



A curated list of delightful [Visual Studio Code](#) packages and resources. For more awesomeness, check out [awesome](#).

 awesome 

## Table of Contents

---

- [Official](#)
- [Syntax](#)
- [Migrating from other editors](#)
  - [Migrating from Vim](#)
  - [Migrating from Atom](#)
  - [Migrating from Sublime Text](#)
  - [Migrating from Visual Studio](#)
  - [Migrating from IntelliJ IDEA](#)
- [Using VS Code with particular technologies](#)
- [Lint and IntelliSense](#)
  - [1C](#)
  - [Bash](#)
    - [Bash IDE](#)
    - [Bash Debug](#)
  - [C++](#)
  - [C#](#)
  - [Clojure](#)
  - [CSS](#)
  - [Go](#)
  - [Haskell](#)

- [Shell](#)
- [Java](#)
- [JavaScript](#)
- [TypeScript](#)
- [Markdown](#)
  - [markdownlint](#)
  - [Markdown All in one](#)
- [PHP](#)
  - [IntelliSense](#)
  - [Laravel](#)
  - [Twig](#)
  - [Other extensions](#)
  - [Read more](#)
- [Python](#)
  - [TensorFlow](#)
- [ReasonML](#)
- [Rust](#)
- [Terraform](#)
- Productivity
  - [Azure Cosmos DB](#)
  - [Azure IoT Toolkit](#)
  - [Bookmarks](#)
  - [Color Tabs](#)
  - [Create tests](#)
  - [Deploy](#)
  - [Duplicate Action](#)
  - [ES7 React/Redux/GraphQL/React-Native snippets](#)
  - [Gi](#)
  - [Git History](#)
  - [Git Project Manager](#)
  - [GitLink](#)
  - [GitLens](#)
  - [Git Indicators](#)
  - [GitHub](#)
  - [GitHub Pull Request Monitor](#)
  - [Icon Fonts](#)
  - [Import Cost](#)
  - [JS Parameter Annotations](#)
  - [Jumpy](#)
  - [Kanban](#)
  - [Live Server](#)
  - [Multiple clipboards](#)
  - [Dotnet Core Test Explorer](#)
  - [Instant Markdown](#)
  - [npm Intellisense](#)
  - [Partial Diff](#)

- [Paste JSON as Code](#)
- [Path IntelliSense](#)
- [Power Tools](#)
- [Project Manager](#)
- [REST Client](#)
- [Remote VSCode](#)
- [Settings Sync](#)
- [Todo Tree](#)
- [Toggle Quotes](#)
- [WakaTime](#)
- [Yo](#)
- [Formatting/Beautification](#)
  - [Better Align](#)
  - [Auto Close Tag](#)
  - [Auto Rename Tag](#)
  - [beautify](#)
  - [html2pug](#)
  - [ECMAScript Quotes Transformer](#)
  - [Paste and Indent](#)
  - [Sort Lines](#)
  - [Surround](#)
  - [Wrap Selection](#)
  - [Formatting Toggle](#)
  - [Bracket Pair Colorizer](#)
  - [Auto Import](#)
  - [shell-format](#)
  - [Vscode Google Translate](#)
- [Explorer Icons](#)
  - [City Lights Icons](#)
  - [VSCode Icons](#)
  - [Seti-UI](#)
  - [Material-Icon-Theme](#)
- [Uncategorized](#)
  - [Browser Preview](#)
  - [Carbon-now-sh](#)
  - [Code Runner](#)
  - [Code Time](#)
  - [Color Highlight](#)
  - [Output Colorizer](#)
  - [Dash](#)
  - [Debugger for Chrome](#)
  - [ECMAScript Quotes Transformer](#)
  - [Edit with Shell Command](#)
  - [Editor Config for VS Code](#)
  - [ftp-sync](#)
  - [Highlight JSX/HTML tags](#)

- [Indent Rainbow](#)
- [PlatformIO](#)
- [Polacode](#)
- [Quokka](#)
- [Remote Workspace](#)
- [Runner](#)
- [Slack](#)
- [SmoothType](#)
- [Spotify](#)
- [SVG](#)
- [SVG Viewer](#)
- [Text Marker \(Highlighter\)](#)
- [ESDOC MDN](#)
- [Interface generator](#)
- Themes
  - [An Old Hope Theme by Dustin Sanders](#)
  - [Ariake Dark by wart](#)
  - [Atom One Dark Theme by Mahmoud Ali](#)
  - [Ayu by teabyii](#)
  - [Captain Sweetheart by ultradracula](#)
  - [City Lights Theme by Yummygum](#)
  - [Cobalt2 Theme Official by Wes Bos](#)
  - [Dracula Official by Dracula Theme](#)
  - [Edge by Bogdan Lazar](#)
  - [Eva Theme by fisheva](#)
  - [Fairyfloss by nopjmp and sailorhg](#)
  - [GitHub Theme by Thomas Pink](#)
  - [Material Palenight Theme by whizkydee](#)
  - [Material Theme by Mattia Astorino](#)
  - [Mno by u29dc](#)
  - [Monokai Oblique](#)
  - [Monokai Pro by monokai \(commercial\)](#)
  - [Night Owl by Sarah Drasner](#)
  - [Plastic by Will Stone](#)
  - [Niketa Theme by Dejan Toteff](#)
  - [Nord by arcticicestudio](#)
  - [Rainglow by Dayle Rees](#)
  - [Relaxed Theme by Michael Kühnel](#)
  - [Shades of Purple by Ahmad Awais](#)
  - [Slime Theme by smlombardi](#)
- People to Follow
- Resources for extension developers
  - [Documentation](#)
  - [Libraries](#)
  - [Tools](#)
- Online Courses

- Visual Studio Code Power User Course (commercial)
- Contribute
- License

## Official

---

- Official website
- Source code on GitHub
- Releases (stable channel)
- Releases (insiders channel)
- Monthly iteration plans

## Syntax

---

Language packages extend the editor with syntax highlighting and/or snippets for a specific language or file format.

- Arduino
- Befunge
- Blink
- Bolt
- Bond
- CMake
- Dart
- Dockerfile
- EJS
- Elixir
- Elm
- Erlang
- F#
- Flatbuffers
- Fortran
- Hack(HHVM)
- Handlebars
- Hive SQL
- KL
- Kotlin
- LaTeX
- Mason
- openHAB
- Parser 3
- Pascal, or OmniPascal (only for Windows)
- Perl HTML-Template
- Protobuf
- Ruby

- [Scala](#)
- [Stylus](#)
- [Swift](#)
- [VEX](#)
- [Zephir](#)

## Migrating from other editors

---

The VSCode team provides keymaps from popular editors, making the transition to VSCode almost seamless and easy.

### [Migrating from Vim](#)

**Vim Mode** - Relatively new, but promising extension implementing Vim features in VSCode

### [Migrating from Atom](#)

Popular Atom key bindings for Visual Studio Code

### [Migrating from Sublime Text](#)

Popular Sublime Text key bindings for VS Code.

### [Migrating from Visual Studio](#)

Popular Visual Studio key bindings for VS Code.

### [Migrating from IntelliJ IDEA](#)

Popular IntelliJ IDEA key bindings for VS Code.

## Using VS Code with particular technologies

---

Microsoft created a collection of recipes for using VS Code with particular technologies (mostly Web).

Make sure to visit it at [Microsoft/vscode-recipes](#)

## Lint and IntelliSense

---

In case the awesome nirvana that is linting has not yet been unleashed upon you:

lint was the name originally given to a particular program that flagged some suspicious and non-portable constructs (likely to be bugs) in C language source code. The term is now applied generically to tools that flag suspicious usage in software written in any computer language.

Unlike some other editors, VS Code supports IntelliSense, linting, outline out-of-the-box and doesn't require any separate extension to run linter packages. Some linters are already integrated in VS Code, you can find the full list in the official documentation, [Languages](#) section.

## 1C

- [1C/OScript](#) - rich 1C:Enterprise 8 (BSL) language support in VSC - add syntax highlighting to \*.bsl и \*.os files in VSC, add IntelliSense and syntax helper for 1C lang

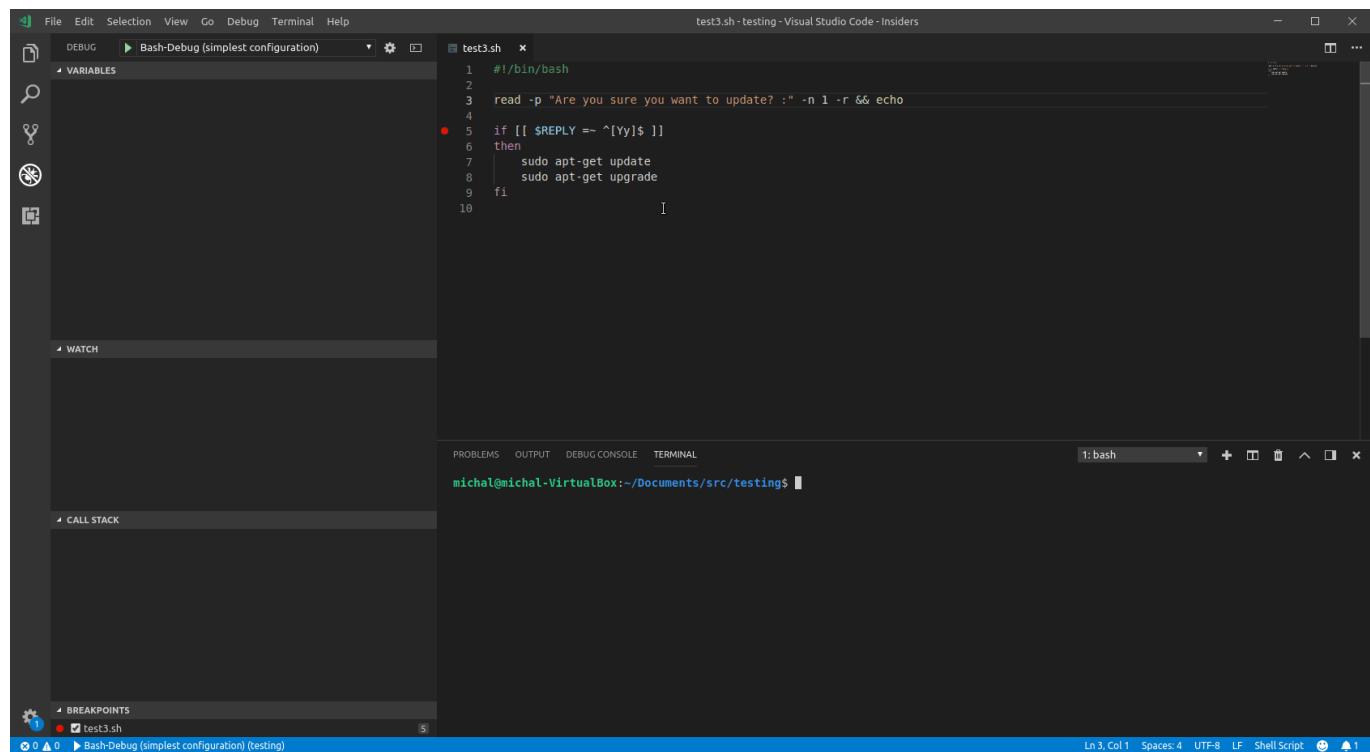
## Bash

### Bash IDE

A language server for Bash

### Bash Debug

A debugger extension for Bash scripts based on [bashdb](#)

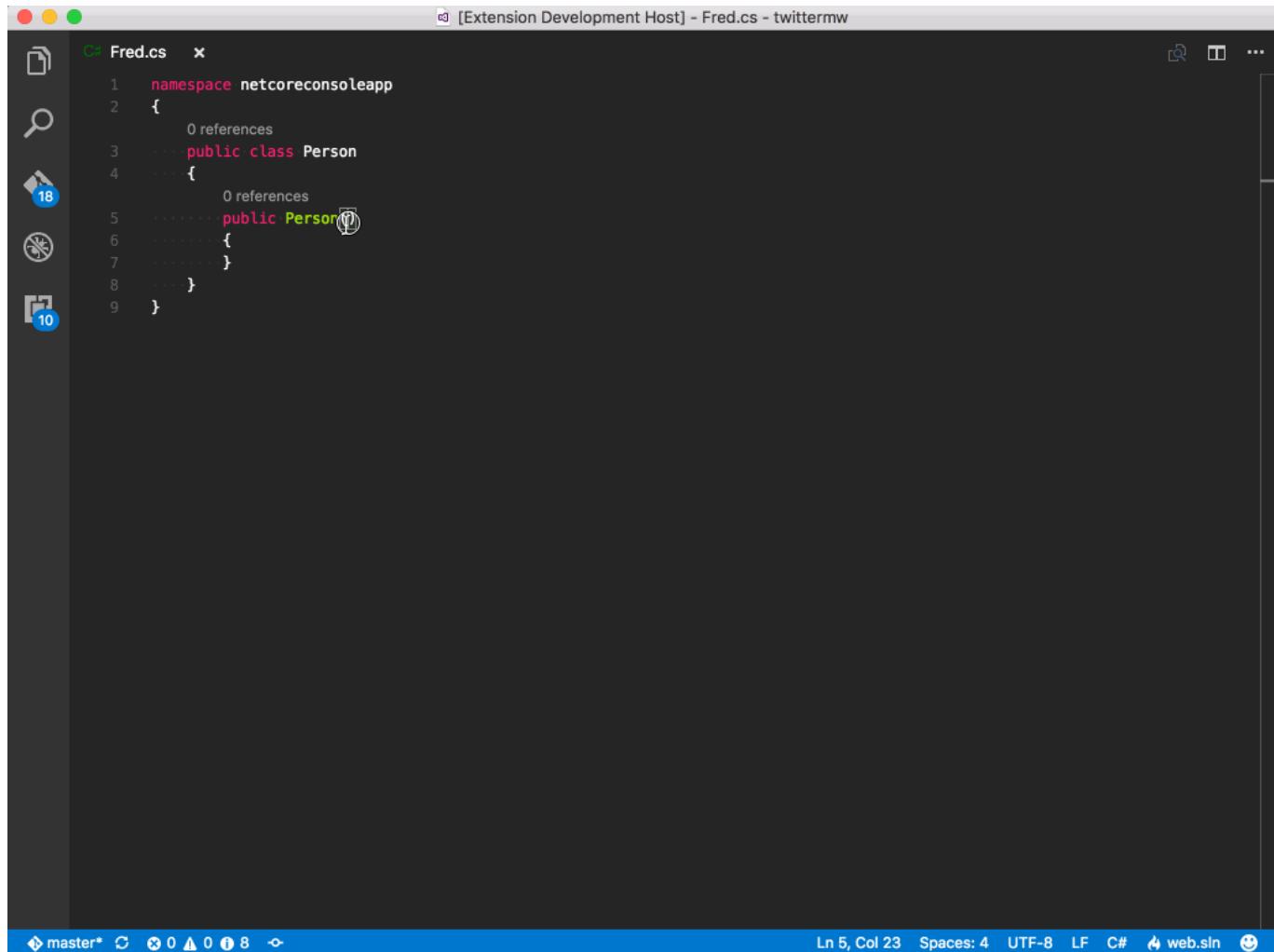


## C++

- [C/C++](#) - Preview C/C++ extension by Microsoft, read [official blog post](#) for the details
- [gnu-global-tags](#) - Provide Intellisense for C/C++ with the help of the GNU Global tool.
- [YouCompleteMe](#) - Provides semantic completions for C/C++ (and TypeScript, JavaScript, Objective-C, Golang, Rust) using [YouCompleteMe](#).
- [C/C++ Clang Command Adapter](#) - Completion and Diagnostic for C/C++/Objective-C using Clang command.
- [CQuery - C/C++ language server](#) supporting multi-million line code base, powered by libclang. Cross references, completion, diagnostics, semantic highlighting and more.

## C#

- [C#](#) - C# extension by Microsoft, read [official documentation](#) for the details
- [C# FixFormat](#) - Fix format of usings / indents / braces / empty lines
- [C# Extensions](#) - Provides extensions to the IDE that will speed up your development workflow.

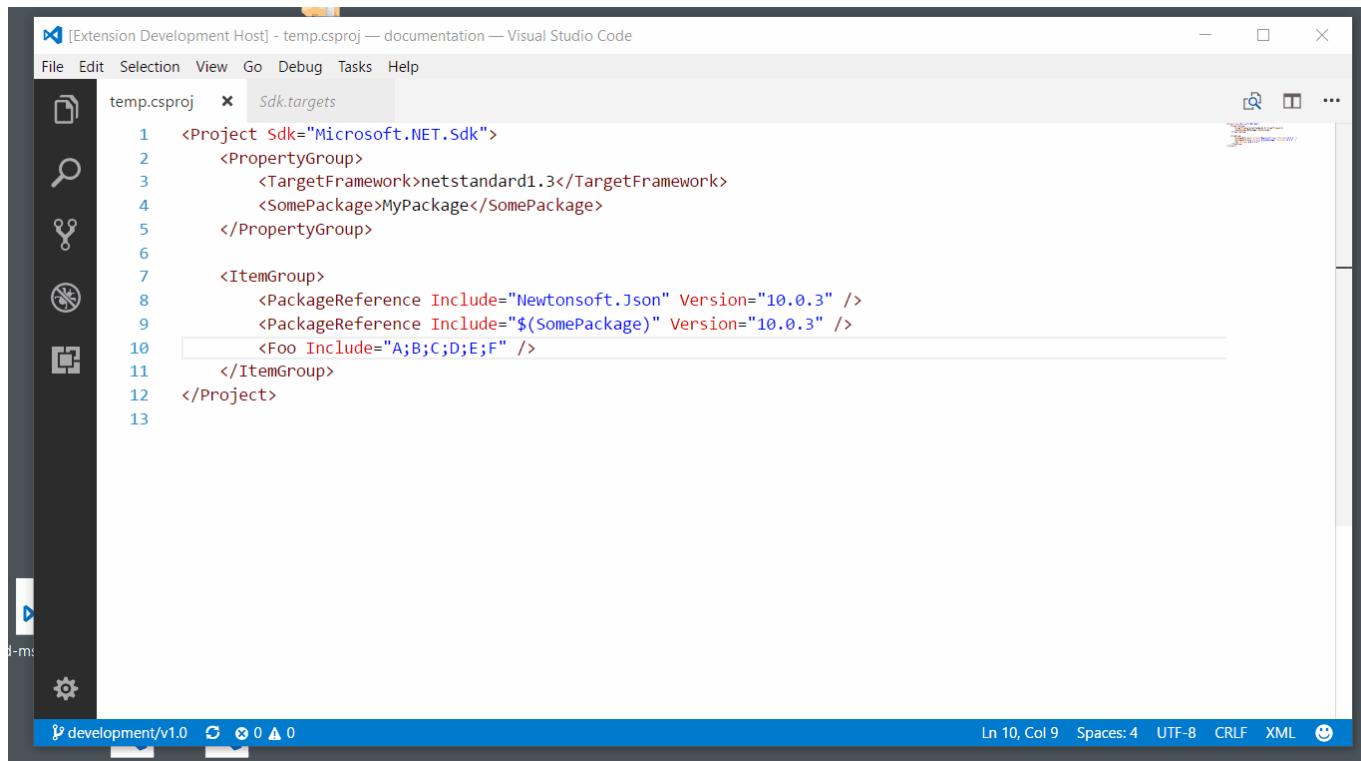


The screenshot shows the Visual Studio Code interface in dark mode. The main area displays a C# file named Fred.cs with the following code:

```
1 namespace netcoreconsoleapp
2 {
3     public class Person
4     {
5         public Person()
6         {
7     }
8 }
```

The sidebar on the left contains icons for file operations like Open, Save, Find, and others, with counts of 18, 10, and 1 respectively. The status bar at the bottom shows the current file path as master\*, line 5, column 23, spaces: 4, encoding: UTF-8, LF, C#, and a file count of 4.

- [MSBuild Project Tools](#)



The screenshot shows the Visual Studio Code interface with the title bar "[Extension Development Host] - temp.csproj — documentation — Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Debug, Tasks, and Help. The left sidebar has icons for file operations like Open, Save, Find, and Settings. The main editor area displays the following XML code:

```
1 <Project Sdk="Microsoft.NET.Sdk">
2   <PropertyGroup>
3     <TargetFramework>netstandard1.3</TargetFramework>
4     <SomePackage>MyPackage</SomePackage>
5   </PropertyGroup>
6
7   <ItemGroup>
8     <PackageReference Include="Newtonsoft.Json" Version="10.0.3" />
9     <PackageReference Include="$(SomePackage)" Version="10.0.3" />
10    <Foo Include="A;B;C;D;E;F" />
11  </ItemGroup>
12 </Project>
13
```

The status bar at the bottom shows "development/v1.0" and "Ln 10, Col 9 Spaces: 4 UTF-8 CRLF XML".

## Clojure

### Calva

Integrated REPL, linting, inline eval, test runner, and more. Powered by Cider & nRepl.



## CSS

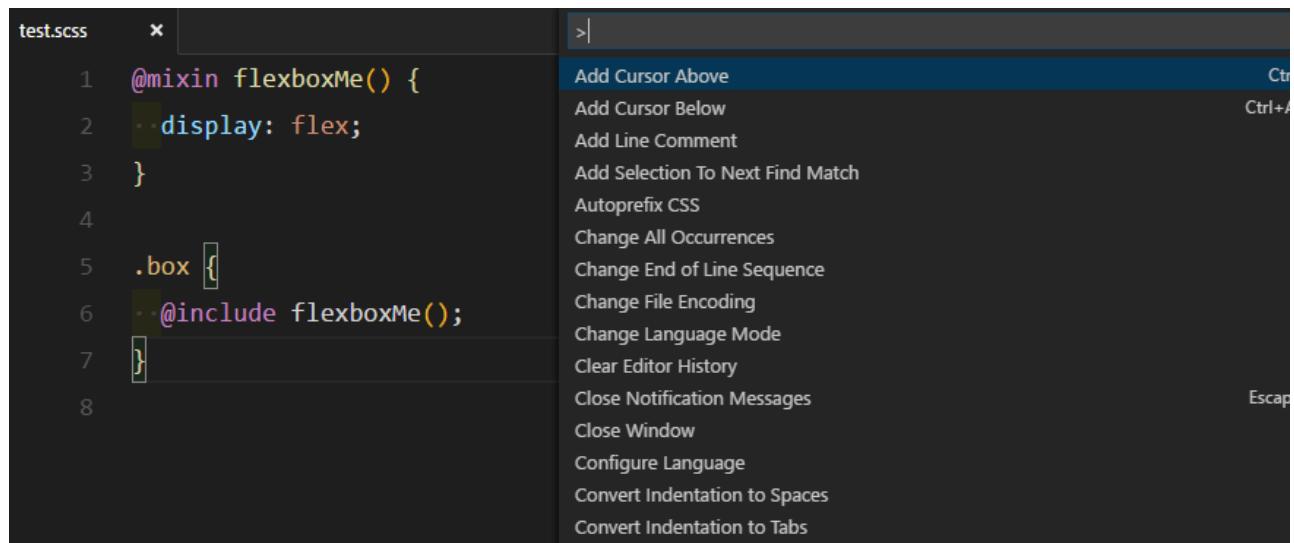
### CSS Peek

Peek or Jump to a CSS definition directly from HTML, just like in Brackets!



- [stylelint](#) - Lint CSS/SCSS.

- [Autoprefixer](#) Parse CSS,SCSS, LESS and add vendor prefixes automatically.



- [Intellisense for CSS class names](#) - Provides CSS class name completion for the HTML class attribute based on the CSS files in your workspace. Also supports React's className attribute.

## Go

- [Go](#) - Rich language support for the Go language.

## Haskell

- [haskell-linter](#)
- [Haskell IDE engine](#) - provides [language server](#) for stack and cabal projects.

