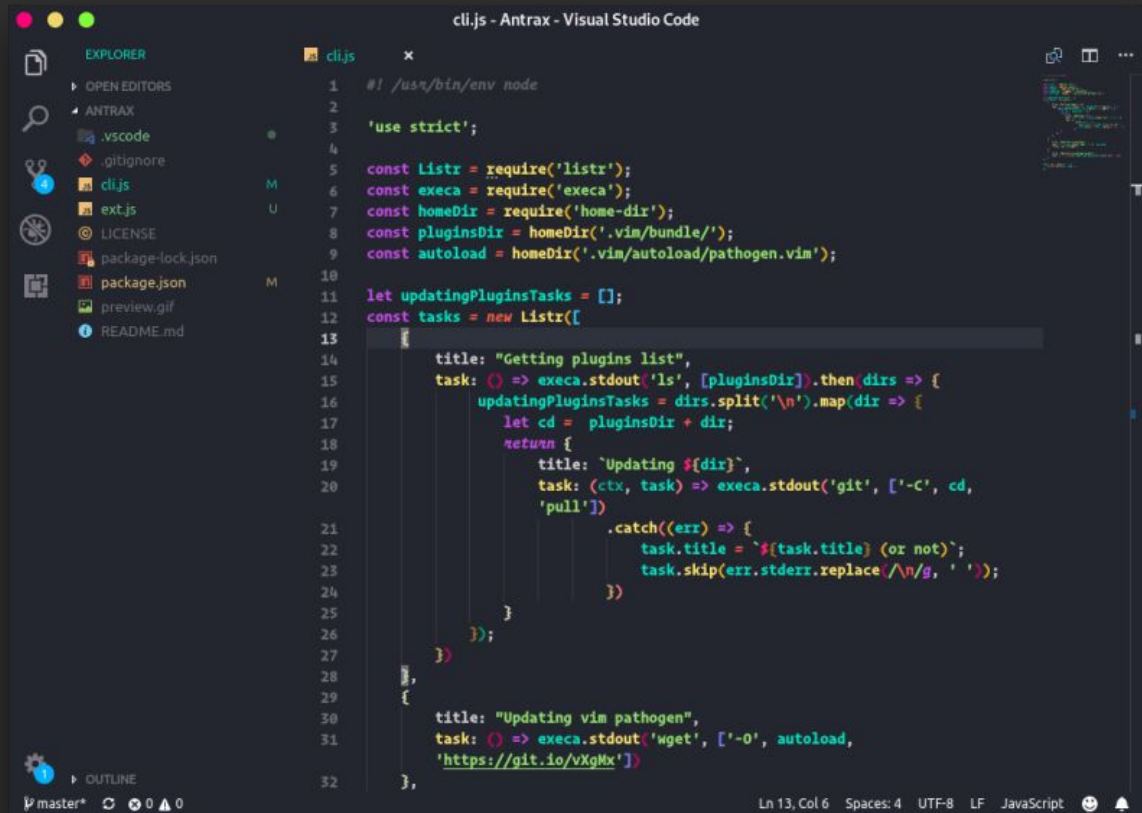
# Nord



The screenshot shows a VS Code editor with a dark theme. On the left is the Explorer sidebar showing a file tree for a project named 'nord-docs'. The tree includes folders like '.cache', '.circleci', '.gatsby', '.git-crypt', '.github', '.vscode', 'assets', 'build', 'content', 'node\_modules', 'public', 'src', and 'components'. The 'src' folder is expanded, showing 'assets', 'components', 'atoms', and 'core'. The 'core' folder is further expanded, showing 'Button', 'ColorPaletteCard', 'html-elements', 'Link', and 'mdx-elements'. The main editor area has four tabs: '.eslintrc.js', 'A.jsx', '.npmrc', and 'gatsby-config.js'. The 'gatsby-config.js' tab is active, displaying a JavaScript component definition for a dynamic and failsafe link. The code uses React's forwardRef and PropTypes to create a component that can render either a standard HTML anchor or a Gatsby Link based on the target type. The status bar at the bottom indicates the current file is 'Ln 81, Col 1', uses 'Spaces: 2', is in 'UTF-8' encoding, and has 'JavaScript React' syntax highlighting. It also shows 'Prettier' formatting is enabled and there is 1 error.

```
75  * A dynamic and failsafe component which either renders to a base HTML link
76  * `` (anchor) or a "Gatsby Link" based on the target/URL type, internal
77  * or external, passed to the `to` and `href` props.
78  */
79  const A = ({ children, href, to, linkRef, ...passProps }) =>
80    isRouteInternal(to) || isRouteInternal(href) ? (
81      <BaseGatsbyLink innerRef={linkRef} to={to || href} {...passProps}>
82        {children}
83      </BaseGatsbyLink>
84    ) : (
85      <BaseComponent ref={linkRef} href={href || to} target="_blank" {...passProps}>
86        {children}
87      </BaseComponent>
88    );
89
90  export default React.forwardRef(({ children, ...passProps }, ref) => (
91    <A linkRef={ref} {...passProps}>
92      {children}
93    </A>
94  ));
```