## Shell

- [autocomplete-shell](#)

## Java

- [Language Support for Java\(TM\) by Red Hat](#)
- [Debugger for Java](#)
- [Maven for Java](#)
- [Lombok](#)

## JavaScript

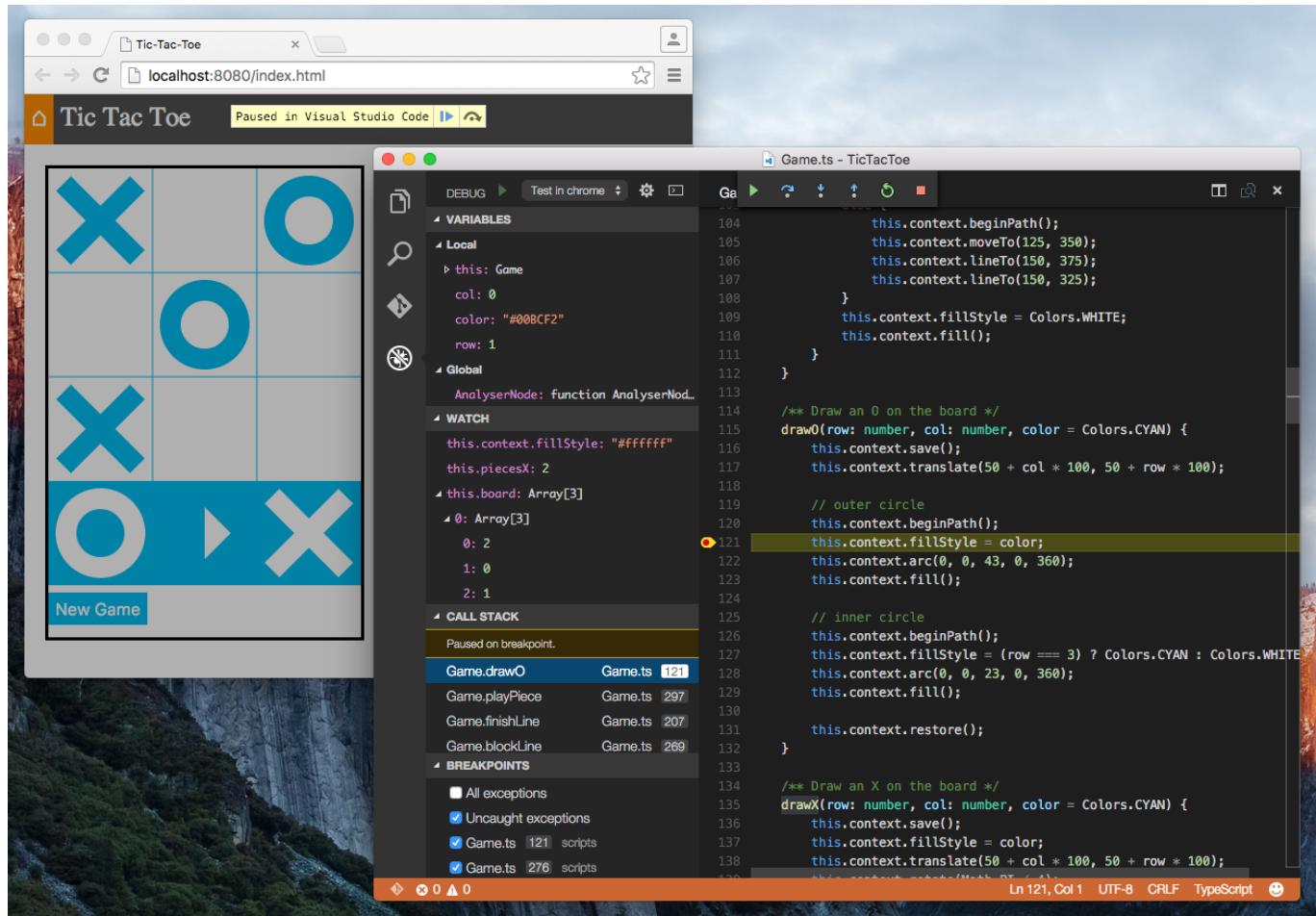
- [Babel JavaScript](#)
- [Sublime Babel](#) - Babel extension ported from Sublime Text

See the difference between these two [here](#)

- [tslint](#) - TSLint for Visual Studio Code (with "tslint.jsEnable": true).
- [eslint](#) - Linter for [eslint](#).
- [XO](#) - Linter for [XO](#).
- [AVA](#) - Snippets for [AVA](#).
- [Prettier](#) - Linter, Formatter and Pretty printer for [Prettier](#).
- [Schema.org Snippets](#) - Snippets for [Schema.org](#).

## Debugger for Chrome

A VS Code extension to debug your JavaScript code in the Chrome browser, or other targets that support the Chrome Debugging Protocol.



## TypeScript

- [tslint](#) - TSLint for Visual Studio Code
- [TypeScript Hero](#) - Code outline view of your open TS, sort and organize your imports.

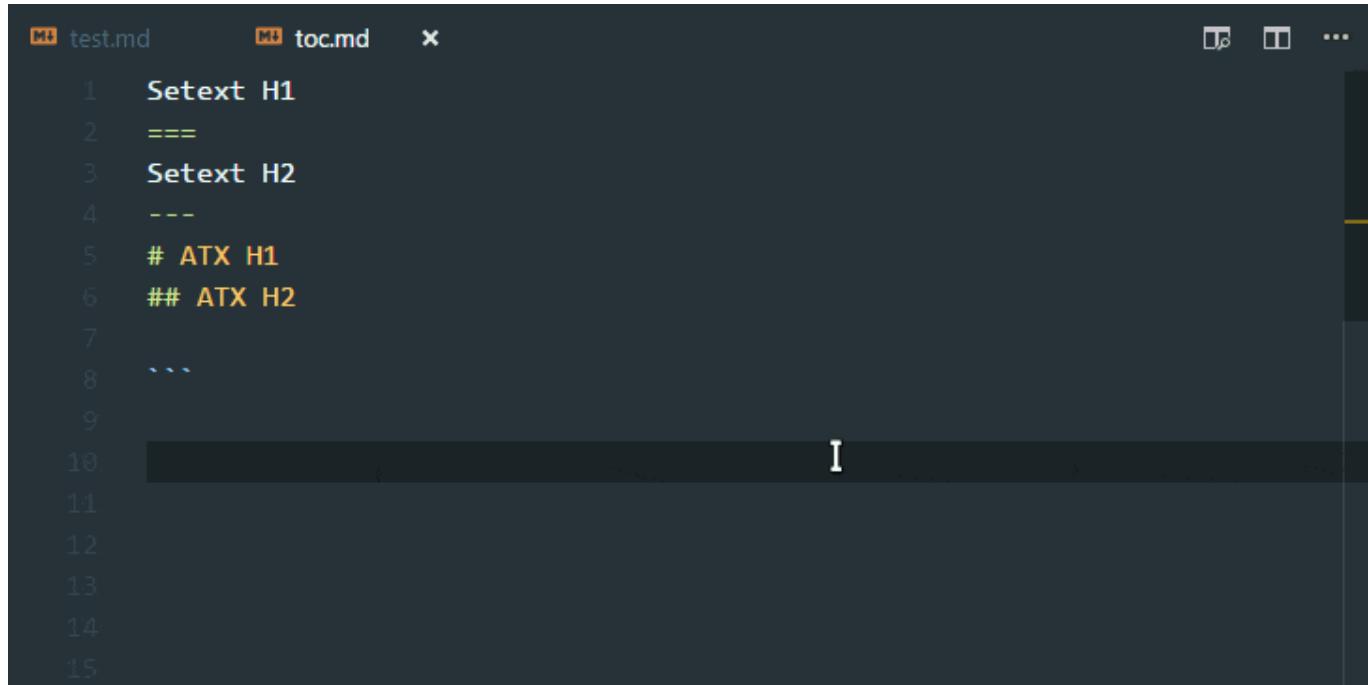
## Markdown

### [markdownlint](#)

Linter for [markdownlint](#).

### [Markdown All in One](#)

All-in-one markdown plugin (keyboard shortcuts, table of contents, auto preview, list editing and more)



```
1 Setext H1
2 ===
3 Setext H2
4 ---
5 # ATX H1
6 ## ATX H2
7
8
9
10
11
12
13
14
15
```

## PHP

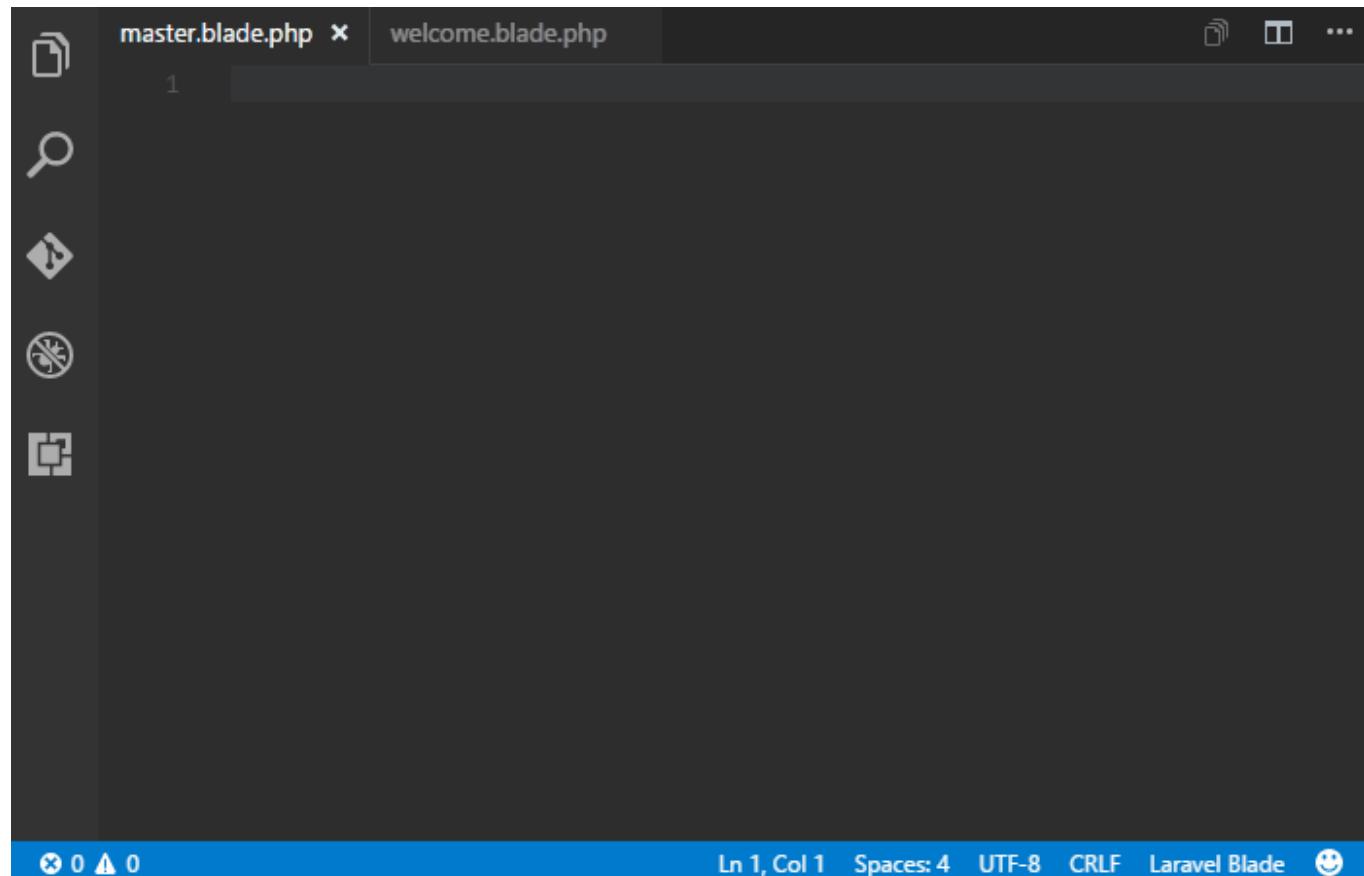
### IntelliSense

These extensions provide slightly different sets of features. While the first one offers better autocompletion support, the second one seems to have more features overall.

- [PHP Intelephense](#)
- [PHP IntelliSense](#)

### Laravel

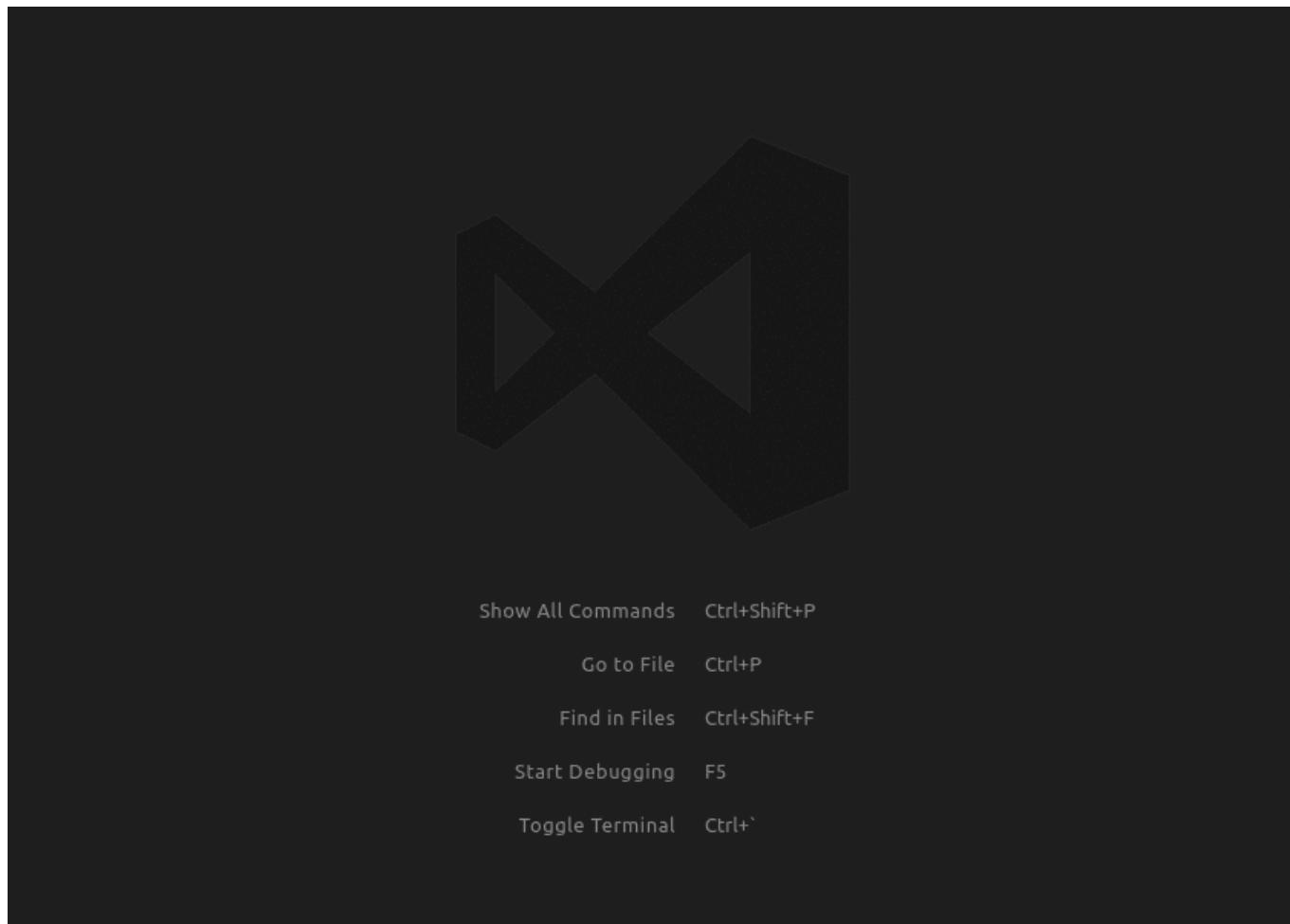
- [Laravel 5 Snippets](#) - Laravel 5 snippets for Visual Studio Code
- [Laravel Blade Snippets](#) - Laravel blade snippets and syntax highlight support



- [Laravel Model Snippets](#) - Quickly get models up and running with Laravel Model Snippets.

```
10
    Unsaved changes (cannot determine recent change or authors)
11 class User extends Authenticatable
12 {
13     |
14
15 }
16
```

- [Laravel Artisan](#) - Laravel Artisan commands within Visual Studio Code



- [DotENV](#) - Support for dotenv file syntax

```
1 # Keyword
2 export TEST_KEYWORD="bar"
3 export TEST_KEYWORD=12345
4 export TEST_KEYWORD=TRUE
5
6 # Variable
7 TEST_VARIABLE="Hello"
8
9 # String interpolation
10 TEST_INTERPOLATION_VARIABLE="$VAR1 test test$VAR2test test"
11 TEST_INTERPOLATION_SYNTAX_ONE="test test${$NVAR1}test${$NVAR2}test test"
12 TEST_INTERPOLATION_SYNTAX_TWO="test test${$NVAR1}test${$NVAR2}test test"
13 TEST_INTERPOLATION_SYNTAX_ALL="test$VAR1test test ${VAR2}test test${$VAR3}test"
14
15 # Unquoted
16 TEST_UNQUOTED=bar
17 TEST_UNQUOTED_NO_VALUE=
18
19 # White spaced
20 TEST_WHITE_SPACE =
21 TEST_WHITE_SPACE_STRING = "Hello"
22 TEST_WHITE_SPACE_UNQUOTED = bar
23 TEST_WHITE_SPACE_UNQUOTED_BOOL = false
24 TEST_WHITE_SPACE_UNQUOTED_NUM = 20
25
26 # language constants
27 TEST_TRUE=true
```

```

28 TEST_FALSE=false
29 TEST_NULL=null
30 TEST_TRUE_CAPITAL=TRUE
31 TEST_FALSE_CAPITAL=FALSE
32 TEST_NULL_CAPITAL=NULL
33
34 # Numerical values
35 TEST_NUM_DECIMAL=54
36 TEST_NUM_FLOAT=5.3
37 TEST_NUM=1e10
38 TEST_NUM_NEGATIVE=-42
39 TEST_NUM_OCTAL=057
40 TEST_NUM_HEX=0x1A
41
42 # Comments
43 #TEST_ONE=foobar
44 # TEST_TWO='foobar'
45 # TEST_THREE="foobar" # a comment on a commented row
46 TEST_FOUR="test test test" # this is a comment
47 TEST_FIVE="comment symbol # inside string" # this is a comment
48 TEST_SIX="comment symbol # and quotes \" \' inside quotes" # " this is a comment
49
50 # Escape sequences
51 TEST_ESCAPE="escaped characters \n \t \r \" \' \$ or maybe a backslash \\..."
52
53 # Double Quotes
54 TEST_DOUBLE="Lorem {$VAR1} ${VAR2} $VAR3 ipsum dolor sit amet\n\r\t\\"
55
56 # Single Quotes
57 TEST_SINGLE='Lorem {$VAR1} ${VAR2} $VAR3 ipsum dolor sit amet\n\r\t\\'

```

## Twig

- [Twig Language 2](#)

To enable Emmet support in .twig files, you'll need to have the following in your settings:

```
{
  "emmet.includeLanguages": {
    "twig": "html"
  }
}
```

## Other extensions

- [Format HTML in PHP](#) - Formatting for the HTML in PHP files. Runs before the save action so you can still have a PHP formatter.

```

1  <?php
2  /**
3   * Format On Save
4   * Ctrl + Alt + F  Format Anytime
5   *
6   * @package format-html-in-php
7   */
8
9 ?>
10
11 <head>
12   <meta charset="utf-8">
13   <meta http-equiv="X-UA-Compatible" content="IE=edge">
14   <meta name="viewport" content="width=device-width, initial-scale=1">
15   <title>Format HTML in PHP</title>
16   <?php wp_head(); ?>
17 </head>
18 <body>
19   <div class="modal modal-fixed-footer">
20     <form method="post">
21       <div class="col s12">
22         <h3 class="red-after">
23           <?php esc_html_e( 'Format HTML', 'in PHP' ); ?>
24         </h3>
25       </div>
26       <div class="col s12">
27         <h5>
28           <?php esc_html_e( 'Choose Own PHP Formatting Ex', 'Format HTML' ); ?>
29         </h5>
30       </div>
31
32       <div class="input-field col s12" style="margin-bottom:0">
33         <?php $field = 'create_campaign_name'; ?>

```

- [Composer](#)
- [PHP Debug](#) - XDebug extension for Visual Studio Code
- [PHP DocBlocker](#)
- [php cs fixer](#) - PHP CS Fixer extension for VS Code, php formatter, php code beautify tool
- [phpcs](#) - PHP CodeSniffer for Visual Studio Code
- [phpfmt](#) - phpfmt for Visual Studio Code

## Read more

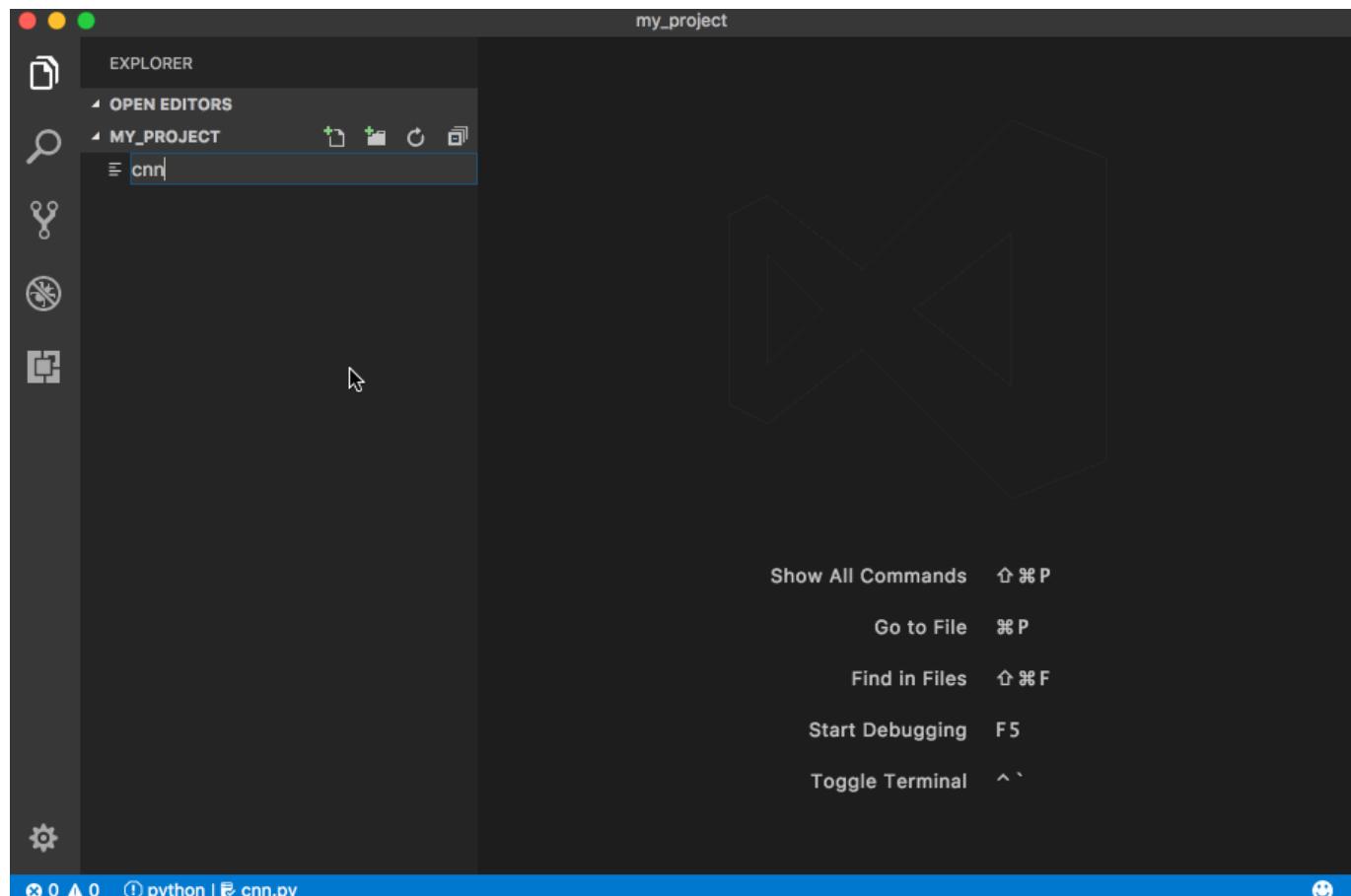
- [Configuring Visual Studio Code for Drupal](#)

## Python

- [Python](#) - Linting, Debugging (multi threaded, web apps), Intellisense, auto-completion, code formatting, snippets, unit testing, and more.

## TensorFlow

- [TensorFlow Snippets](#) - This extension includes a set of useful code snippets for developing TensorFlow models in Visual Studio Code.



## ReasonML

- [ReasonML](#) - Intellisense, code formatting, refactoring, code lens and more

## Rust

- [Rust](#) - Linting, auto-completion, code formatting, snippets and more

## Terraform

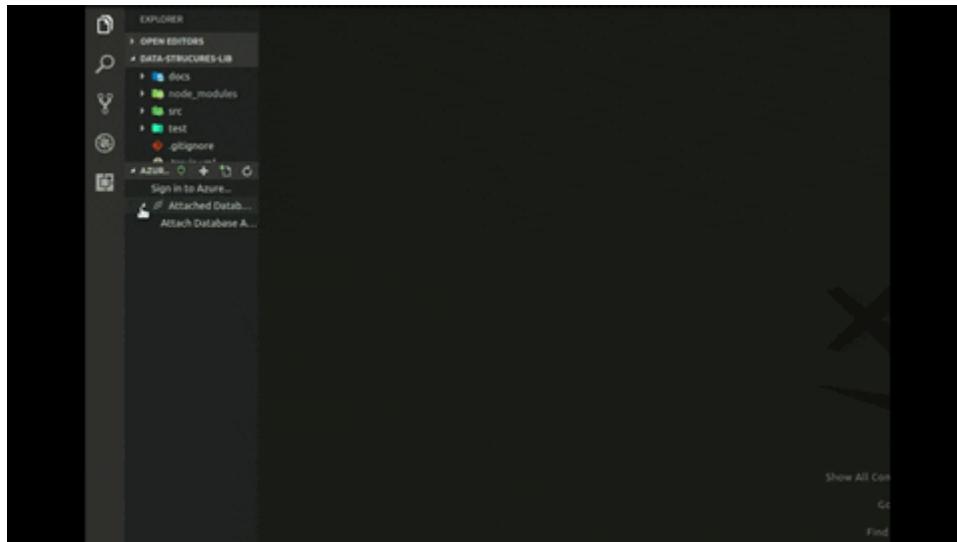
- [Terraform](#) - Syntax highlighting, linting, formatting, and validation for Hashicorp's Terraform

## Productivity

---

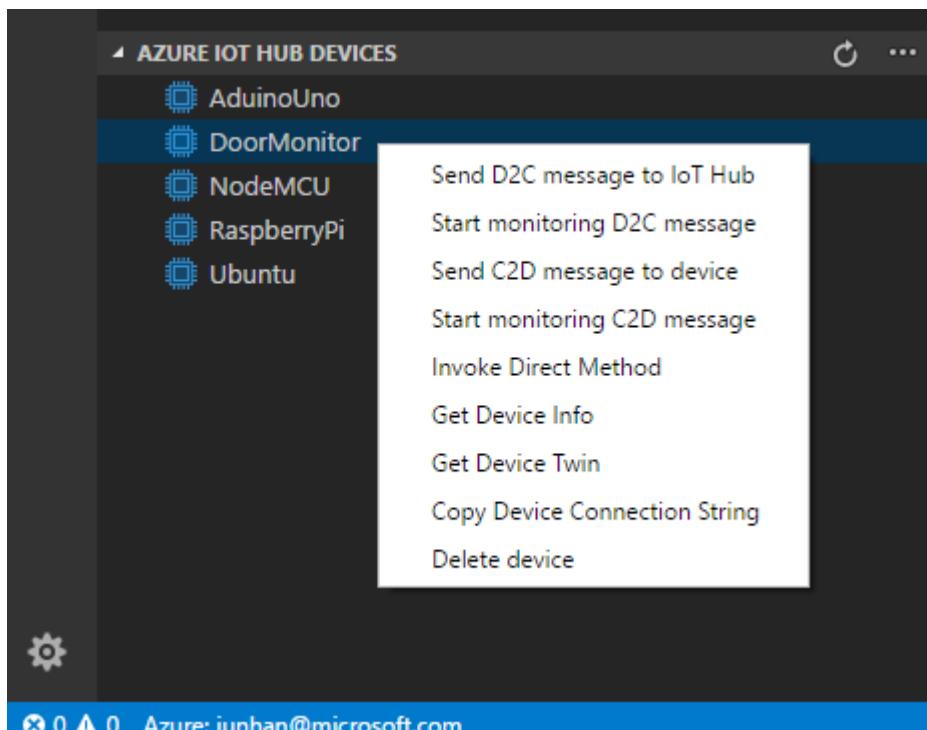
### Azure Cosmos DB

Browse your database inside the vs code editor



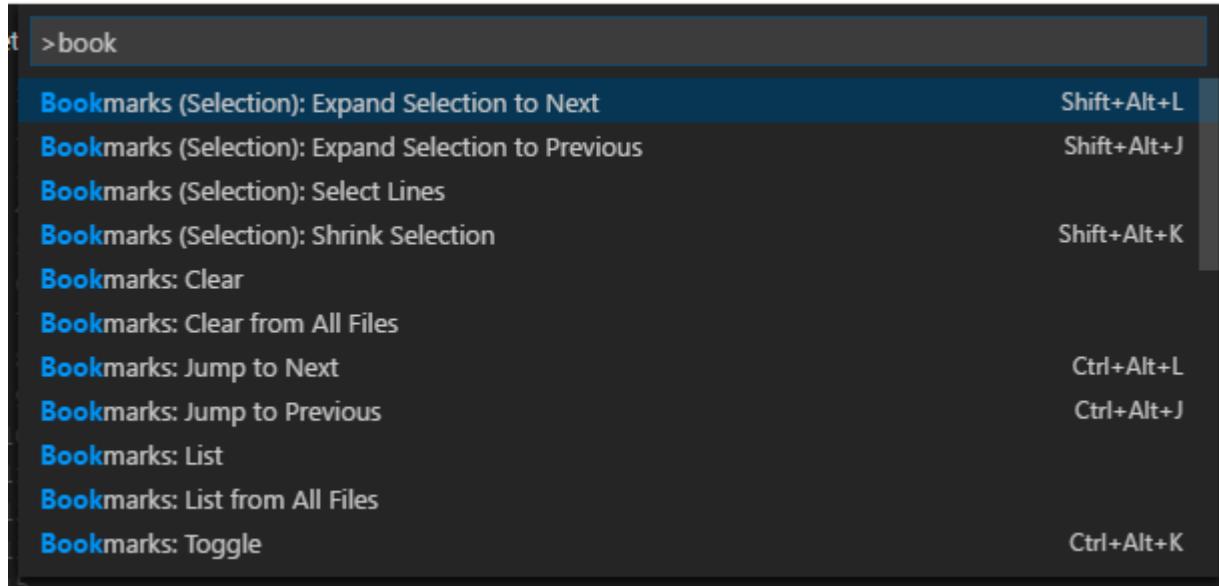
## Azure IoT Toolkit

Everything you need for the Azure IoT development: Interact with Azure IoT Hub, manage devices connected to Azure IoT Hub, and develop with code snippets for Azure IoT Hub



## Bookmarks

Mark lines and jump to them

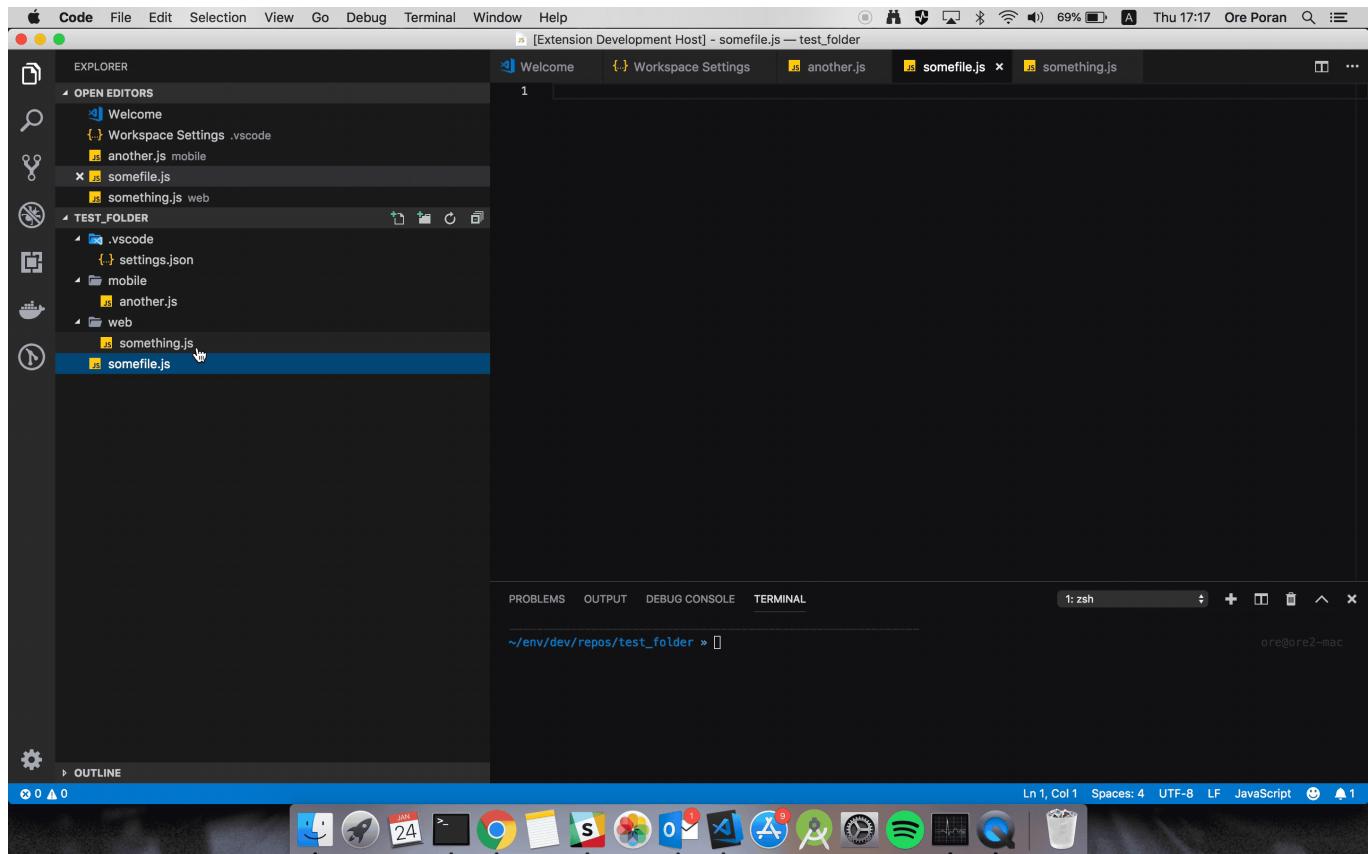


The screenshot shows the 'extension.ts' file open in the VS Code editor. The code is as follows:

```
extension.ts C:\Users\Alessandro\Documents\GitHub\vscode-bookmarks
5 import ts = require('ts');
6
7 // this method is called when vs code is activated
8 export function activate(context: vscode.ExtensionContext) {
9
10    class Bookmark {
11        fsPath: string;
12        bookmarks: number[];
13
14        constructor(uri: vscode.Uri) {
15            this.fsPath = uri.fsPath;
16            this.bookmarks = [];
17        }
18    }
19
20    class Bookmarks {
21        bookmarks: Bookmark[];
22    }
```

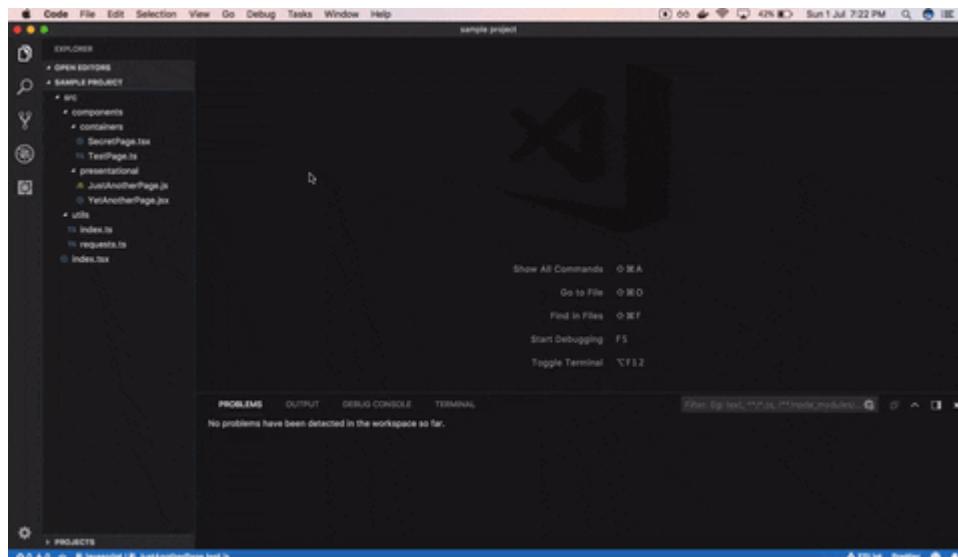
## Color Tabs

An extension for big projects or monorepos that colors your tab/titlebar based on the current package



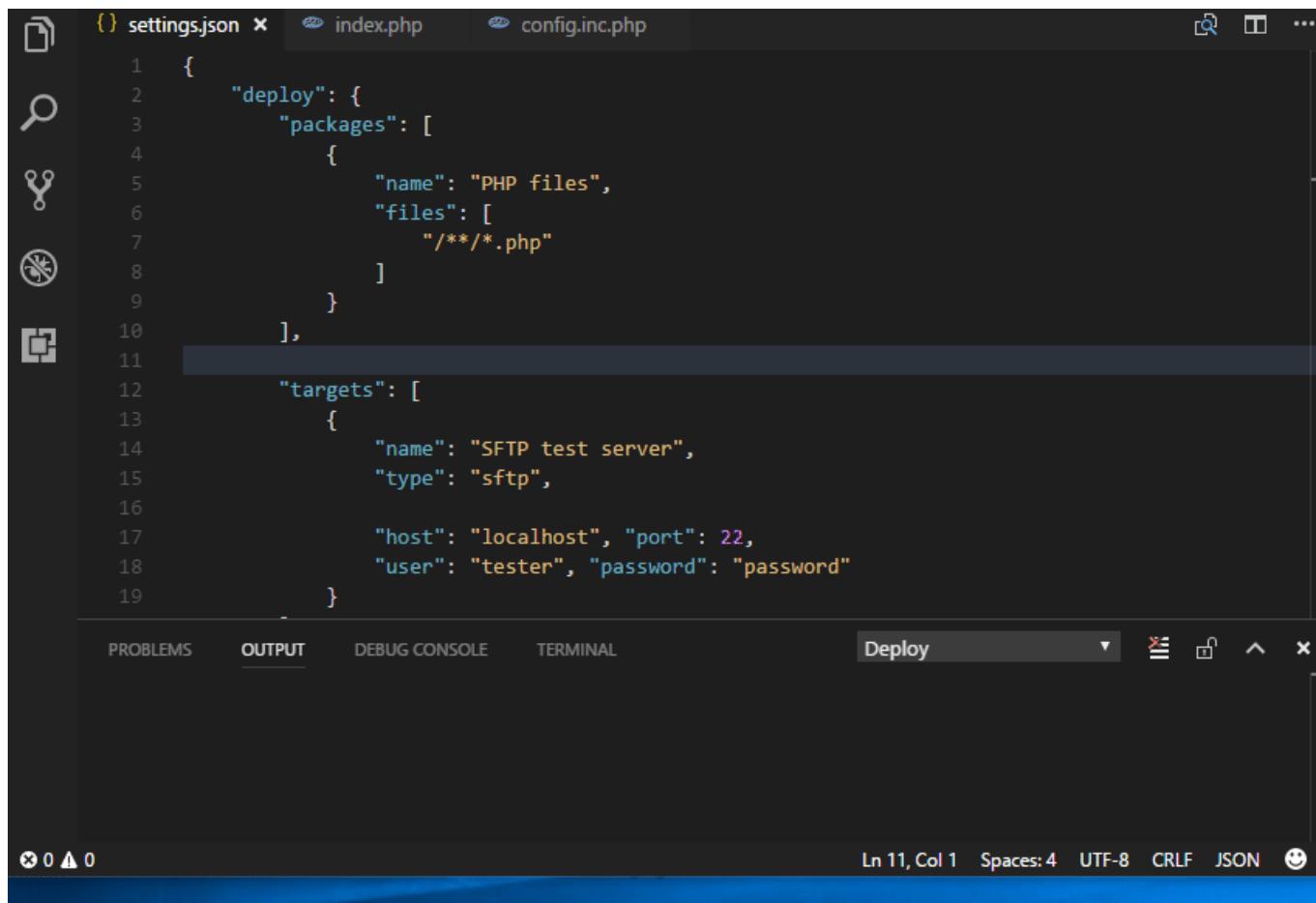
## Create tests

An extension to quickly generate test files.



## Deploy

Commands for upload or copy files of a workspace to a destination.



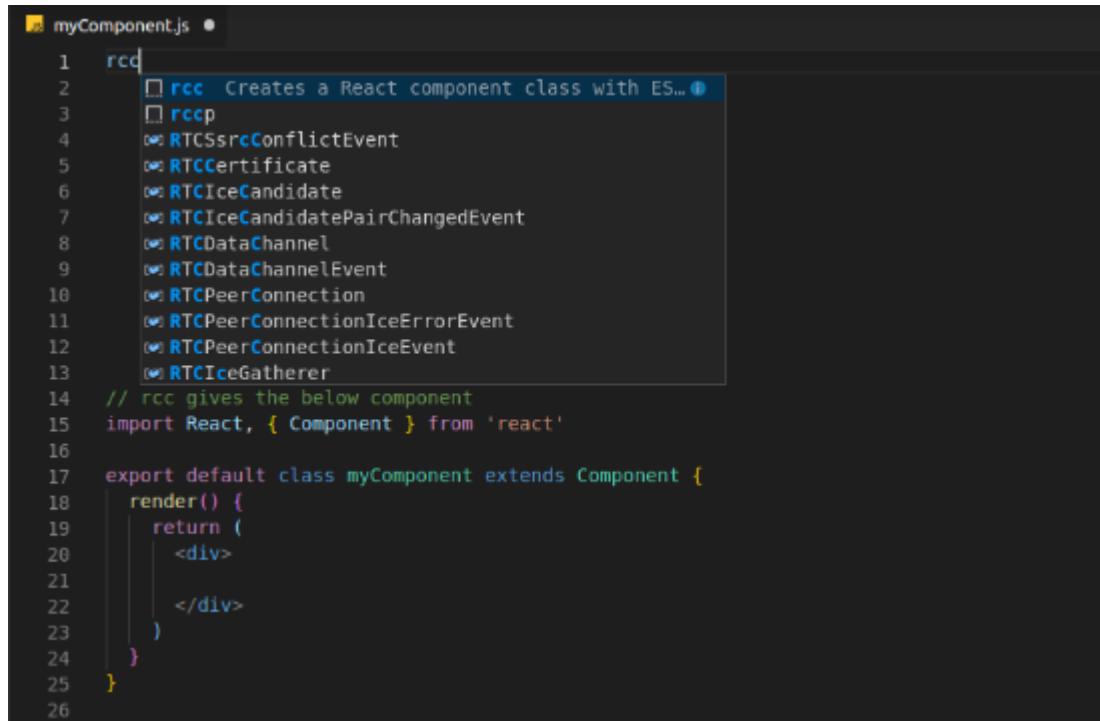
```
{} settings.json x  index.php config.inc.php ...  
1 {  
2     "deploy": {  
3         "packages": [  
4             {  
5                 "name": "PHP files",  
6                 "files": [  
7                     "**/*.php"  
8                 ]  
9             }  
10        ],  
11  
12        "targets": [  
13            {  
14                "name": "SFTP test server",  
15                "type": "sftp",  
16  
17                "host": "localhost", "port": 22,  
18                "user": "tester", "password": "password"  
19            }  
20        ]  
21    }  
22}  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL Deploy ▾      
Ln 11, Col 1 Spaces: 4 UTF-8 CRLF JSON 😊
```

## Duplicate Action

Ability to duplicate files and directories.

## ES7 React/Redux/GraphQL/React-Native snippets

Provides Javascript and React/Redux snippets in ES7



A screenshot of the VSCode interface showing a code completion dropdown for the variable 'rcc'. The dropdown lists several options starting with 'rcc', such as 'Creates a React component class with ES...', 'rccp', 'RTCsrcConflictEvent', 'RTCCertificate', etc. The main code editor shows a file named 'myComponent.js' with the following content:

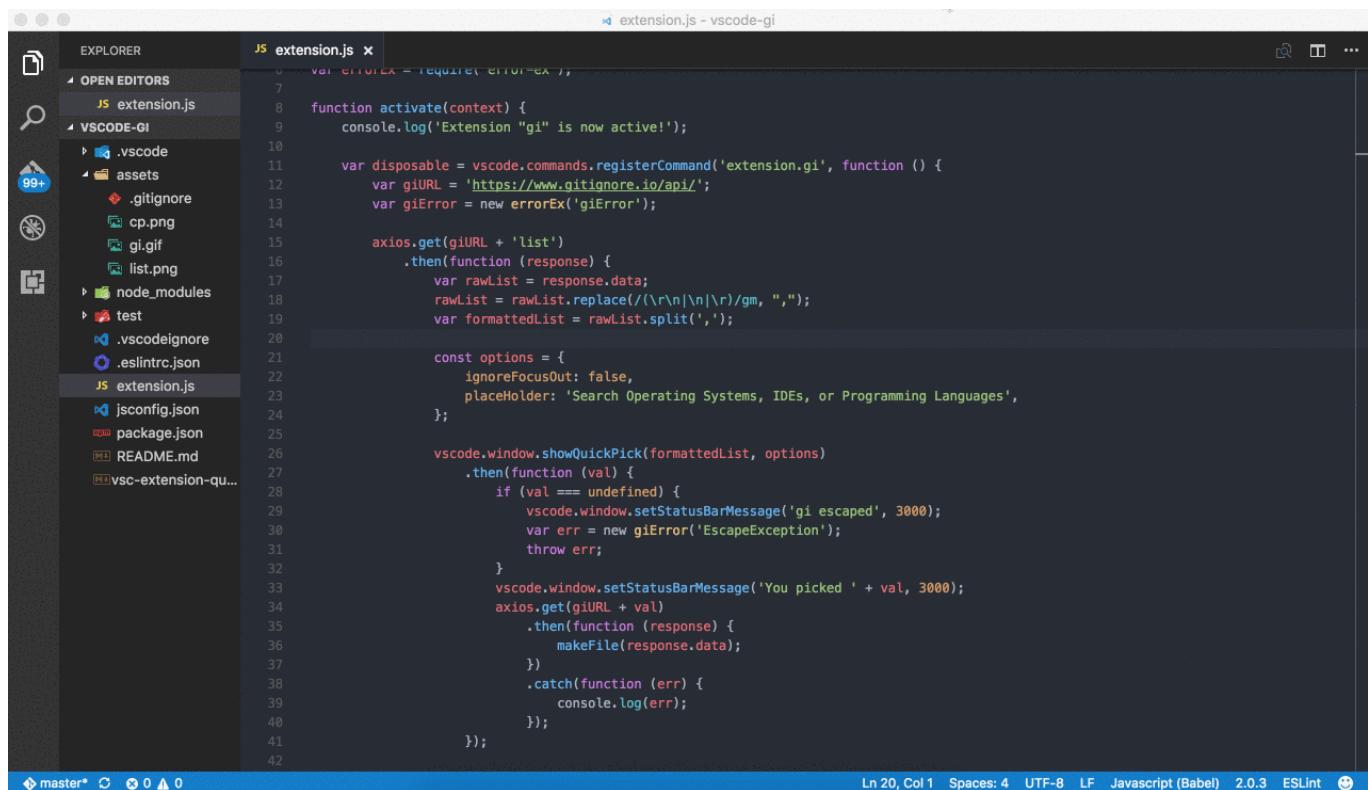
```

1  rcc
2  □ rcc Creates a React component class with ES...
3
4
5
6
7
8
9
10
11
12
13
14 // rcc gives the below component
15 import React, { Component } from 'react'
16
17 export default class myComponent extends Component {
18   render() {
19     return (
20       <div>
21
22       </div>
23     )
24   }
25 }
26

```

## Gi

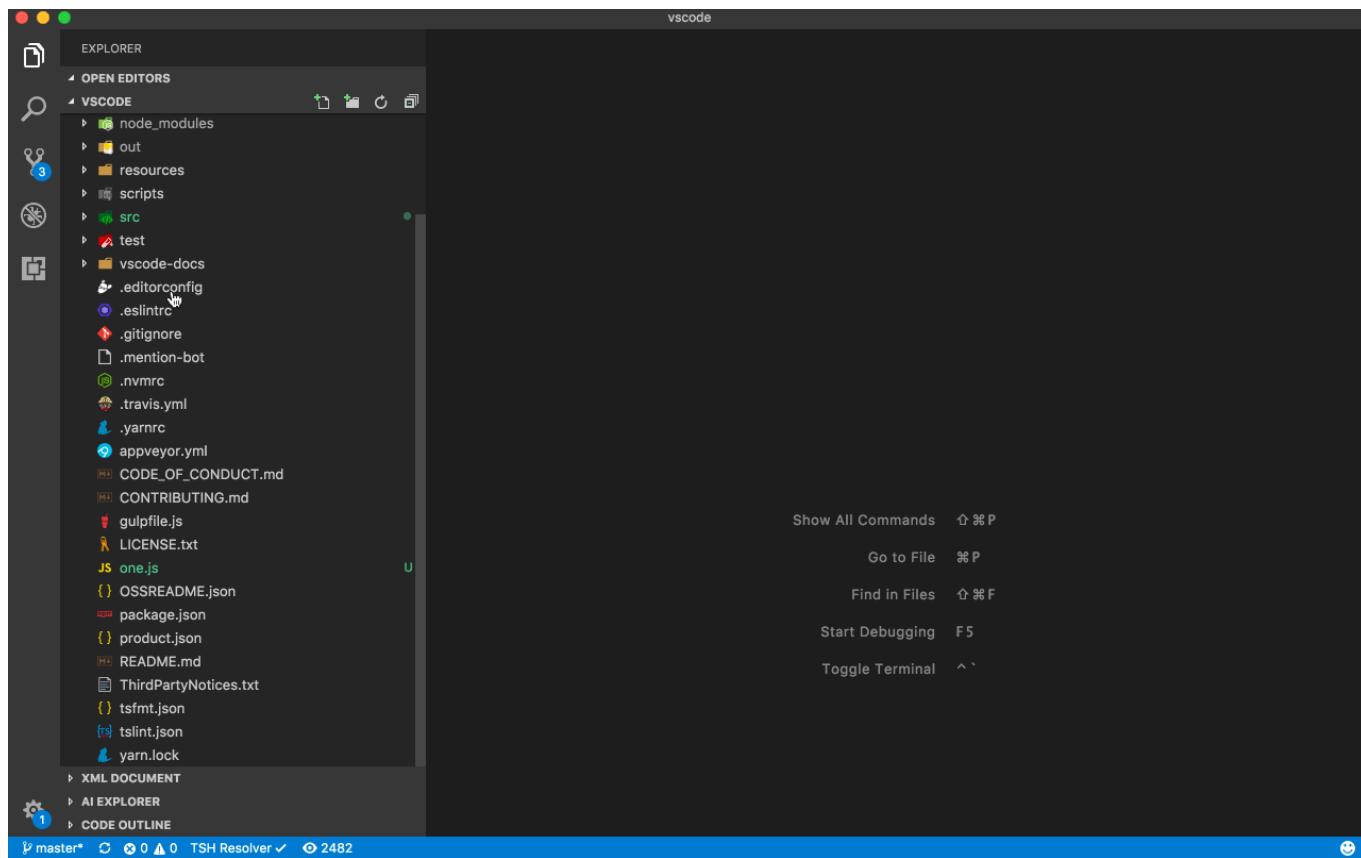
Generating .gitignore files made easy.