# Andromeda



```
cli.js - Antrax - Visual Studio Code

1  #! /usr/bin/env node
2
3  'use strict';
4
5  const Listr = require('listr');
6  const execa = require('execa');
7  const homeDir = require('home-dir');
8  const pluginsDir = homeDir('.vim/bundle/');
9  const autoload = homeDir('.vim/autoload/pathogen.vim');
10
11  let updatingPluginsTasks = [];
12  const tasks = new Listr([
13    {
14      title: "Getting plugins list",
15      task: () => execa.stdout('ls', [pluginsDir]).then(dirs => {
16        updatingPluginsTasks = dirs.split('\n').map(dir => {
17          let cd = pluginsDir + dir;
18          return {
19            title: `Updating ${dir}`,
20            task: (ctx, task) => execa.stdout('git', ['-C', cd,
21              'pull'])
22              .catch((err) => {
23                task.title = `${task.title} (or not)`;
24                task.skip(err.stderr.replace(/\n/g, ' '));
25              })
26          };
27        });
28      }
29    },
30    {
31      title: "Updating vim pathogen",
32      task: () => execa.stdout('wget', ['-O-', autoload,
33        'https://git.io/vXgMx'])
34    }
35  ])
```

master\* 0 0 0

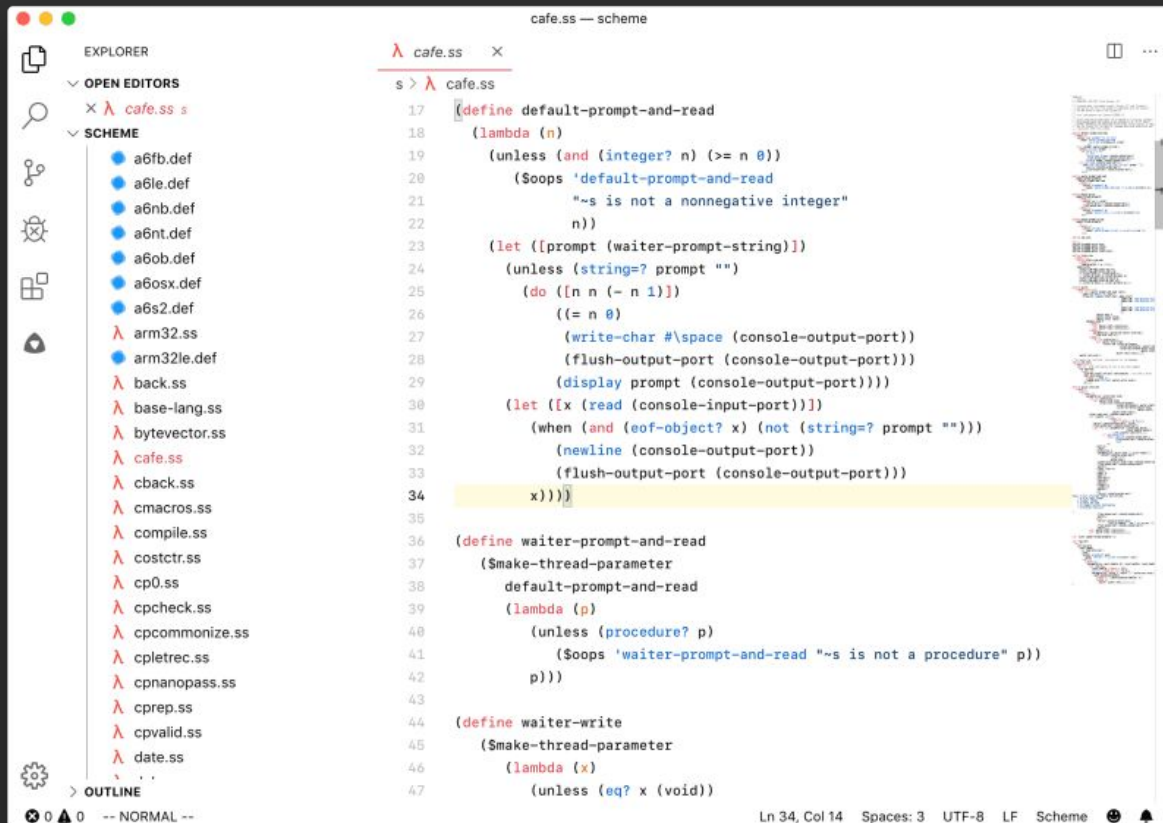
Ln 13, Col 6 Spaces: 4 UTF-8 LF JavaScript