A screenshot of the VSCode interface showing the 'extension.js - vscode-gi' editor tab. The code in the editor is as follows:

```

1  var errorEx = require('error-ex');
2
3
4
5
6
7
8  function activate(context) {
9    console.log('Extension "gi" is now active!');
10
11  var disposable = vscode.commands.registerCommand('extension.gi', function () {
12    var giURL = 'https://www.gitignore.io/api/';
13    var giError = new errorEx('giError');
14
15    axios.get(giURL + 'list')
16      .then(function (response) {
17        var rawList = response.data;
18        rawList = rawList.replace(/(\r\n|\n|\r)/gm, ",");
19        var formattedList = rawList.split(',');
20
21        const options = {
22          ignoreFocusOut: false,
23          placeHolder: 'Search Operating Systems, IDEs, or Programming Languages',
24        };
25
26        vscode.window.showQuickPick(formattedList, options)
27          .then(function (val) {
28            if (val === undefined) {
29              vscode.window.setStatusBarMessage('gi escaped', 3000);
30              var err = new giError('EscapeException');
31              throw err;
32            }
33            vscode.window.setStatusBarMessage('You picked ' + val, 3000);
34            axios.get(giURL + val)
35              .then(function (response) {
36                makeFile(response.data);
37              })
38              .catch(function (err) {
39                console.log(err);
40              });
41          });
42      });
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
187
188
189
189
190
191
192
193
194
195
196
197
197
198
199
199
200
201
202
203
204
205
206
207
207
208
209
209
210
211
212
213
214
215
215
216
217
217
218
218
219
219
220
221
221
222
222
223
223
224
224
225
225
226
226
227
227
228
228
229
229
230
230
231
231
232
232
233
233
234
234
235
235
236
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1
```



## Git Project Manager

Automatically indexes your git projects and lets you easily toggle between them

## GitLink

GoTo current file's online link in browser and Copy the link in clipboard.

```
package.json — vscode-gitlink
{} package.json x ts index.ts
1 {
2   "name": "gitlink",
3   "displayName": "gitlink",
4   "description": "Goto/Copy current file's online link, supports multiple remote sources in GitHub/GitLab/BitBucket/VSTS"
5   "version": "0.4.2",
6   "publisher": "qezhu",
7   "author": {
8     "name": "Qinen Zhu",
9     "email": "qezhu@outlook.com"
10    },
11   "icon": "images/logo.png",
12   "bugs": {
13     "url": "https://github.com/qinezh/vscode-gitlink/issues",
14     "email": "qezhu@outlook.com"
15   },
16   "homepage": "https://github.com/qinezh/vscode-gitlink/blob/master/README.md",
17   "repository": {
18     "type": "git+"
19   }
}
Ln 4, Col 122 (168 selected) Spaces: 2 UTF-8 LF JSON ☺
```

## GitLens

Provides Git CodeLens information (most recent commit, # of authors), on-demand inline blame annotations, status bar blame information, file and blame history explorers, and commands to compare changes with the working tree or previous versions.

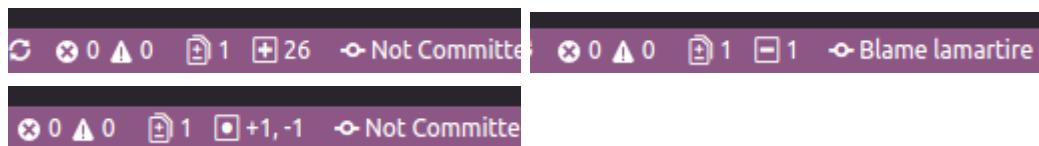
```

644     return undefined;
645   }
646 }
647
648 2 references | Eric Amodio, 5 days ago | author (Eric Amodio)
649   async getDiffForLine(uri: GitUri, line: number, sha1: string, sha2?: string): Promise<[IGitDiffLine | undefined, IGitDiffLine | undefined]> {
650     try {
651       const diff = await this.getDiffForFile(uri, sha1, sha2);
652       if (diff === undefined) return [undefined, undefined];
653
654       const chunk = diff.chunks.find(_ => _currentStart < line && _currentEnd > line);
655       if (chunk === undefined) return [undefined, undefined];
656
657       // Search for the line (skipping deleted lines -- since they don't currently exist in the editor)
658       // Keep track of the deleted lines for the original version
659       line = line - chunk.currentStart + 1;
660       let count = 0;
661       let deleted = 0;
662       for (const l of chunk.current) {
663         if (l === undefined) {
664           deleted++;
665           if (count === line) break;
666
667           continue;
668         }
669
670         if (count === line) break;
671         count++;
672       }
673
674       return [
675         chunk.previous[line + deleted - 1],
676         chunk.current[line + deleted + (chunk.currentStart - chunk.previousStart)]];
677     } catch (ex) {
678       return [undefined, undefined];
679     }
680   }
681
682 11 references | Eric Amodio, 25 days ago
683   async getLogCommit(repoPath: string | undefined, fileName: string, options?: { firstIfMissing?: boolean, previous?: boolean }): Promise<GitLogCommit | undefined>;
684   async getLogCommit(repoPath: string | undefined, fileName: string, sha: string | undefined, options?: { firstIfMissing?: boolean, previous?: boolean }): Promise<GitLogCommit | undefined>;
685   async getLogCommit(repoPath: string | undefined, fileName: string, shaOrOptions?: string | undefined | { firstIfMissing?: boolean, previous?: boolean }, options?: { fi
686   let sha: string | undefined = undefined;
687   if (typeof shaOrOptions === 'string') {
688     sha = shaOrOptions;
689   }
690   else if (options) {
691     options = shaOrOptions;
692   }
693
694   options = options || {};
695
696   const log = await this.getLogForFile(repoPath, fileName, sha, options.previous ? 2 : 1);
697   if (!log) return undefined;
698
699   const commit = sha ?? log.commits.get(sha);
700   if (!commit ?? sha ?? !options.firstIfMissing) return undefined;
701
702   return commit ?? Iterables.first(log.commits.values());
703 }
704

```

## Git Indicators

Atom-like git indicators on active panel



## GitHub

Provides GitHub workflow support. For example browse project, issues, file (the current line), create and manage pull request. Support for other providers (e.g. gitlab or bitbucket) is planned. Have a look at the [README.md](#) on how to get started with the setup for this extension.

## GitHub Pull Request Monitor

This extension uses the GitHub api to monitor the state of your pull requests and let you know when it's time to merge or if someone requested changes.

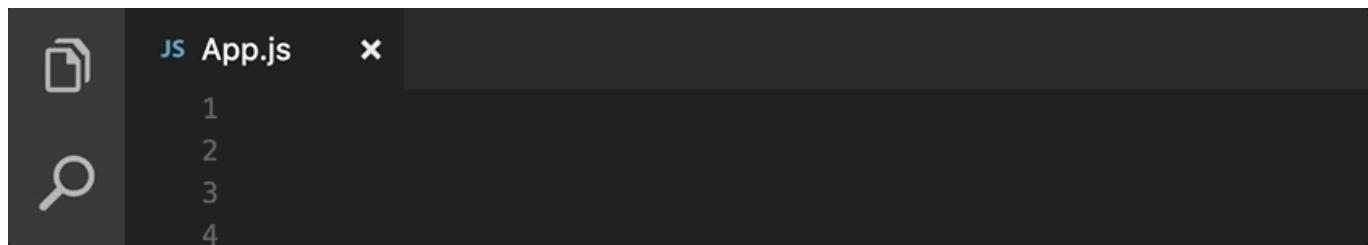


## Icon Fonts

Snippets for popular icon fonts such as Font Awesome, Ionicons, Glyphicons, Octicons, Material Design Icons and many more!

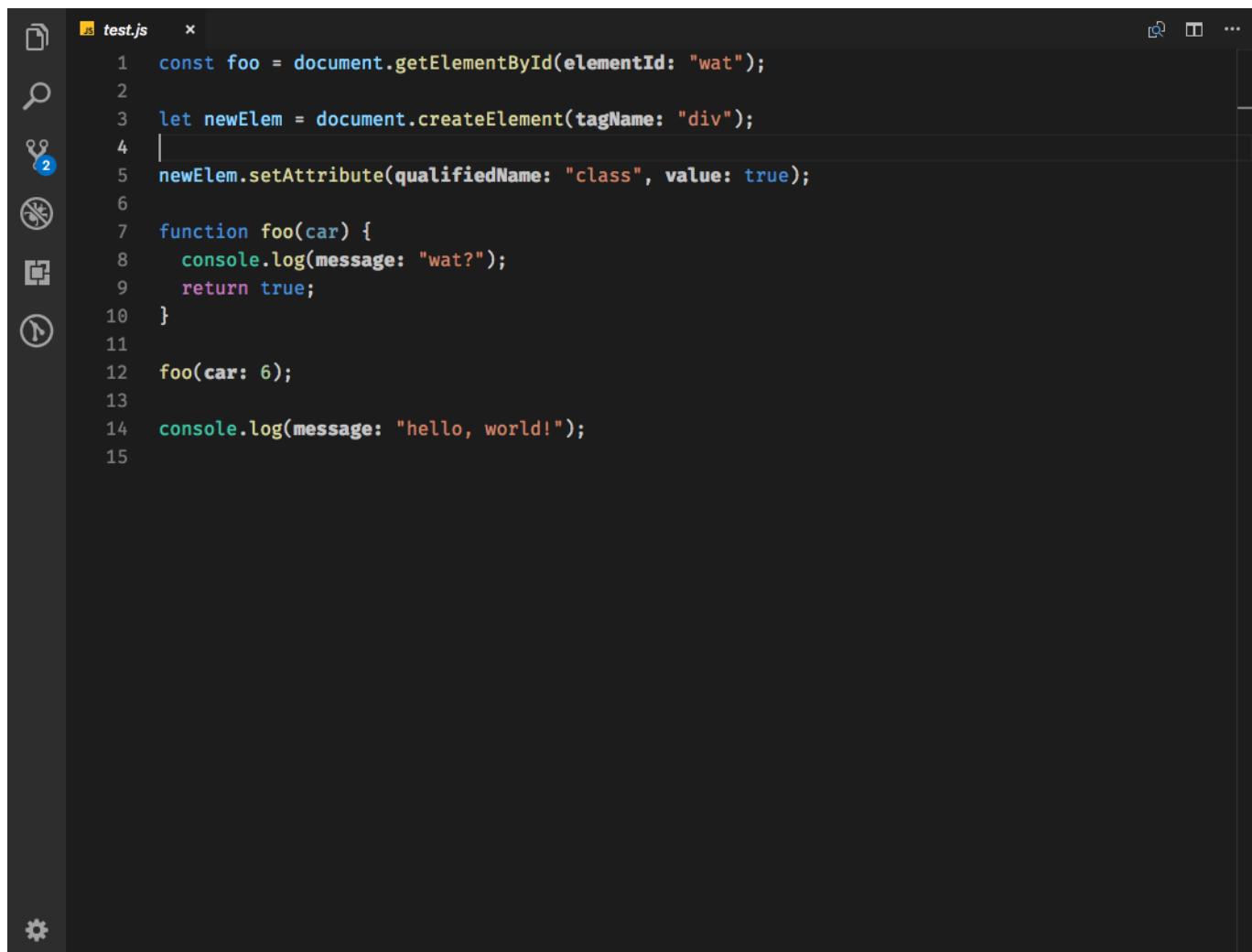
## Import Cost

This extension will display inline in the editor the size of the imported package. The extension utilizes webpack with babili-webpack-plugin in order to detect the imported size.



## JS Parameter Annotations

Provides annotations on function calls in JS/TS files to provide parameter names to arguments.

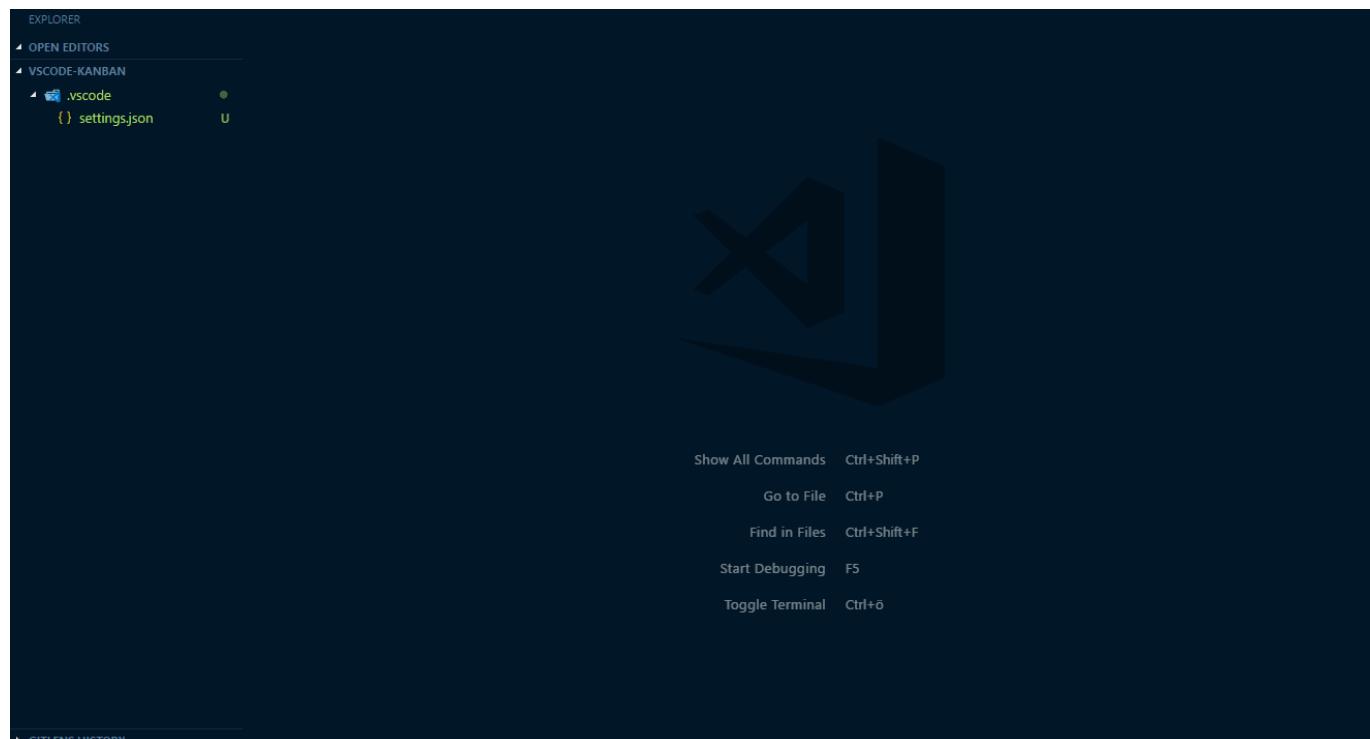


## Jumpy

Provides fast cursor movement, inspired by Atom's package of the same name.

```
1 import * as vscode from 'vscode';
2
3 const plusMinusLines = 60;
4 const numCharCodes = 26;
5
6 export function createCodeArray(): string[] {
7     const codeArray = new Array(numCharCodes * numCharCodes);
8     let codeIndex = 0;
9     for (let i = 0; i < numCharCodes; i++) {
10         for (let j = 0; j < numCharCodes; j++) {
11             codeArray[codeIndex++] = String.fromCharCode(97 + i) + String.fromCharCode(97 + j);
12         }
13     }
14     return codeArray;
15 }
16
17 let darkDataUriCache: { [index: string]: vscode.Uri } = {};
18 let lightDataUriCache: { [index: string]: vscode.Uri } = {};
19
20 export function createDataUriCaches(codeArray: string[]) {
21     codeArray.forEach(code => darkDataUriCache[code] = getSvgDataUri(code, 'white', 'black'));
22     codeArray.forEach(code => lightDataUriCache[code] = getSvgDataUri(code, 'black', 'white'));
23 }
24
```

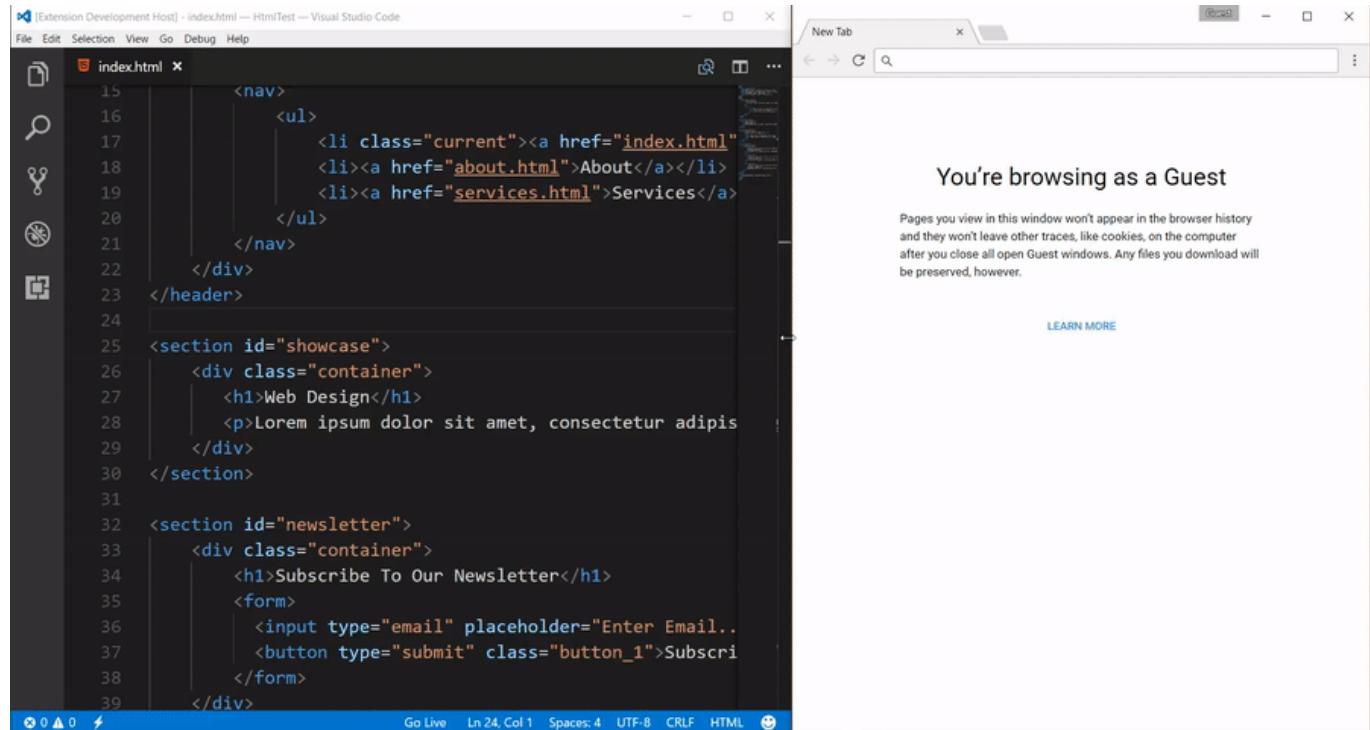
## Kanban



Simple Kanban board for use in Visual Studio Code, with time tracking and Markdown support.

## Live Server

Launch a development local Server with live reload feature for static & dynamic pages.

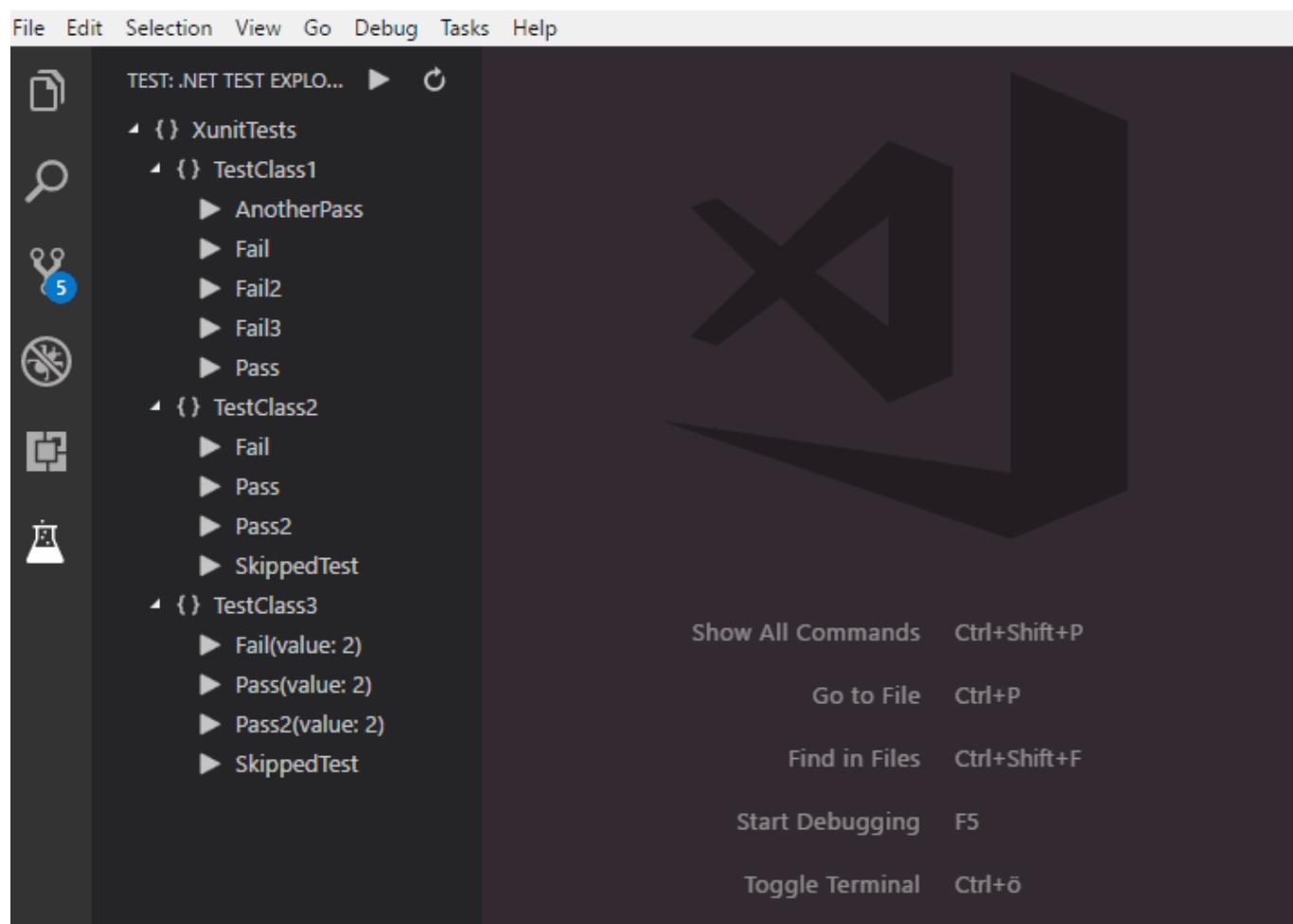


## Multiple clipboards

Override the regular Copy and Cut commands to keep selections in a clipboard ring

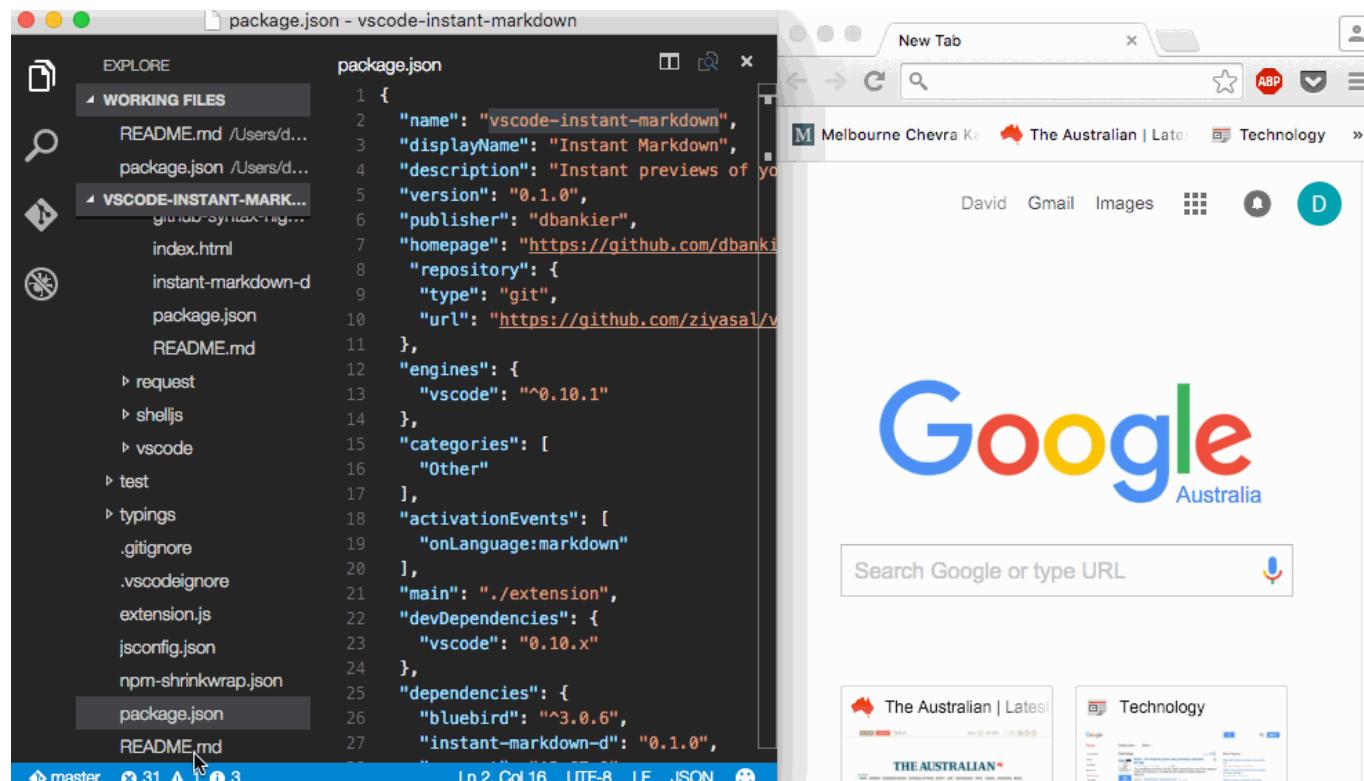
## Dotnet Core Test Explorer

View and run your .NET Core tests directly in the editor.



## Instant Markdown

Simply, edit markdown documents in vscode and instantly preview it in your browser as you type.



## npm Intellisense

Visual Studio Code plugin that autocompletes npm modules in import statements.



server.js

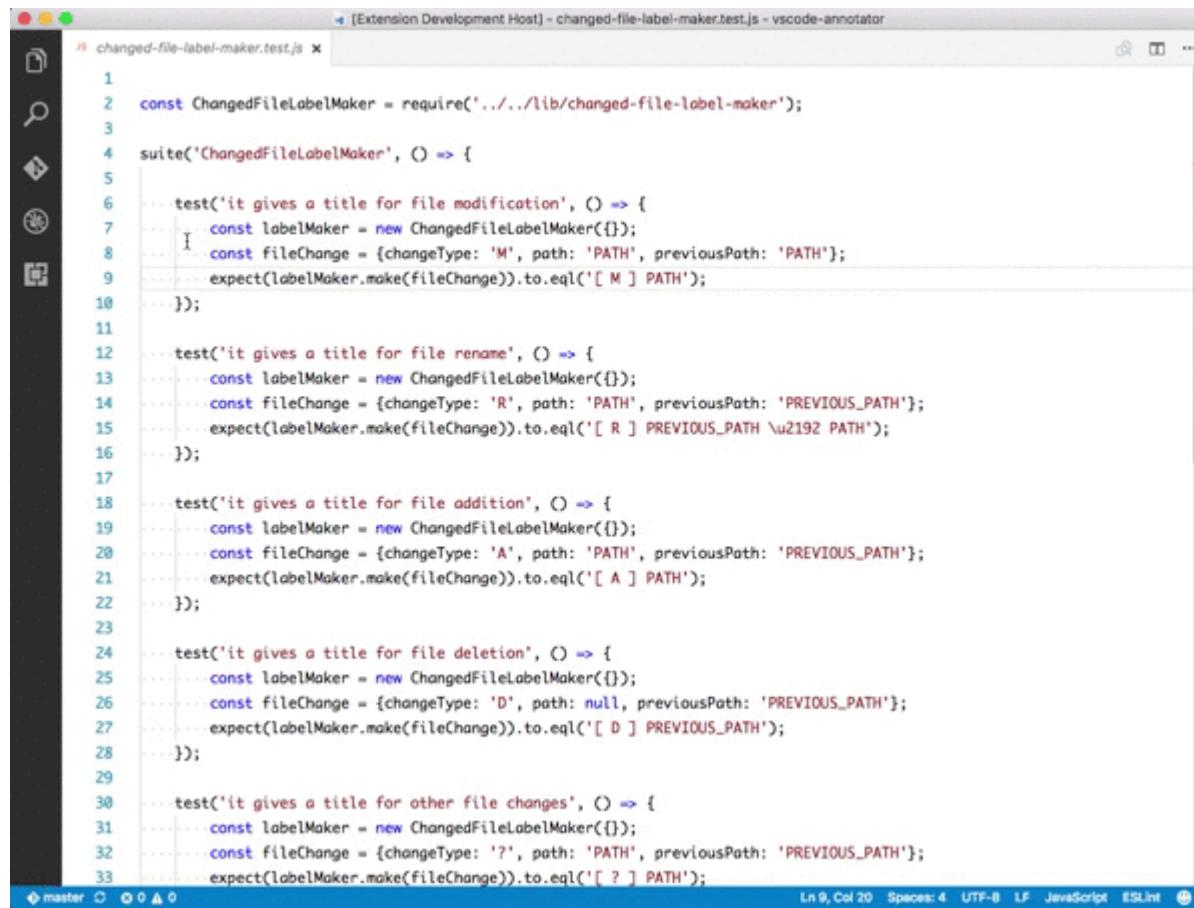
```

● server.js server
1 var express = require('express');
2 var bodyParser = require('body-parser')
3
4 var database = require('./database');
5
6 // Create server
7 var server = express();
8 server.use(bodyParser.json());
9

```

## Partial Diff

Compare (diff) text selections within a file, across different files, or to the clipboard



changed-file-label-maker.test.js

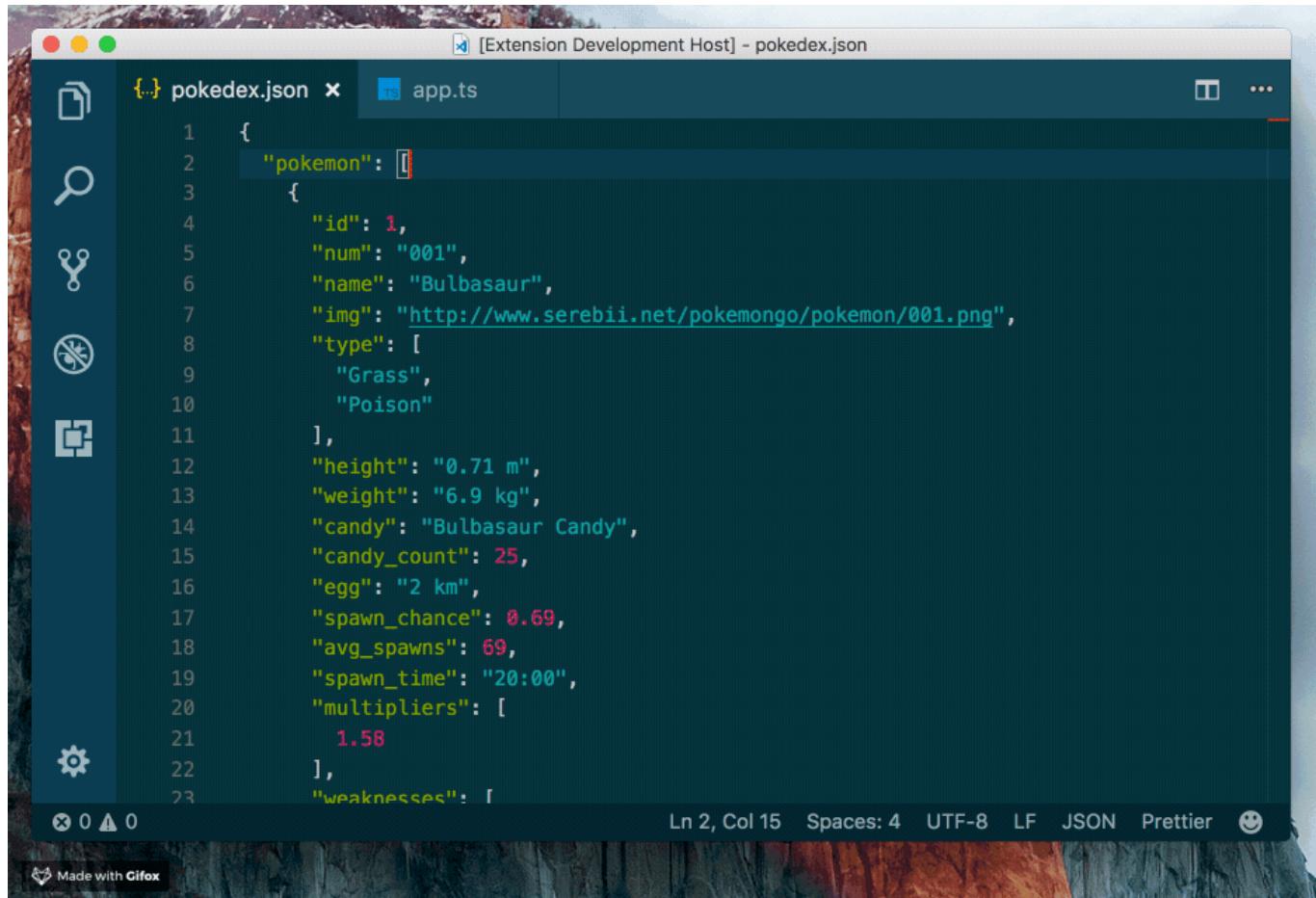
```

1
2 const ChangedFileLabelMaker = require('../lib/changed-file-label-maker');
3
4 suite('ChangedFileLabelMaker', () => {
5
6   test('it gives a title for file modification', () => {
7     const labelMaker = new ChangedFileLabelMaker();
8     const fileChange = {changeType: 'M', path: 'PATH', previousPath: 'PATH'};
9     expect(labelMaker.make(fileChange)).to.eql('[ M ] PATH');
10  });
11
12  test('it gives a title for file rename', () => {
13    const labelMaker = new ChangedFileLabelMaker();
14    const fileChange = {changeType: 'R', path: 'PATH', previousPath: 'PREVIOUS_PATH'};
15    expect(labelMaker.make(fileChange)).to.eql('[ R ] PREVIOUS_PATH \u2192 PATH');
16  });
17
18  test('it gives a title for file addition', () => {
19    const labelMaker = new ChangedFileLabelMaker();
20    const fileChange = {changeType: 'A', path: 'PATH', previousPath: 'PREVIOUS_PATH'};
21    expect(labelMaker.make(fileChange)).to.eql('[ A ] PATH');
22  });
23
24  test('it gives a title for file deletion', () => {
25    const labelMaker = new ChangedFileLabelMaker();
26    const fileChange = {changeType: 'D', path: null, previousPath: 'PREVIOUS_PATH'};
27    expect(labelMaker.make(fileChange)).to.eql('[ D ] PREVIOUS_PATH');
28  });
29
30  test('it gives a title for other file changes', () => {
31    const labelMaker = new ChangedFileLabelMaker();
32    const fileChange = {changeType: '?', path: 'PATH', previousPath: 'PREVIOUS_PATH'};
33    expect(labelMaker.make(fileChange)).to.eql('[ ? ] PATH');

```

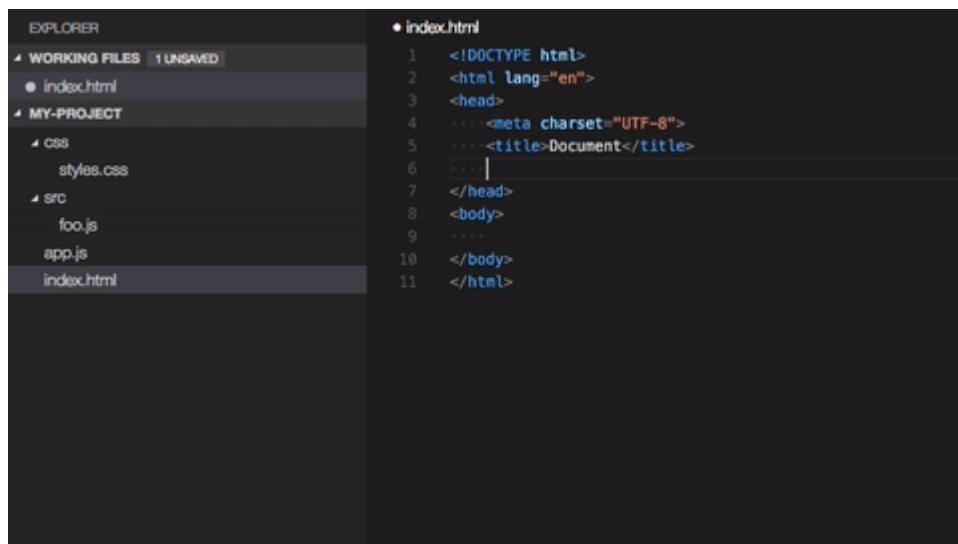
## Paste JSON as Code

Infer the structure of JSON and paste it as types in many programming languages



## Path IntelliSense

Visual Studio Code plugin that autocompletes filenames



## Power Tools

Extends Visual Studio Code via things like Node.js based scripts or shell commands, without writing separate extensions

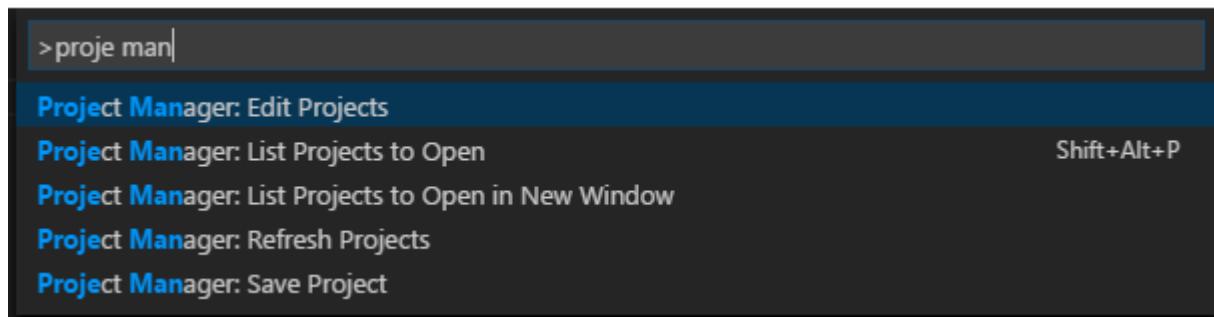
A screenshot of the Visual Studio Code interface. On the left, there is a dark-themed code editor window titled "settings.json". The code inside is as follows:

```
1  {
2      "ego.power-tools": {
3          }
4 }
```

The status bar at the bottom of the screen shows the file path "Zeile 4, Spalte 2", the number of blank lines "Leerzeichen: 4", the encoding "UTF-8", and the file type "JSON with Comments".

## Project Manager

Easily switch between projects.



## Remote VSCode

Allow user to edit files from Remote server in Visual Studio Code directly.

## REST Client

Allows you to send HTTP request and view the response in Visual Studio Code directly.

```

1 GET https://api.github.com/users/Huachao
2
3 ###
4
5 GET https://api.github.com/repos/Huachao/vscode-restclient HTTP/1.1
6
7 ###
8
9 POST http://requestb.in/tcdce4tc?k=a%20bc
Date: {{$guid}}
10
11 {
12     "key": "{$guid}",
13     "key2": "{$timestamp - 1 d}",
14     "key3": "{$guid}",
15     "key4": "{$Guid}"
16     #rest
17     "key5": "{$timestampfasdfa}",
18     "key6": "{}",
19     "key7": "{$d}",
20     "key8": "{$timestamp}",
21     "key9": "{$randomInt 1 19}",
22     "key10": "{$randomInt 21 19}",
23 }

```

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF HTTP ☺

## Settings Sync

Synchronize settings, snippets, themes, file icons, launch, key bindings, workspaces and extensions across multiple machines using GitHub Gist

## Todo Tree

Custom keywords, highlighting, and colors for TODO comments. As well as a sidebar to view all your current tags.

```

// handles the user input and output, main function controlling the game
int play_game(char *in) {
    game *board = file_load(in);

    int gameOver = 0;

    int row = -1;
    int col = -1;
    int move = -1;

    do {

```

## Toggle Quotes

Cycle between single, double and backtick quotes

```
const sentence = 'wes is kinda cool';
```



## WakaTime

Automatic time tracker and productivity dashboard showing how long you coded in each project, file, branch, and language.

## Yo

Scaffold projects using [Yeoman](#)



## Formatting & Beautification

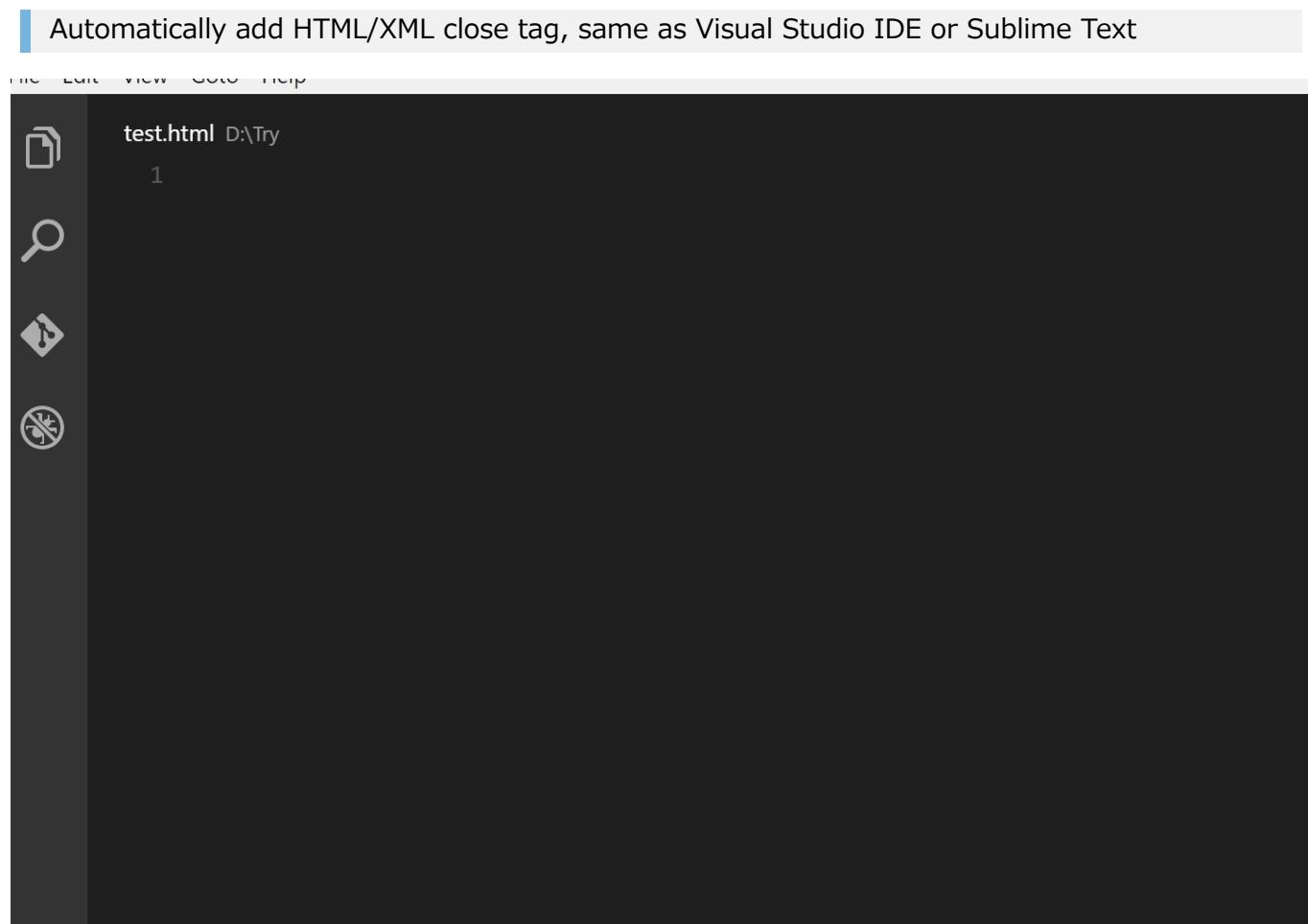
### Better Align

Align your code by colon(colon), assignment(=,+,-,\*,/=,/=) and arrow(=>). It has additional support for comma-first coding style and trailing comment.

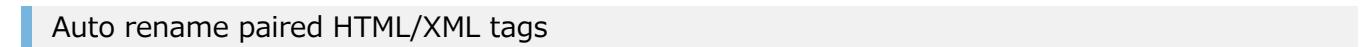
And it doesn't require you to select what to be aligned, the extension will figure it out by itself.