# Quiet Light

```
<!doctype html>
<html class="no-js" lang="">
<head>
  <meta charset="utf-8">
  <meta http-equiv="x-ua-compatible" content="ie=edge">
  <title>My Excellent Webpage</title>
  <style>
    body, html {
      background: #fff;
      font: Helvetica, Arial, sans-serif;
    }
  </style>
  <script>
    $(document).ready(function() {
      var foo = 'bar';
      fooTheBar(foo);
    });
  </script>
</head>
```

# GitHub Light



cafe.ss — scheme

EXPLORER

OPEN EDITORS

× **cafe.ss**

SCHEME

- a6fb.def
- a6le.def
- a6nb.def
- a6nt.def
- a6ob.def
- a6osx.def
- a6s2.def
- arm32.ss
- arm32le.def
- back.ss
- base-lang.ss
- bytevector.ss
- cafe.ss**
- cback.ss
- cmacros.ss
- compile.ss
- costctr.ss
- cp0.ss
- cpcheck.ss
- cpcommonize.ss
- cpletrec.ss
- cpnanopass.ss
- cprep.ss
- cpvalid.ss
- date.ss

OUTLINE

0 0 -- NORMAL --

```
λ cafe.ss
s > λ cafe.ss

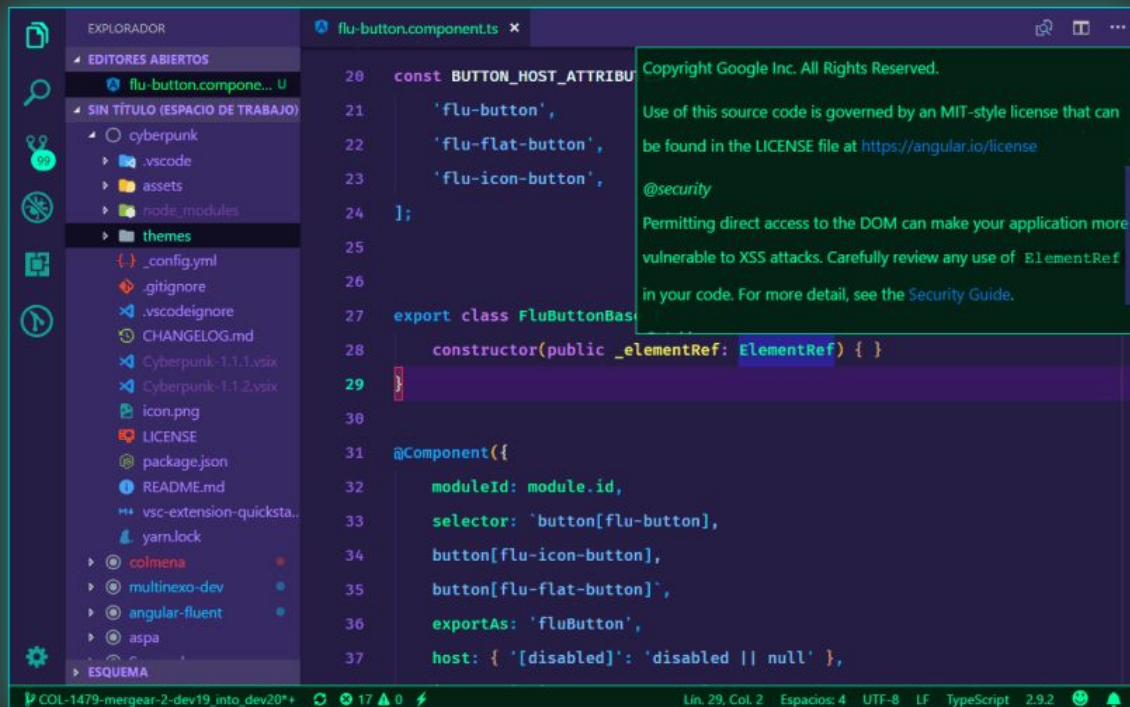
17 (define default-prompt-and-read
18   (lambda (n)
19     (unless (and (integer? n) (>= n 0))