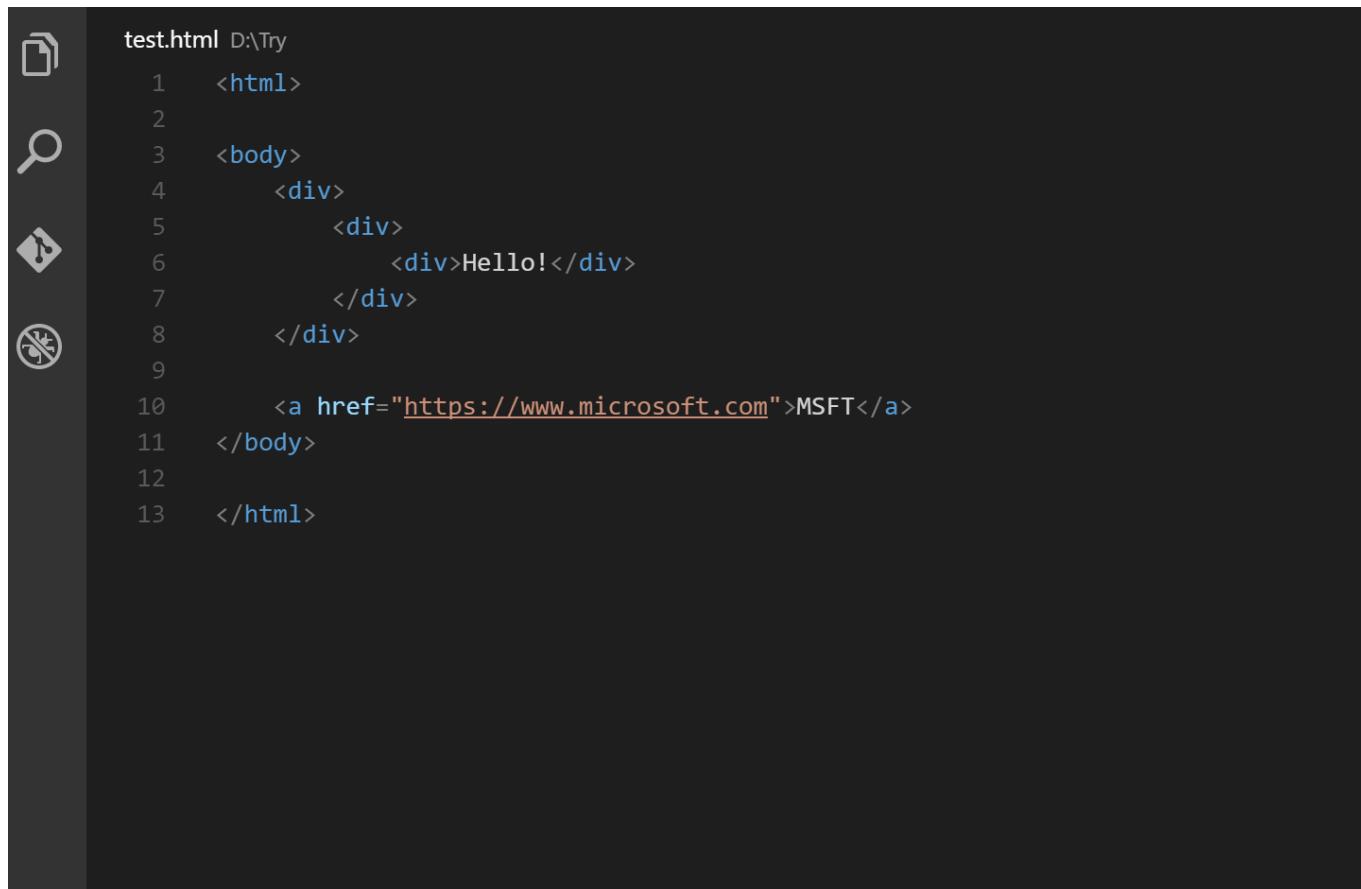
```
4  
3   ··this.abc = fdasf;  
2   ····this.dfafdf = dasfas;  
1   ···this.fdafdsf += fadsf;  
24  ····this.fadf = this.fdfad = this.ffdaf;  
1   ·····this.fdafd *= afdf /= fdaskjfdf = fda;  
2  
3  
4   ··this.abc=fdasf;// afdsa  
5   ··this.dfafdf=dasfas;// afdsa  
6   ···this.fdafdsf-=fadsf;····// afdsa  
7   ···this.fadf*=this.fdfad=this.ffdaf;····// afdsa  
8   ···this.fdafd=afdf=fdaskjfdf=fda;····// afdsafdaf  
9  }
```

## Auto Close Tag



## Auto Rename Tag





```
test.html D:\Try
1 <html>
2
3 <body>
4   <div>
5     <div>
6       <div>Hello!</div>
7     </div>
8   </div>
9
10  <a href="https://www.microsoft.com">MSFT</a>
11 </body>
12
13 </html>
```

## beautify

Beautify code in place for VS Code

## html2pug

Transform html to pug inside your Visual Studio Code, forget about using an external page anymore.

## ECMAScript Quotes Transformer

Transform quotes of ECMAScript string literals

A screenshot of the Visual Studio Code interface. On the left, there's a file named "es-quotes.ts". The code in the editor is:

```
1 /*  
2      ES Quotes for Visu  
3      https://github.com/vilic/vscode-es-quotes  
4 */  
5  
6 'switch me to "double[quotes" and back';  
7  
8 // Blame Code for incorrect highlighting, please.  
9 'switch me\n\  
10 to `template ${string}` \  
11 and back';  
12  
13 "even I'm broken..."
```

The cursor is at the end of the line "6 'switch me to "double[quotes" and back';". A code completion dropdown menu is open, showing the suggestion ">double[quotes" and back'" with the description "ES Quotes: Switch between Single and Double".

## Paste and Indent

Paste code with "correct" indentation



## Sort Lines

Sorts lines of text in specific order

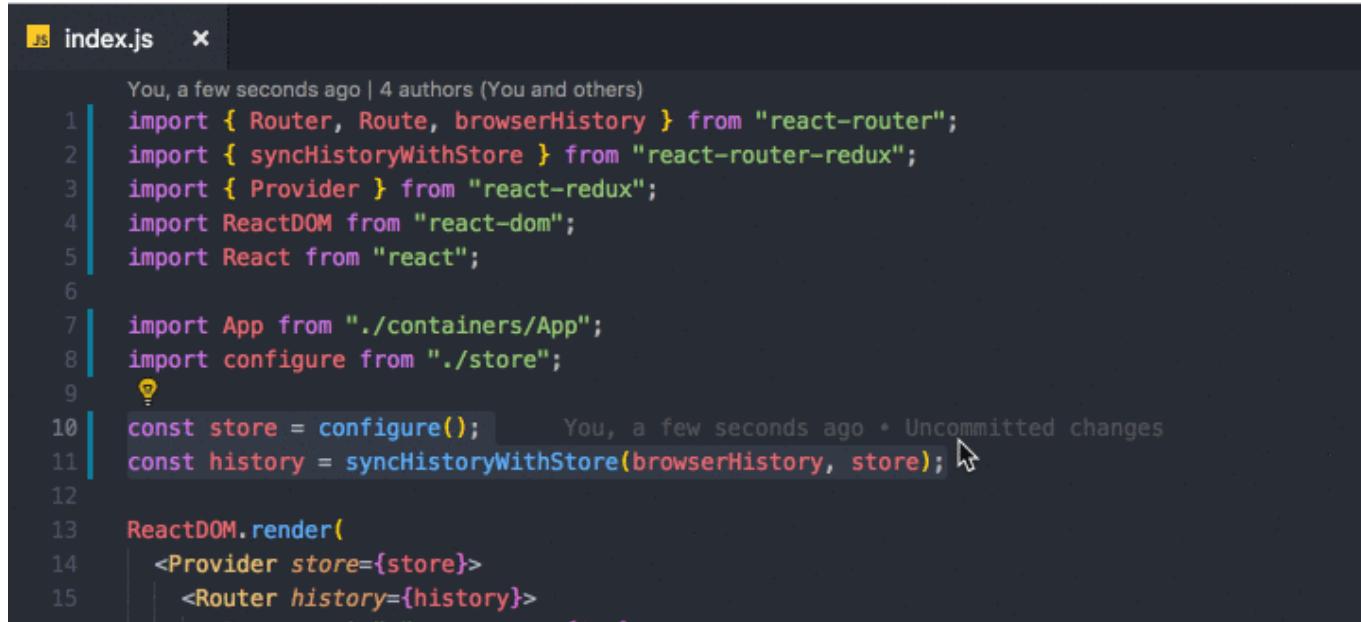
A screenshot of the Visual Studio Code interface. The title bar shows "Untitled-2 - Visual Studio Code". The menu bar includes File, Edit, View, Goto, Help. The sidebar on the left has icons for files (with a '2' count), search, and other extensions. The main editor area contains the following text:

```
1 Test  
2 Foo  
3 Sort  
4 Bar  
5 Lines
```

The word "Lines" is currently selected. The status bar at the bottom shows "Ln 5, Col 6" and "Plain Text".

## Surround

A simple yet powerful extension to add wrapper templates around your code blocks.



```

1 import { Router, Route, browserHistory } from "react-router";
2 import { syncHistoryWithStore } from "react-router-redux";
3 import { Provider } from "react-redux";
4 import ReactDOM from "react-dom";
5 import React from "react";
6
7 import App from "./containers/App";
8 import configure from "./store";
9
10 const store = configure(); You, a few seconds ago | 4 authors (You and others)
11 const history = syncHistoryWithStore(browserHistory, store); ⚡
12
13 ReactDOM.render(
14   <Provider store={store}>
15     <Router history={history}>
16       <Route path="/" component={App} />
17     </Router>
18   </Provider>
19 )

```

## Wrap Selection

 Wraps selection or multiple selections with symbol or multiple symbols

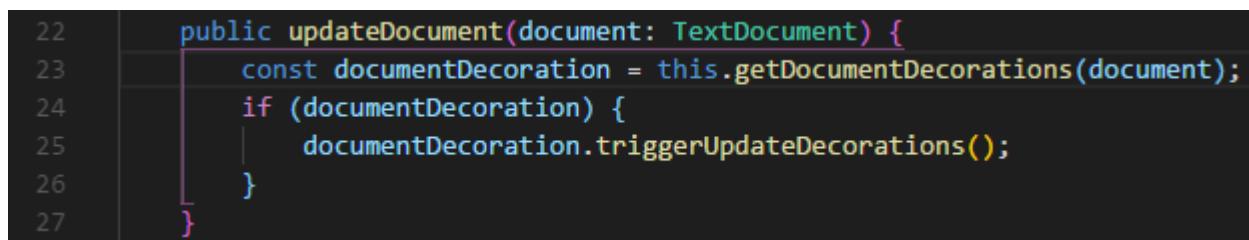


## Formatting Toggle

 Allows you to toggle your formatter on and off with a simple click

## Bracket Pair Colorizer

 This extension allows matching brackets to be identified with colours. The user can define which characters to match, and which colours to use.



```

22   public updateDocument(document: TextDocument) {
23     const documentDecoration = this.getDocumentDecorations(document);
24     if (documentDecoration) {
25       documentDecoration.triggerUpdateDecorations();
26     }
27   }

```

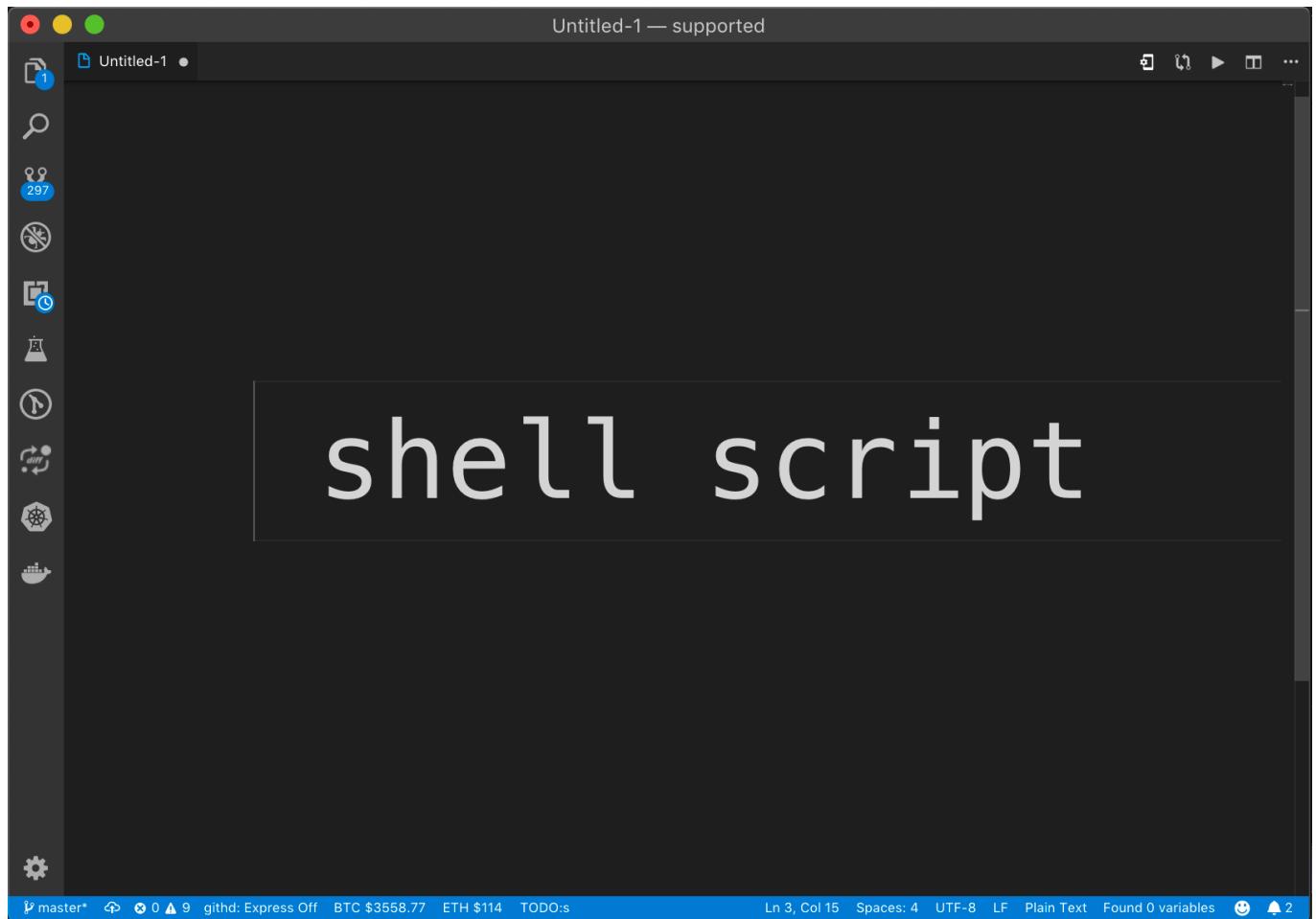
## Auto Import

 Automatically finds, parses and provides code actions and code completion for all available imports. Works with Typescript and TSX.



## shell-format

 shell script & Dockerfile & dotenv format



## Vscode Google Translate

Quickly translate selected text right in your code

The screenshot shows the VS Code interface with a dark theme. The Explorer sidebar on the left contains icons for files, folders, search, and other extensions. The main editor window titled 'Untitled-1' displays a JSON configuration file named 'en'. The status bar at the bottom shows '7 selections (122 characters selected)' and various extension status indicators like ESLint and Prettier.

```
const en = {  
  "enroll": {  
    "manage_enroll": "Manage enrollments",  
    "user_enroll": "Enroll this user",  
    "alert": {  
      "error_email_exist": "Email exist",  
      "error_wrong_format": "Wrong format",  
      "error_empty_value": "Empty value",  
      "enrollment_success": "User successfully enrolled",  
      "unenrollment_success": "User successfully unenrolled",  
    },  
  },  
};  
;
```

## Explorer Icons

### City Lights Icons

The screenshot shows the VS Code interface using the 'city-lights' theme. The Explorer sidebar on the left lists various configuration files and dependencies. The main editor window titled 'start.ts' shows a TypeScript file with code related to theme activation. The status bar at the bottom shows 'Ln 12, Col 10' and other standard status indicators.

```
import * as vscode from 'vscode';  
import { ThemeStatus } from '../helpers/versioning';  
  
/** Initialization of the icons every time the theme get activated */  
export const showStartMessages = (themeStatus: Promise<ThemeStatus>) => {  
  return themeStatus.then((status: ThemeStatus) => {  
    if (status === ThemeStatus.updated) {  
      console.log('UPDATED');  
    }  
    else if (status === ThemeStatus.neverUsedBefore) {  
      vscode.window.showInformationMessage('Thanks for using City Lights');  
    }  
  });  
};
```

## VSCode Icons

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists the project structure, with the 'examples' folder selected. The main editor area displays the 'supportedExtensions.ts' file, which contains TypeScript code defining file icons. The status bar at the bottom provides build information and file details.

```
/* tslint:disable max-line-length */
import { languages } from './languages';
import {
  FileFormat,
  IFileCollection,
} from '../models';

export const extensions: IFileCollection = {
  default: [
    file: { icon: 'file', format: FileFormat.svg },
  ],
  supported: [
    { icon: 'access', extensions: ['accdb', 'accdt', 'accdr'] },
    { icon: 'actionscript', extensions: [], languages: [{}], (property) icon: string['ai'] }, format: FileFormat.svg },
    { icon: 'ai2', extensions: ['ai'], format: FileFormat.svg },
    { icon: 'angular', extensions: ['angular-cli.json'] },
    { icon: 'ng_component_ts', extensions: ['component'] },
    { icon: 'ng_component_js', extensions: ['component'] },
    { icon: 'ng_smart_component_ts', extensions: ['pag'] },
    { icon: 'ng_smart_component_js', extensions: ['pag'] },
    { icon: 'ng_directive_ts', extensions: ['directive'] },
    { icon: 'ng_directive_js', extensions: ['directive'] },
    { icon: 'ng_pipe_ts', extensions: ['pipe.ts'] }, format: FileFormat.svg },
    { icon: 'ng_pipe_js', extensions: ['pipe.js'] }, format: FileFormat.svg },
    { icon: 'ng_service_ts', extensions: ['service.ts'] },
    { icon: 'ng_service_js', extensions: ['service.js'] }
  ]
}
```

master 0 0 ▲ 0 [TypeScript Importer]: Symbols: 81 Ln 8, Col 17 Spaces: 2 UTF-8 LF TypeScript 2.1.4 TSLint 😊

## Seti Icons

```

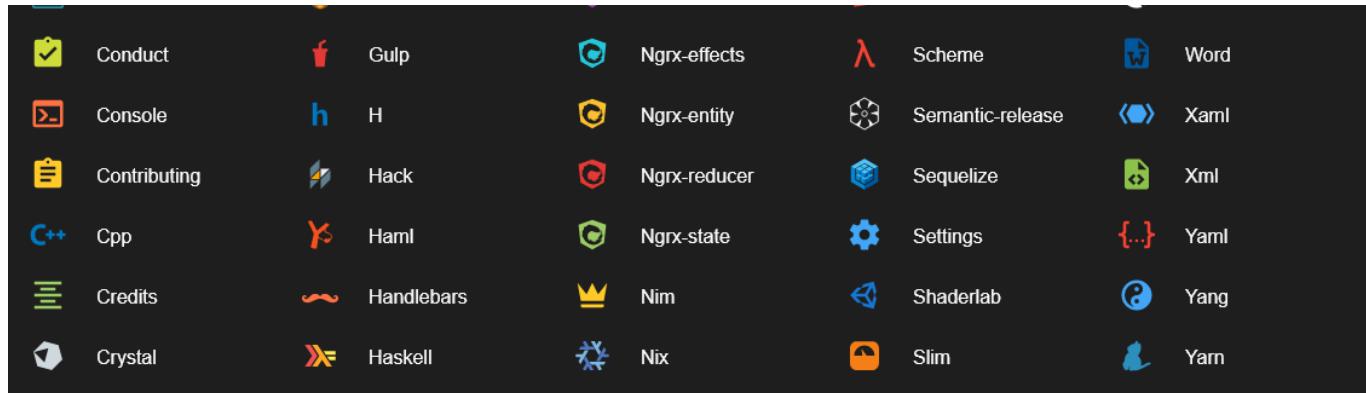
{
  "fonts": [
    {
      "id": "seti",
      "src": [
        {
          "path": "./fonts/seti.woff",
          "format": "woff"
        }
      ],
      "weight": "normal",
      "style": "normal",
      "size": "150%"
    }
  ],
  "iconDefinitions": {
    "_mp3_icon": {
      "fontCharacter": "\ue003",
      "fontColor": "#a074c4"
    },
    "_mp3_icon_light": {
      "fontCharacter": "\ue003",
      "fontColor": "#5f8b3b"
    },
    "_wav_icon": {
      "fontCharacter": "\ue003",
      "fontColor": "#a074c4"
    },
    "_wav_icon_light": {
      "fontCharacter": "\ue003",
      "fontColor": "#5f8b3b"
    },
    "_ogg_icon": {
      "fontCharacter": "\ue003",
      "fontColor": "#a074c4"
    },
    "_ogg_icon_light": {
      "fontCharacter": "\ue003",
      "fontColor": "#5f8b3b"
    },
    "_bowerrc_icon": {
      "fontCharacter": "\ue006",
      "fontColor": "#e37933"
    },
    "_bowerrc_icon_light": {
      "fontCharacter": "\ue006",
      "fontColor": "#1c86cc"
    }
  }
}

```

## Material Icon Theme

Icon	Name	Icon	Name	Icon	Name	Icon	Name	Icon	Name
3d	3d	C#	Csharp	X	Haxe	Nodejs	Smarty	💡	Smarty
{AS}	Actionscript	.Css	Css	Hcl	Hcl	Nodejs_alt	Snyk	🛠	Snyk
Android	Android	.Css-map	Css-map	helm	Helm	Nodemon	Solidity	💎	Solidity
Angular	Angular	Cucumber	Cucumber	Heroku	Heroku	Npm	Stencil	☰	Stencil
Angular-component	Angular-component	Cuda	Cuda	hpp	Hpp	Nunjucks	Storybook	ſ	Storybook
Angular-directive	Angular-directive	D	D	Html	Html	Nuxt	Stylelint	呔	Stylelint
Angular-guard	Angular-guard	Dart	Dart	Http	Http	Ocaml	Stylus	ſ	Stylus
Angular-pipe	Angular-pipe	Database	Database	I18n	I18n	Pdf	Sublime	⚡	Sublime
Angular-resolver	Angular-resolver	Diff	Diff	Image	Image	Perl	Svelte	ſ	Svelte
Angular-service	Angular-service	Django	Django	Ionic	Ionic	Php	Svg	*	Svg
Apiblueprint	Apiblueprint	Docker	Docker	Istanbul	Istanbul	Php_elephant	Swc	🔗	Swc

 Applescript	 Document	 Java	 Postcss	 Swift
 Appveyor	 Dotjs	 Javascript	 Powershell	 Table
 Arduino	 Drone	 Javascript-map	 Prettier	 Tailwindcss
 Asciidoc	 Edge	 Jenkins	 Prisma	 Terraform
 Assembly	 Editorconfig	 Jest	 Processing	 Test-js
 Audio	 Ejs	 Jinja	 Protractor	 Test-javascript
 Aurelia	 Elixir	 Json	 Pug	 Test-is
 Authors	 Elm	 Julia	 Puppet	 Tex
 Autohotkey	 Email	 Jupyter	 Purescript	 Todo
 Autoit	 Erlang	 Karma	 Python	 Travis
 Azure	 Eslint	 Key	 Python-misc	 Tune
 Azure-pipelines	 Exe	 Kivy	 R	 Twig
 Babel	 Fastlane	 Ki	 Racket	 Typescript
 Ballerina	 Favicon	 Kotlin	 Raml	 Typescript-def
 Bazel	 Firebase	 Laravel	 Razor	 Url
 Bitbucket	 Flash	 Less	 React	 Vagrant
 Bithound	 Flow	 Lib	 React_ts	 Velocity
 Blink	 Font	 Livescript	 Readme	 Verilog
 Bower	 Foxpro	 Lock	 Reason	 Vfl
 Browserlist	 Fsharp	 Log	 Red	 Video
 Bucklescript	 Fusebox	 Lua	 Redux-action	 Virtual
 Buildkite	 Gatsby	 Makefile	 Redux-reducer	 Visualstudio
 C	 Gemfile	 Markdown	 Redux-store	 Vscode
 Cabal	 Git	 Markojs	 Restql	 Vue
 Cake	 Gitlab	 Mathematica	 Riot	 Vue-config
 Certificate	 Go	 Matlab	 Robot	 Vuex-store
 Changelog	 Go_gopher	 Mdx	 Rollup	 Wakatime
 Circleci	 Godot	 Merlin	 Routing	 Wallaby
 Clojure	 Godot-assets	 Mint	 Ruby	 Watchman
 Cloudfoundry	 Gradle	 Mjml	 Rust	 Webassembly
 Cmake	 Graphcool	 Mocha	 San	 Webhint
 Code-climate	 Graphql	 Mxml	 Sass	 Webpack
Coffee	Groovy	Netlify	Sbt	Wepy
Coldfusion	Grunt	Ngrx-actions	Scala	Wolframlanguage



# Uncategorized

# Browser Preview

Browser Preview for VS Code enables you to open a real browser preview inside your editor that you can debug. Browser Preview is powered by Chrome Headless, and works by starting a headless Chrome instance in a new process. This enables a secure way to render web content inside VS Code, and enables interesting features such as in-editor debugging and more!

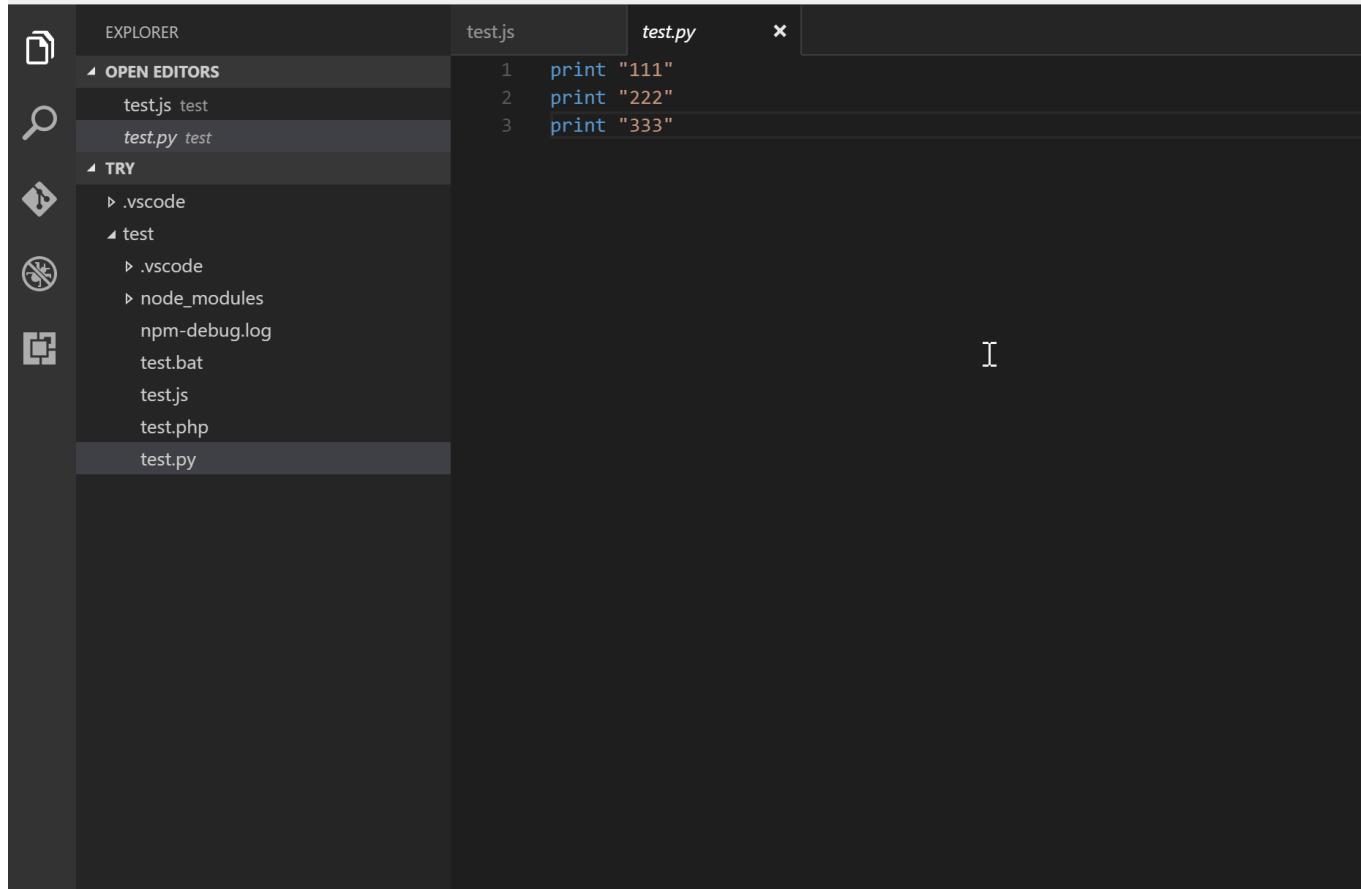
The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `App.tsx`, `react-dom.development.js`, and `node_modules`.
- Variables:** A sidebar showing variables with their current values.
- Watch:** A sidebar showing the value of the `onClick` function as `<button onClick={this.onClicked}> Click here</button>`.
- Call Stack:** A sidebar showing the current call stack.
- Code Editor:** The main editor area contains the `App.tsx` file with the following code:

```
1 import React, { Component } from 'react';
2 import Logo from './logo.svg';
3 import './App.css';
4
5 class App extends Component {
6
7   onClicked() {
8     console.log('Hello from React!');
9   }
10
11   render() {
12     return (
13       <div className="App">
14         <header className="App-header">
15           <img src={Logo} className="App-Logo" alt="logo" />
16           <p>
17             Edit <code>src/App.tsx</code> and save to reload.
18           </p>
19           <button onClick={this.onClicked}> Click here</button>
20           <a href="https://reactjs.org" target="_blank" rel="noopener noreferrer">
21             Learn React
22           </a>
23         </header>
24       </div>
25     );
26   }
27 }
28
29 </body>
30 </html>
```
- Terminal:** Shows the command `1: node`.
- Output:** Shows the message `Compiled successfully!`.
- Browser Preview:** Shows the browser preview of the application at `http://localhost:3000`.
- Breakpoints:** Shows breakpoints set in the code.
- Status Bar:** Shows the file path as `App.tsx — my-app`, line count as `Ln 21, Col 13`, and other settings like spaces, encoding, and type script version.

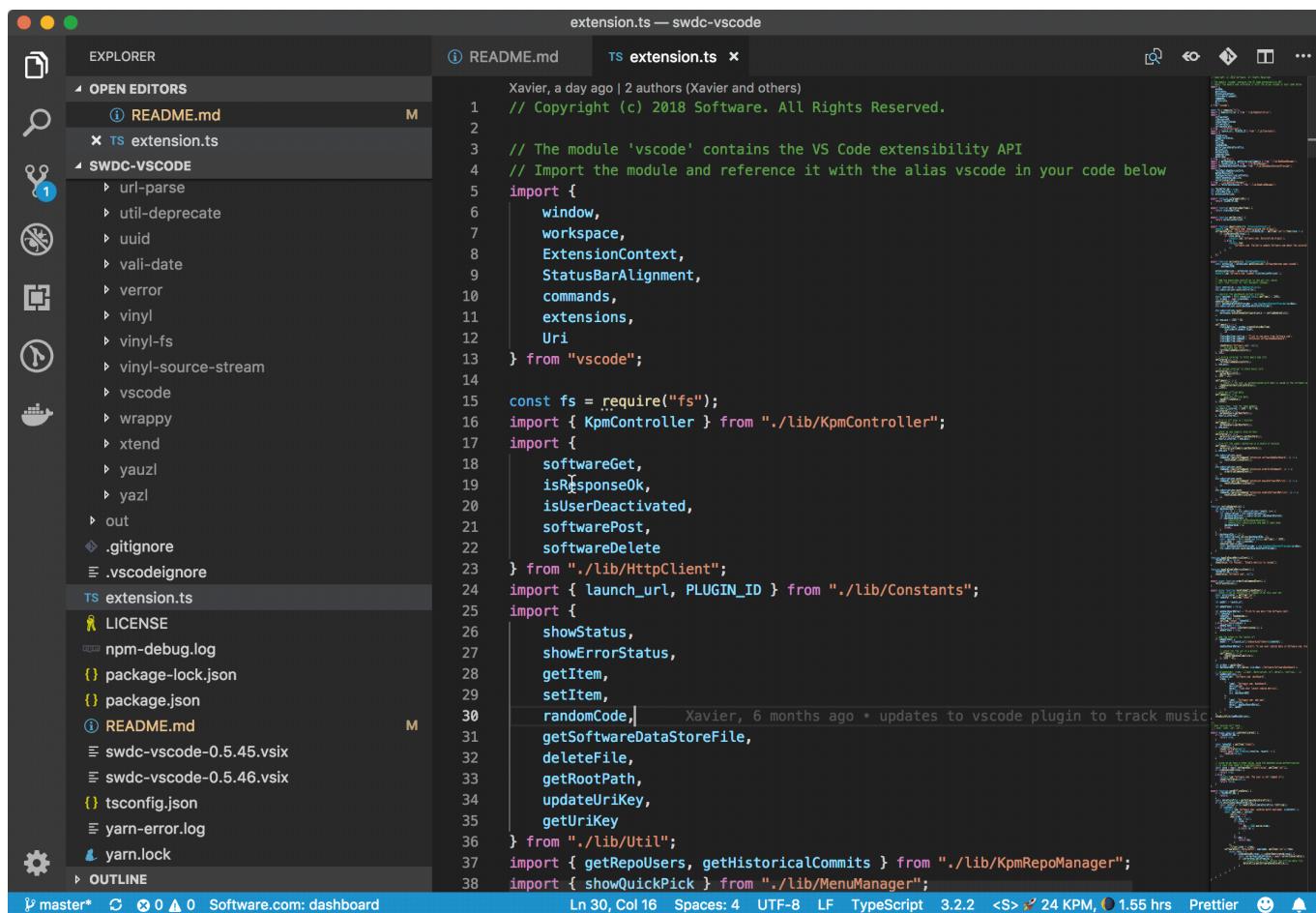
# Code Runner

Run code snippet or code file for multiple languages: C, C++, Java, JavaScript, PHP, Python, Perl, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, C# Script, VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script



## Code Time

Automatic time reports by project and other programming metrics right in VS Code.



## Color Highlight

Highlight web colors in your editor

```
1 .test {  
2     background: #fff;  
3     color: darkblue;  
4 }  
5  
6 .test2 {  
7     background: rgb(150, 50, 50);  
8 }
```

## Output Colorizer

Syntax highlighting for the VS Code Output Panel and log files

```
OUTPUT  
#0 RUNNING 0/29/2010 09:58:01 +4 00.00% 05.52 MB OT 1.00 GB 02.80 MB OT 1.00 GB  
LOCAL: Logs connection closed  
WARN: Bluemix logs connection interrupted or stalled.  
WARN: Attempting reconnection...  
LOCAL: Terminating existing web socket  
LOCAL: Fetching Bluemix logs for app: ea106447-6e1d-47b8-9930-43f8f777777d  
LOCAL: Logs connection opened  
LOCAL: Stopping app...  
SERVER: Updated app with guid ea106447-6e1d-47b8-9930-43f8f777777d {"state":>"STOPPED"}Y  
LOCAL: Starting app...  
SERVER: Updated app with guid ea106447-6e1d-47b8-9930-43f8f777777d {"state":>"STARTED"}  
SERVER: Starting app instance (index 0) with guid ea106447-6e1d-47b8-9930-43f8f777777d  
SERVER: Stopping app instance (index 0) with guid ea106447-6e1d-47b8-9930-43f8f777777d  
SERVER: Stopped app instance (index 0) with guid ea106447-6e1d-47b8-9930-43f8f777777dZ
```

## Dash

Dash integration in Visual Studio Code

The screenshot shows the Microsoft Edge browser's developer tools. On the left, the 'Elements' tab is open, displaying the CSS rule `.test { background: #fff; }`. The sidebar on the left lists various CSS properties under the heading 'background'. The main content area shows the MDN Web Docs article for the `background` property, which includes a summary, an 'IN THIS ARTICLE' section, and a detailed explanation of how it's a shorthand for setting individual background values.

## Edit with Shell Command

Leverage your favourite shell commands to edit text

The screenshot shows the VS Code editor with a file named 'Untitled-1'. The file contains a CSV dataset with 12 rows. The columns are labeled 'FIRST\_NAME', 'LAST\_NAME', 'EMAIL', and 'TEL'. The data includes five entries for John Smith, one for Mark Ryan, one for Kevin Williams, one for Kelly Charles, and one for David Harris. The last two rows are blank. The left sidebar shows icons for file operations like new file, save, and search.

	FIRST_NAME	LAST_NAME	EMAIL	TEL
1				
2	Extract only emails from the CSV text and sort them			
3				
4	John,Smith	john.smith@example.com	043-111-111	
5	Mark,Ryan	mark.ryan@example.com	042-222-222	
6	Kevin,Williams	kevin.williams@example.com	041-333-333	
7	Kelly,Charles	kelly.charles@example.com	044-444-444	
8	David,Harris	david.harris@example.com	045-555-555	
9				
10				
11				
12				

## Editor Config for VS Code

Editor Config for VS Code

## ftp-sync

Auto-sync your work to remote FTP server

## Highlight JSX/HTML tags

Highlights matching tags in the file.

```
1 const component = () => (
2   <div style="margin: 10px;">
3     <input onChange={(e) => changeValue(e.target.value)} />
4     <button onClick={() => submit()}>
5       Submit
6     </button>
7     <a href="http://site.info/">
8       
9     </a>
10    </div>
11  )
```

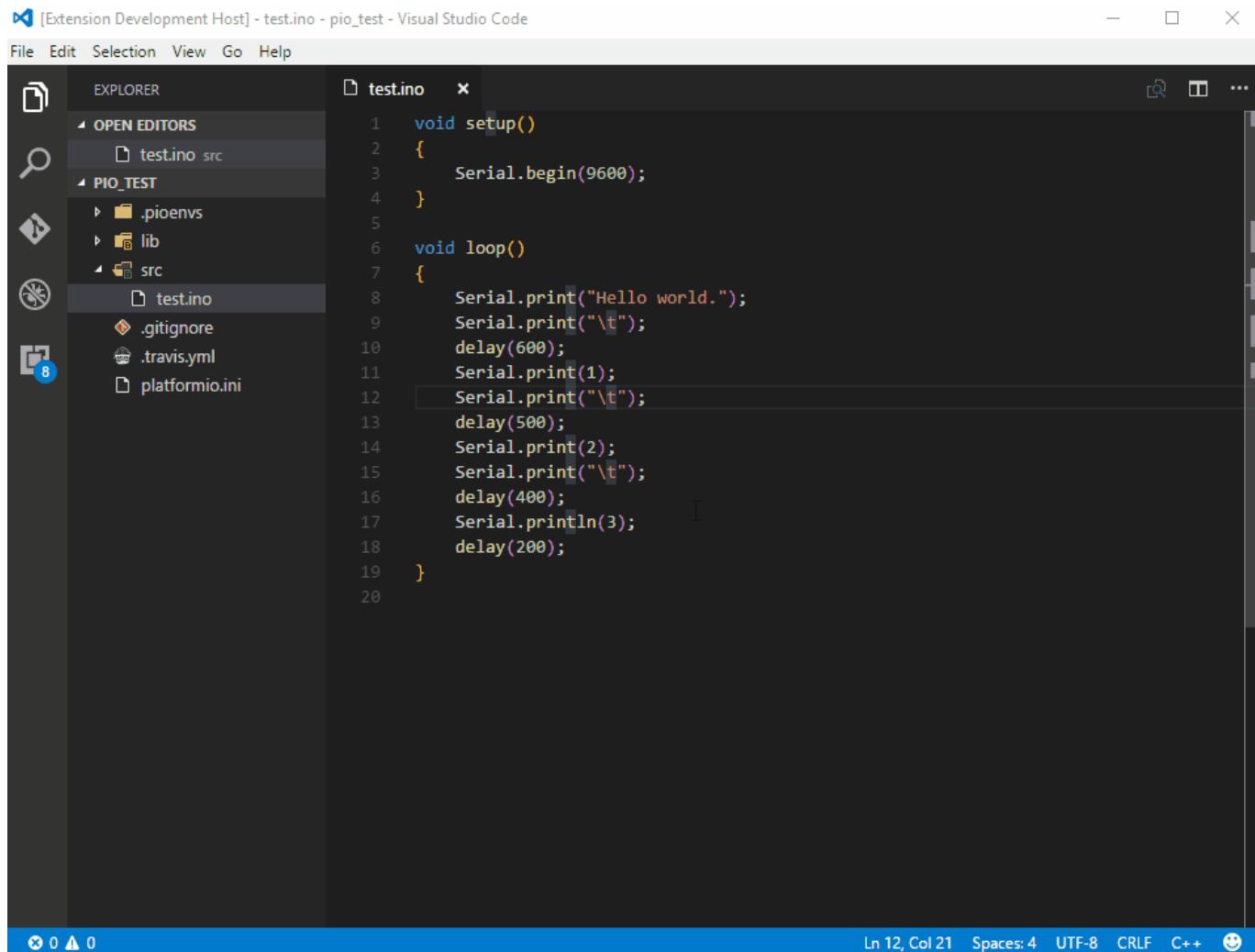
## Indent Rainbow

A simple extension to make indentation more readable.

```
nimbench_ex2.nim src/misc
1 import nimbench
2
3 bench(str, m):
4   for n in 0..<m:
5     var dest ="apple"
6     while(dest.len<100):
7       var
8         offs=dest.len-5
9         i=offs
10        while i<offs+10:
11          dest.add(dest[i])
12          i.inc
13        doNotOptimizeAway(dest)
14
15 runBenchmarks()
```

## PlatformIO

An open source ecosystem for IoT development: supports 350+ embedded boards, 20+ development platforms, 10+ frameworks. Arduino and ARM mbed compatible.



The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** [Extension Development Host] - test.ino - pio\_test - Visual Studio Code
- File Menu:** File Edit Selection View Go Help
- Explorer View:** Shows the project structure:
  - OPEN EDITORS: test.ino src
  - PIO\_TEST:
    - .pioenvs
    - lib
    - src
      - test.ino
  - .gitignore
  - .travis.yml
  - platformio.ini
- Code Editor:** The file test.ino is open, displaying the following code:

```
1 void setup()
2 {
3     Serial.begin(9600);
4 }
5
6 void loop()
7 {
8     Serial.print("Hello world.");
9     Serial.print("\t");
10    delay(600);
11    Serial.print(1);
12    Serial.print("\t");
13    delay(500);
14    Serial.print(2);
15    Serial.print("\t");
16    delay(400);
17    Serial.println(3);
18    delay(200);
19 }
20
```
- Status Bar:** Ln 12, Col 21 Spaces: 4 UTF-8 CRLF C++ 🎨

## Polacode

Polaroid for your code 📸.

```
src > App.vue > () "App.vue"
1  <template>
2    <div id="app">
3      
4      <HelloWorld msg="Welcome to Your Vue.js App"/>
5    </div>
6  </template>
7  |
8  <script>
9    import HelloWorld from './components/HelloWorld.vue'
10   |
11   export default {
12     name: 'app',
13     components: {
14       HelloWorld
15     }
16   }
17 </script>
18   |
19   <style>
20   #app {
21     font-family: 'Avenir', Helvetica, Arial, sans-serif;
22     -webkit-font-smoothing: antialiased;
23     -moz-osx-font-smoothing: grayscale;
24     text-align: center;
25     color: #2c3e50;
26     margin-top: 60px;
27   }
28 </style>
29
```

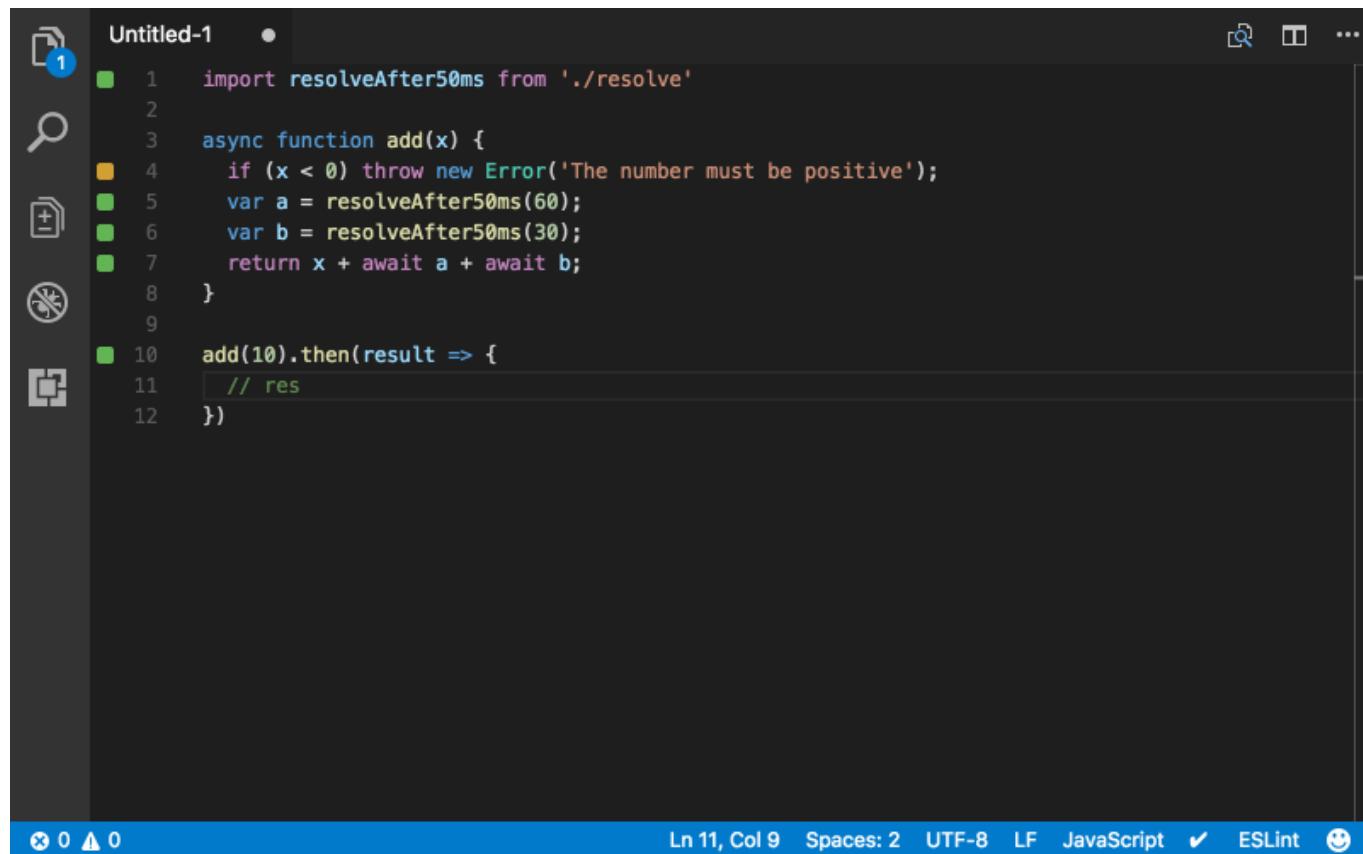
## carbon-now-sh

Send your code to [carbon.now.sh](https://carbon.now.sh).

```
index.js — website
index.js src/components/content index.js src/components/code x
1  import React from 'react';
2  import styled from 'styled-components';
3
4  const CodeBlock = styled.img`
5    max-width: 100%;
6  `;
7
8  const Container = styled.span`
9    position: absolute;
10   width: 100vw;
11   height: 100%;
12   z-index: 1;
13 `;
14
15 export default function Code({ className, ...props }) {
16   return (
17     <Container className={className}>
18       <CodeBlock {...props} />
19     </Container>
20   );
21 }
```

## Quokka

Rapid prototyping playground for JavaScript and TypeScript in VS Code, with access to your project's files, inline reporting, code coverage and rich output formatting.