20       ($oops 'default-prompt-and-read
21         "~s is not a nonnegative integer"
22         n))
23     (let ([prompt (waiter-prompt-string)])
24       (unless (string=? prompt "")
25         (do ([n n (~ n 1)])
26             [(= n 0)
27              (write-char #\space (console-output-port))
28              (flush-output-port (console-output-port))]
29              (display prompt (console-output-port)))]
30         (let ([x (read (console-input-port))])
31           (when (and (eof-object? x) (not (string=? prompt "")))
32             (newline (console-output-port))
33             (flush-output-port (console-output-port)))
34           x))))))

36 (define waiter-prompt-and-read
37   ($make-thread-parameter
38     default-prompt-and-read
39     (lambda (p)
40       (unless (procedure? p)
41         ($oops 'waiter-prompt-and-read "~s is not a procedure" p))
42       p)))

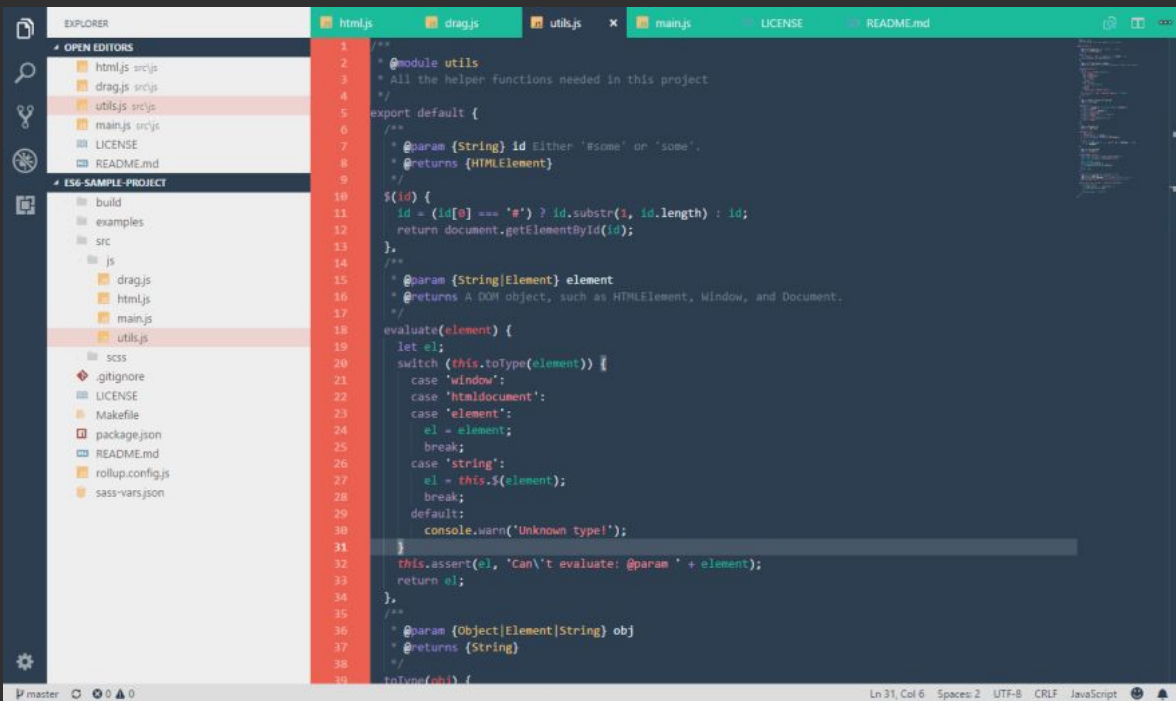
44 (define waiter-write
45   ($make-thread-parameter
46     (lambda (x)
47       (unless (eq? x (void))
```

Ln 34, Col 14 Spaces: 3 UTF-8 LF Scheme

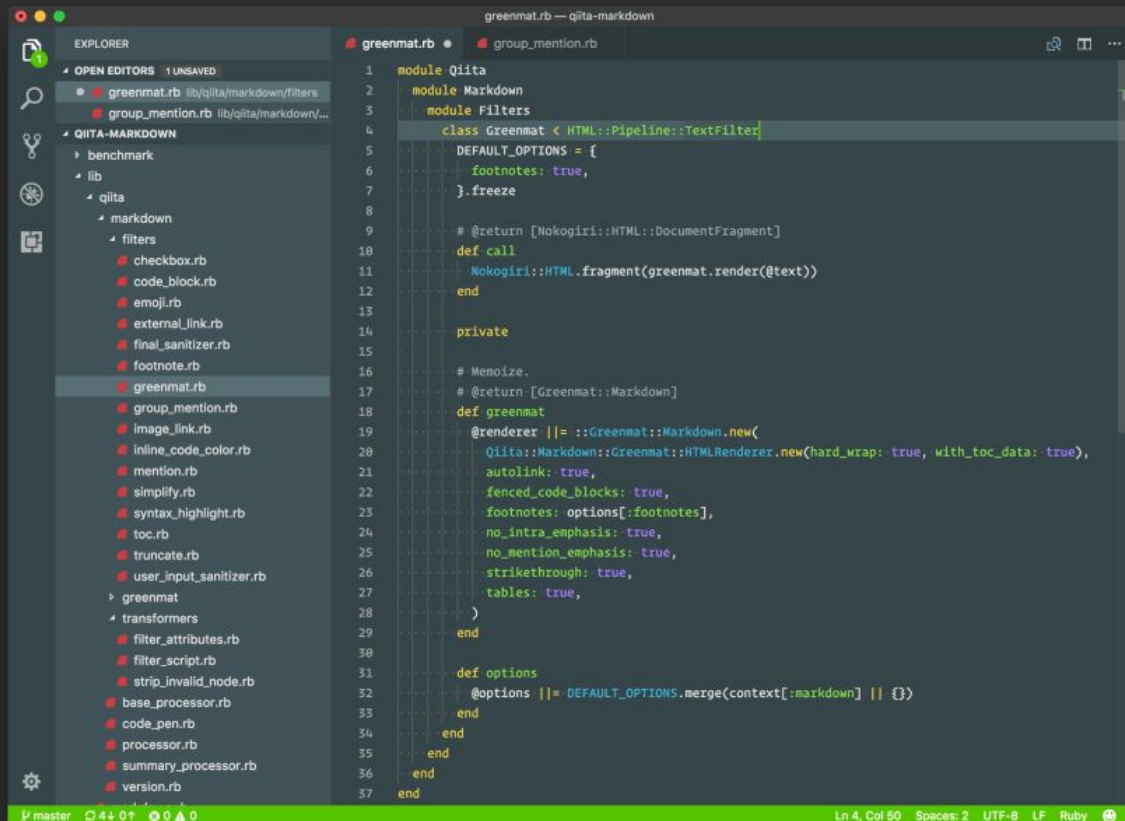
# Cyberpunk



# Hipster



# Qiita



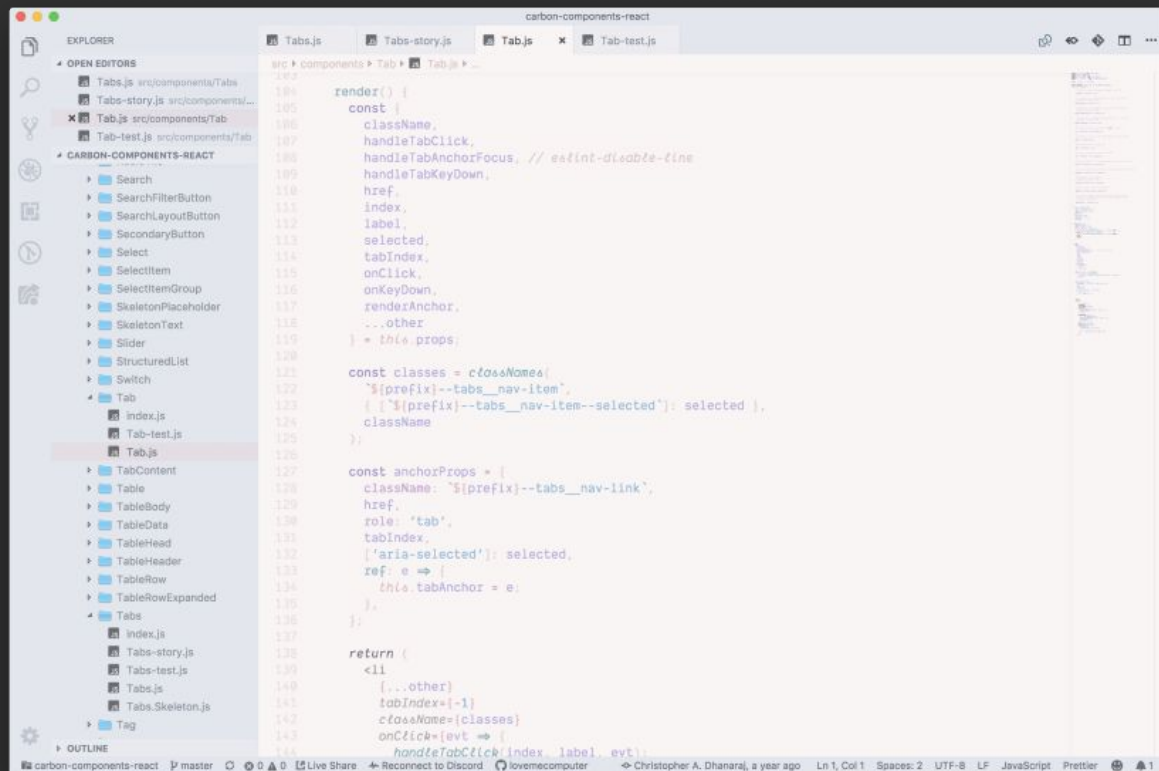
The screenshot shows a VS Code editor with the following components:

- EXPLORER:** Displays the project structure. The 'lib' directory is expanded, showing subdirectories 'qiita' and 'greenmat'. The 'qiita' directory is further expanded, showing 'markdown' and 'filters'. The 'filters' directory is expanded, showing various filter files. The 'greenmat' directory is also expanded, showing 'transformers' and 'processors'.
- OPEN EDITORS:** Shows the current file being edited: 'greenmat.rb'.
- EDITOR:** Displays the code for 'greenmat.rb'. The code defines a 'Qiita' module with a 'Markdown' class and a 'Filters' module. The 'Filters' module contains a 'Greenmat' class that inherits from 'HTML::Pipeline::TextFilter'. The 'Greenmat' class has a 'DEFAULT\_OPTIONS' hash and a 'render' method that uses 'Nokogiri' to render the text.

```
1 module Qiita
2   module Markdown
3     module Filters
4       class Greenmat < HTML::Pipeline::TextFilter
5         DEFAULT_OPTIONS = {
6           footnotes: true,
7         }.freeze
8
9         # @return [Nokogiri::HTML::DocumentFragment]
10        def call
11          Nokogiri::HTML.fragment(greenmat.render(@text))
12        end
13
14        private
15
16        # Memoize.
17        # @return [Greenmat::Markdown]
18        def greenmat
19          @renderer ||= ::Greenmat::Markdown.new(
20            Qiita::Markdown::Greenmat::HTMLRenderer.new(hard_wrap: true, with_toc_data: true),
21            autolink: true,
22            fenced_code_blocks: true,
23            footnotes: options[:footnotes],
24            no_intra_emphasis: true,
25            no_mention_emphasis: true,
26            strikethrough: true,
27            tables: true,
28          )
29        end
30
31        def options
32          @options ||= DEFAULT_OPTIONS.merge(context[:markdown] || {})
33        end
34      end
35    end
36  end
37 end
```

At the bottom of the editor, the status bar shows: 'master', '44 0+', '0 0', 'Ln 4, Col 50', 'Spaces: 2', 'UTF-8', 'LF', 'Ruby'.

# Soft Era

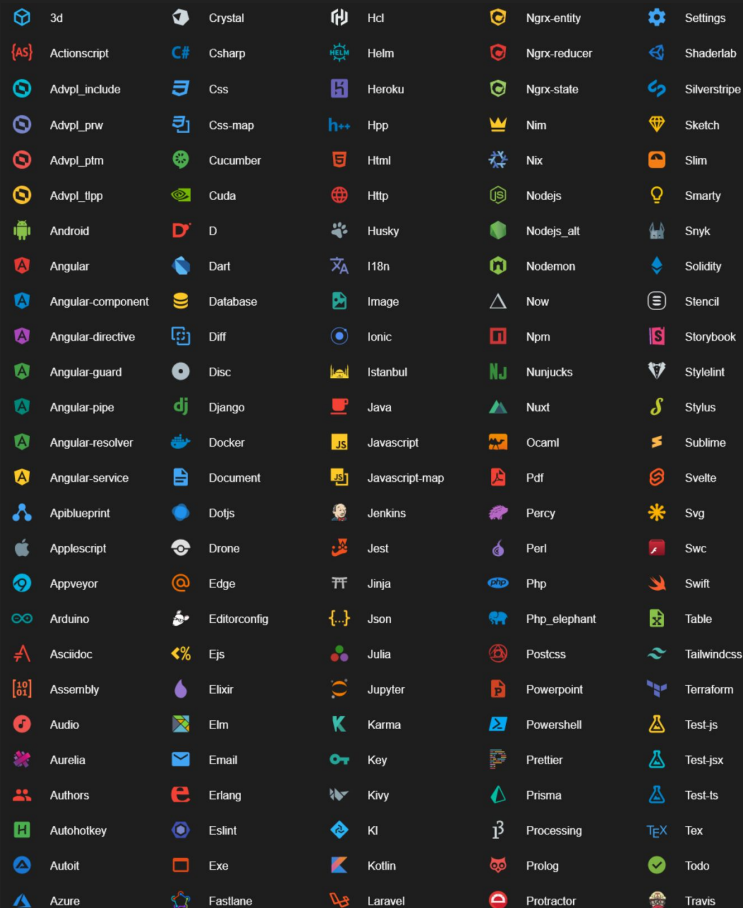


```
1402 render() {
1403   const {
1404     className,
1405     handleTabClick,
1406     handleTabAnchorFocus, // eslint-disable-line
1407     handleTabKeyDown,
1408     href,
1409     index,
1410     label,
1411     selected,
1412     tabIndex,
1413     onClick,
1414     onKeyDown,
1415     renderAnchor,
1416     ...other
1417   } = this.props;
1418
1419   const classes = classNames(
1420     `${prefix}--tabs__nav-item`,
1421     { `${prefix}--tabs__nav-item--selected`: selected },
1422     className
1423   );
1424
1425   const anchorProps = {
1426     className: `${prefix}--tabs__nav-link`,
1427     href,
1428     role: 'tab',
1429     tabIndex,
1430     ['aria-selected']: selected,
1431     ref: e => {
1432       this.tabAnchor = e;
1433     },
1434   };
1435
1436   return (
1437     <li
1438       {...other}
1439       tabIndex={-1}
1440       className={classes}
1441       onClick={evt => {
1442         handleTabClick(index, label, evt);
1443       }}
1444     >
```

carbon-components-react > master | Live Share | Reconnect to Discord | lovemecomputer | Christopher A. Dhanaraj, a year ago | Ln 1, Col 1 | Spaces: 2 | UTF-8 | LF | JavaScript | Prettier | 1

# Icons

Todavía más chulo



# Quill

## ▼ APP

data

config.json

src

utilities

app.js

styles

layout.css

.eslintrc

.gitignore

.prettierrc

<> index.html

package.json



































































































































README.md

> OUTLINE

> NPM SCRIPTS



# Helium

|   |  |  |  |  |
|---|--|--|--|--|
|  3d                 |  Crystal       |  Hcl             |  NgRx-entity   |  Settings      |
|  Actionscript      |  Csharp       |  Helm           |  NgRx-reducer |  Shadertab    |
|  Advpl_include     |  Css          |  Heroku         |  NgRx-state   |  Silverstripe |
|  Advpl_prw         |  Css-map      |  Hpp            |  Nim          |  Sketch       |
|  Advpl_plm         |  Cucumber     |  Html           |  Nix          |  Slim         |
|  Advpl_tlp         |  Cuda         |  Http           |  Nodejs       |  Smarty       |
|  Android           |  D            |  Husky          |  Nodejs_alt   |  Snyk         |
|  Angular           |  Dart         |  I18n           |  Nodemon      |  Solidity     |
|  Angular-component |  Database     |  Image          |  Now          |  Stencil      |
|  Angular-directive |  Diff         |  Ionic          |  Npm          |  Storybook    |
|  Angular-guard     |  Disc         |  Istanbul       |  Nunjucks     |  Stylelint    |
|  Angular-pipe      |  Django       |  Java           |  Nuxt         |  Stylus       |
|  Angular-resolver  |  Docker       |  Javascript     |  Ocaml        |  Sublime      |
|  Angular-service   |  Document     |  Javascript-map |  Pdf          |  Svelte       |
|  Apibluemprint     |  Dotjs        |  Jenkins        |  Percy        |  Svg          |
|  Applescript       |  Drone        |  Jest           |  Perl         |  Swc          |
|  Appveyor          |  Edge         |  Jinja          |  Php          |  Swift        |
|  Arduino           |  Editorconfig |  Json           |  Php_elephant |  Table        |
|  Ascii8doc         |  Ejs          |  Julia          |  Postcss      |  Tailwindcss  |
|  Assembly          |  Elair        |  Jupyter        |  Powerpoint   |  Terraform    |
|  Audio             |  Elm          |  Karma          |  Powershell   |  Test-js      |
|  Aurelia           |  Email        |  Key            |  Prettier     |  Test-jsx     |
|  Authors           |  Erlang       |  Kivy           |  Prisma       |  Test-ts      |
|  Autohotkey        |  Eslint       |  KI             |  Processing   |  TeX          |
|  Autoit            |  Exe          |  Kotlin         |  Prolog       |  Todo         |
|  Azure            |  Fastlane    |  Laravel       |  Protractor  |  Travis      |





# Seedling

```
> node_modules
> public
▼ src
  # App.css
  ⚙ App.test.tsx
  ⚙ App.tsx
  # index.css
  ⚙ index.tsx
  🖼 logo.svg
  JS profile.js
  TS registerServiceWorker.ts
  📄 .gitignore
  TS images.d.ts
  📄 package.json
  📄 README.md
  📄 tsconfig.json
  📄 tsconfig.prod.json
  📄 tsconfig.test.json
  📄 tslint.json
  📄 yarn.lock
```





```
> node_modules
> public
▼ src
  # App.css
  ⚙ App.test.tsx
  ⚙ App.tsx
  # index.css
  ⚙ index.tsx
  🖼 logo.svg
  JS profile.js
  TS registerServiceWorker.ts
  📄 .gitignore
  TS images.d.ts
  📄 package.json
  📄 README.md
  📄 tsconfig.json
  📄 tsconfig.prod.json
  📄 tsconfig.test.json
  📄 tslint.json
  📄 yarn.lock
```

# Chalice

```

 closed
 demo
  ( ) clojure.clj
   file.txt
  λ  haskell.hs
   image.ico
  <> index.html
  /* index.js
  <> index.jsx
  {} json.json
  ≡ markdown.md
  >_ script.sh
  # style.css
```

```

 closed
 demo
  ( ) clojure.clj
   file.txt
  λ  haskell.hs
   image.ico
  <> index.html
  /* index.js
  <> index.jsx
  {} json.json
  ≡ markdown.md
  >_ script.sh
  # style.css
```