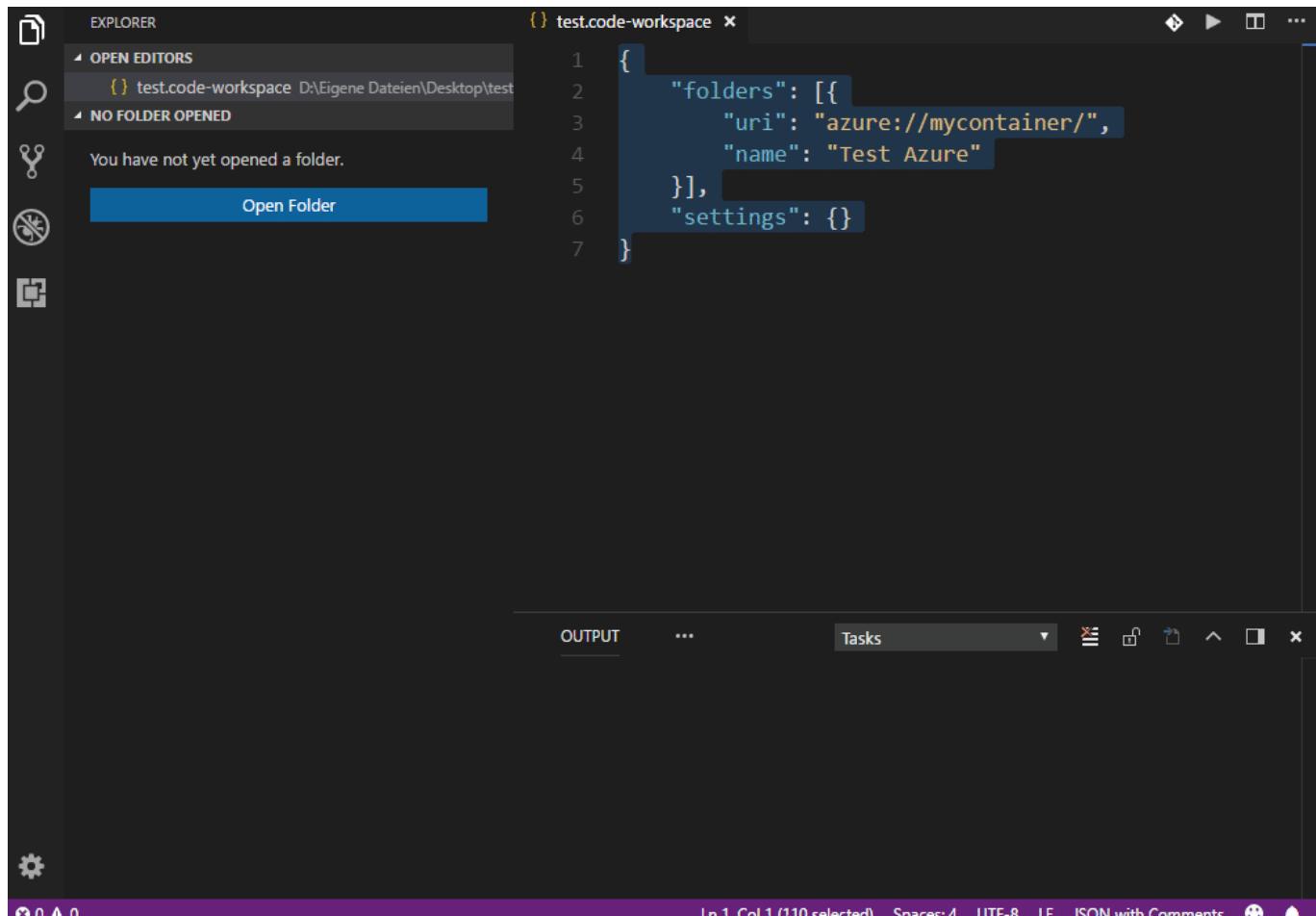
A screenshot of the Quokka extension in VS Code. The main area shows a code editor with the following content:

```
Untitled-1
1 import resolveAfter50ms from './resolve'
2
3 async function add(x) {
4     if (x < 0) throw new Error('The number must be positive');
5     var a = resolveAfter50ms(60);
6     var b = resolveAfter50ms(30);
7     return x + await a + await b;
8 }
9
10 add(10).then(result => {
11     // res
12 })
```

The status bar at the bottom indicates: Ln 11, Col 9 Spaces: 2 UTF-8 LF JavaScript ✓ ESLint 😊

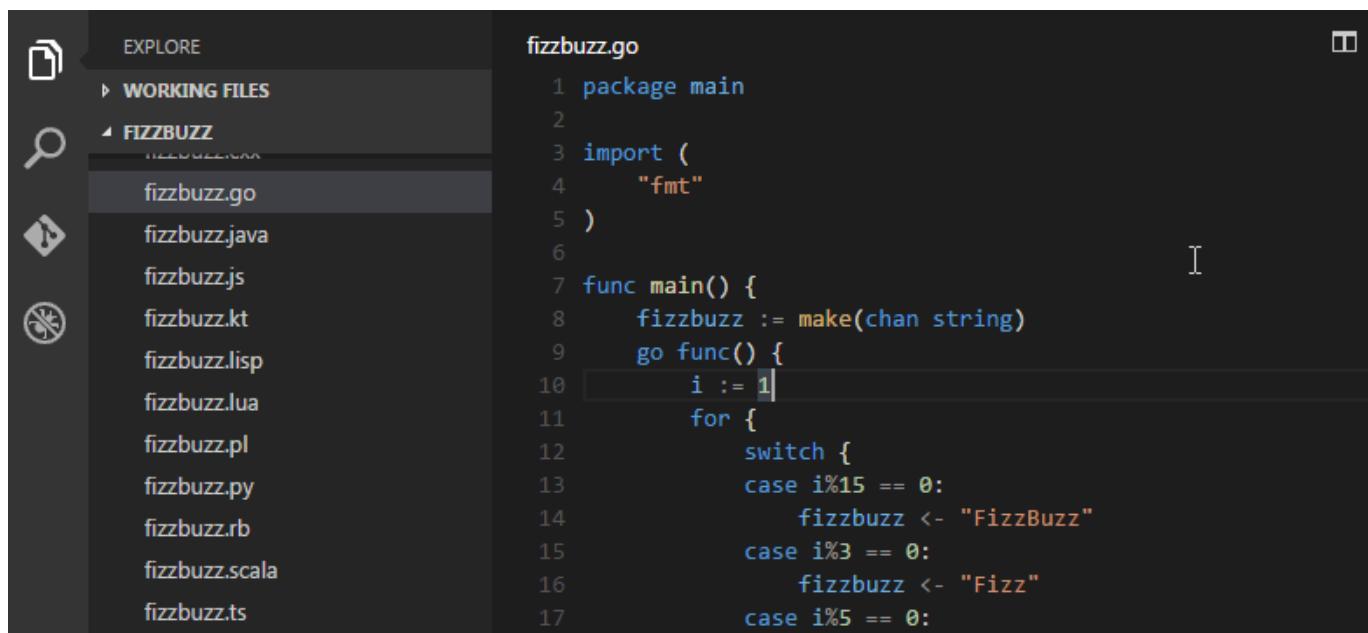
## Remote Workspace

Multi protocol support for things, like Azure blobs, S3 buckets, Dropbox, (S)FTP or WebDAV files, by using new [FileSystem API](#), especially for accessing resources like local files and folders in the editor as [workspace folders](#).



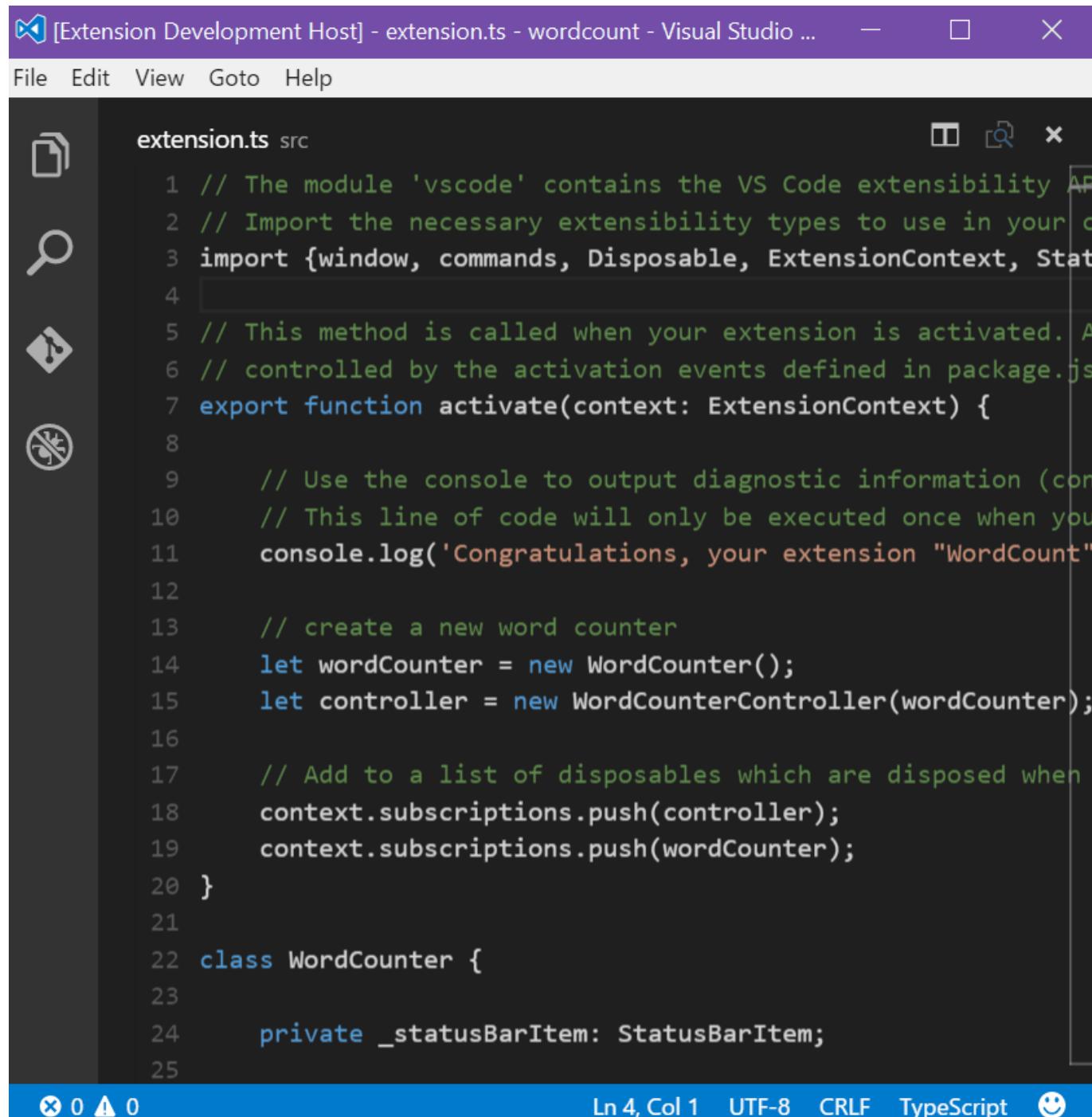
## Runner

Run various scripts right from VS Code



## Slack

Send messages and code snippets, upload files to Slack



The screenshot shows the Visual Studio Code interface with the title bar "[Extension Development Host] - extension.ts - wordcount - Visual Studio ...". The left sidebar has icons for file operations, search, and extensions. The main editor area displays the following TypeScript code:

```
extension.ts src
1 // The module 'vscode' contains the VS Code extensibility API
2 // Import the necessary extensibility types to use in your extension
3 import {window, commands, Disposable, ExtensionContext, StatusBarItem} from 'vscode'
4
5 // This method is called when your extension is activated. An extension
6 // can have multiple activation events, e.g. {fileType: 'ts', language: 'typescript'}
7 export function activate(context: ExtensionContext) {
8
9     // Use the console to output diagnostic information (optional)
10    // This line of code will only be executed once when you start your extension
11    console.log('Congratulations, your extension "WordCount" is now activated')
12
13    // Create a new word counter
14    let wordCounter = new WordCounter();
15    let controller = new WordCounterController(wordCounter);
16
17    // Add to a list of disposables which are disposed when the extension is deactivated
18    context.subscriptions.push(controller);
19    context.subscriptions.push(wordCounter);
20 }
21
22 class WordCounter {
23
24     private _statusBarItem: StatusBarItem;
25 }
```

The status bar at the bottom shows "Ln 4, Col 1" and "TypeScript".

## SmoothType

Add a smooth typing animation, similar to MS Office and the Windows 10 Mail app.

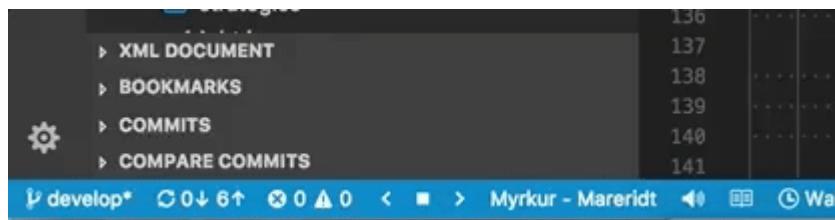
```

25   "\n\t\t" + beginComment +
26   "\n\t\t<style>.cursor { transition: all {duration}ms }</style>" +
27   "\n\t\t" + endComment;
28
29 // Pattern to check if the code is injected.
30 const injectionPattern = new RegExp("\s*" + beginComment + "(?:.|\\s)*" + endComment);
31
32 // Function to reload the window, asks and displays a message if "message" is provided.
33 You, 19 minutes ago | 1 author (You)
34 function reloadWindow(message = null) {
35   if (message === null || message === false)
36     vscode.commands.executeCommand("workbench.action.reloadWindow");
37   else vscode.window.showInformationMessage(message, {
38     title: "Reload Window"
39   }).then(clicked => {
40     if (clicked !== undefined) reloadWindow();

```

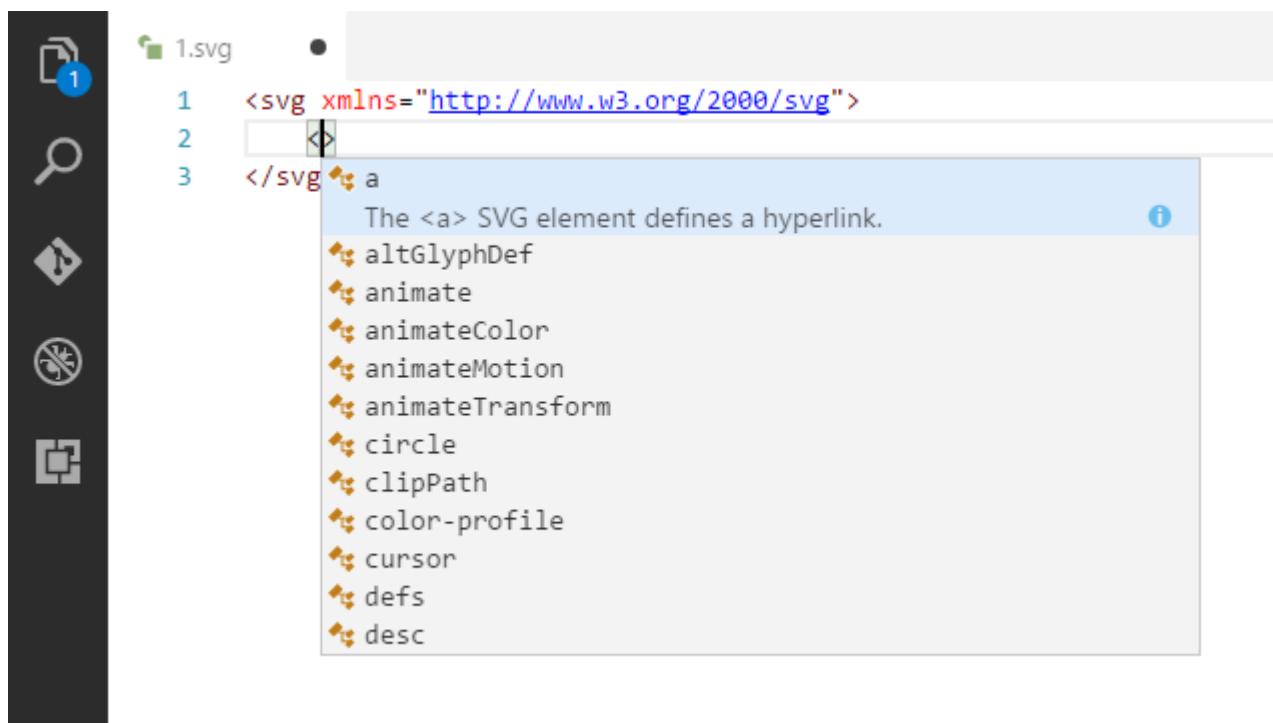
## Spotify

Provides integration with Spotify Desktop client. Shows the currently playing song in status bar, search lyrics and provides commands for controlling Spotify with buttons and hotkeys.



## SVG

A Powerful SVG Language Support Extension(beta). Almost all the features you need to handle SVG.



## SVG Viewer

View an SVG in the editor and export it as data URI scheme or PNG.



## Text Marker (Highlighter)

Highlight multiple text patterns with different colors at the same time. Highlighting a single text pattern can be done with the editor's search functionality, but it cannot highlight multiple patterns at the same time, and this is where this extension comes handy.

```
regex.test.js - vscode-text-marker
roggle-case-sensitivity-mode.test.js - vscode-text-marker
```

```
const RegexPattern = require('../lib/patterns/regex');
const StringPattern = require('../lib/patterns/string');

suite('RegexPattern', () => {
    test('it returns list of ranges which locates all the strings match the g', () => {
        const pattern = new RegexPattern({
            phrase: 'Zx'
        });
        const ranges = pattern.locateIn('ENTIRE TEXT z Zx');
        expect(ranges).to.eql([
            {start: 12, end: 13},
            {start: 14, end: 16}
        ]);
    });

    test('it returns list of ranges which locates all the strings match the g', () => {
        const pattern = new RegexPattern({
            phrase: 'Zx',
            ignoreCase: true
        });
        const ranges = pattern.locateIn('ENTIRE TEXT z Zx');
        expect(ranges).to.eql([
            {start: 12, end: 13},
            {start: 14, end: 16}
        ]);
    });

    test('it finds all matches with whole word search', () => {
        const pattern = new RegexPattern({
            phrase: 'text',
            wholeMatch: true
        });
        const ranges = pattern.locateIn('ENTIRE texto text');
        expect(ranges).to.eql([
            {start: 13, end: 17}
        ]);
    });

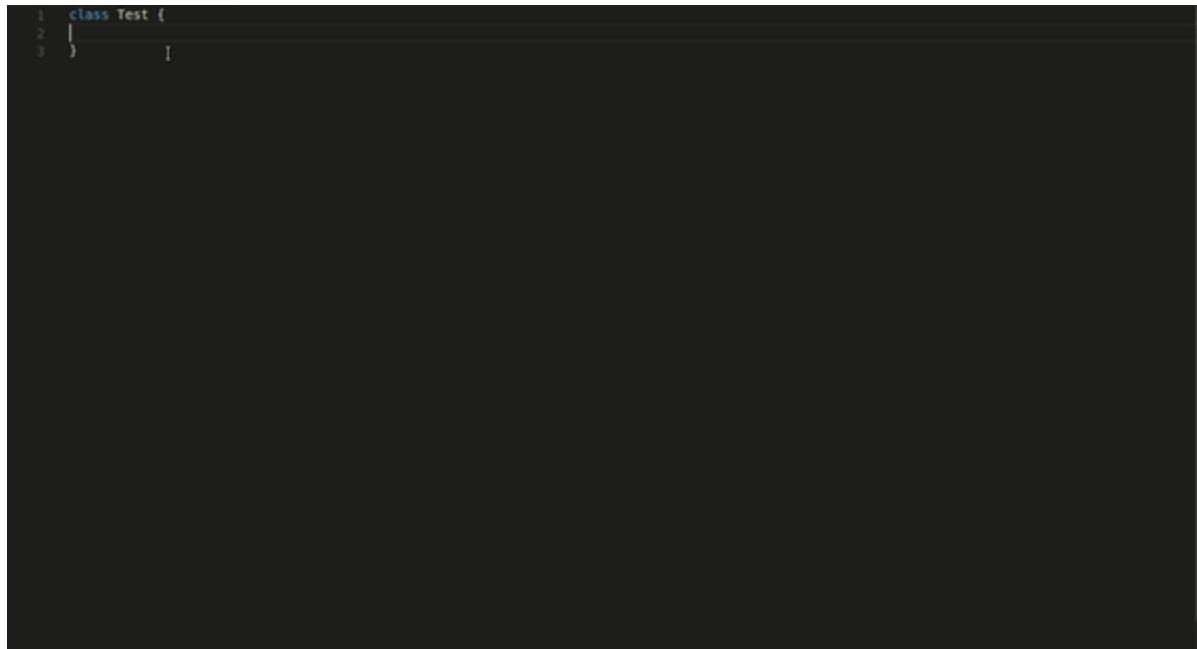
    test('it finds all matches with whole word search - edge of the whole text', () => {
        const pattern = new RegexPattern({
            phrase: 'text'
        });
        const ranges = pattern.locateIn('ENTIRE texto text');
        expect(ranges).to.eql([
            {start: 13, end: 17}
        ]);
    });
});

suite('ToggleCaseSensitivityModeButton', () => {
    test('it initialises and shows the button once matching mode is prepared', () => {
        const eventBus = new EventEmitter();
        const statusBarItem = (show: sinon.spy) => new ToggleCaseSensitivityModeButton(eventBus, statusBarItem); // +
        eventBus.on(Event.MATCHING_MODE_INITIALISED, () => {
            expect(statusBarItem).to.contain({
                command: 'textmarker.toggleModeForCaseSensitivity',
                text: 'Ao',
                tooltip: 'TextMarker: Case Sensitive Mode'
            });
            expect(statusBarItem.show).to.have.been.called;
            done();
        });
        eventBus.emit(Event.MATCHING_MODE_INITIALISED, {ignoreCase: false});
    });

    test('it updates button appearance on receiving case sensitivity mode change', () => {
        const eventBus = new EventEmitter();
        const statusBarItem = (show: sinon.spy) => new ToggleCaseSensitivityModeButton(eventBus, statusBarItem); // +
        eventBus.on(Event.TOGGLED_CASE_SENSITIVITY, () => {
            expect(statusBarItem).to.contain({
                text: 'Aa',
                tooltip: 'TextMarker: Case Insensitive Mode'
            });
            expect(statusBarItem.show).to.not.have.been.called;
            done();
        });
        eventBus.emit(Event.TOGGLED_CASE_SENSITIVITY, {ignoreCase: true});
    });
});
```

ESDOC MDN

Quickly bring up helpful MDN documentation in the editor



regex.test.js

```

1 class Test {
2 }
3 }

const RegexPattern = require('.././lib/patterns/regex');
const StringPattern = require('.././lib/patterns/string');

suite('RegexPattern', () => {
  test('it returns list of ranges which locates all the strings match the g', () => {
    const pattern = new RegexPattern({
      phrase: 'Zé'
    });
    const ranges = pattern.locateIn('ENTIRE TEXT z ZZ');
    expect(ranges).to.eql([
      {start: 12, end: 13},
      {start: 14, end: 16}
    ]);
  });

  test('it returns list of ranges which locates all the strings match the g', () => {
    const pattern = new RegexPattern({
      phrase: 'Zé',
      ignoreCase: true
    });
    const ranges = pattern.locateIn('ENTIRE TEXT z ZZ');
    expect(ranges).to.eql([
      {start: 12, end: 13},
      {start: 14, end: 16}
    ]);
  });

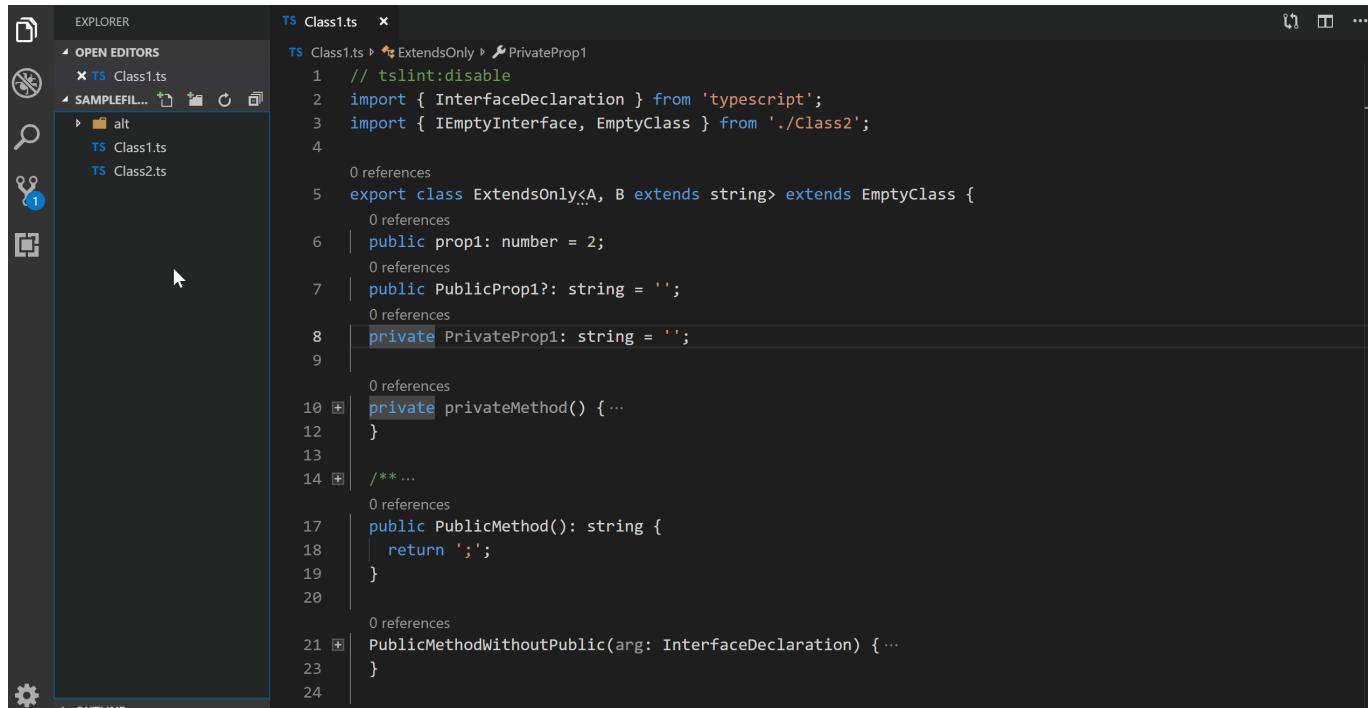
  test('it finds all matches with whole word search', () => {
    const pattern = new RegexPattern({
      phrase: 'text',
      wholeMatch: true
    });
    const ranges = pattern.locateIn('ENTIRE texta text');
    expect(ranges).to.eql([
      {start: 13, end: 17}
    ]);
  });

  test('it finds all matches with whole word search - edge of the whole text', () => {
    const pattern = new RegexPattern({
      phrase: 'text'
    });
    const ranges = pattern.locateIn('ENTIRE texta text');
    expect(ranges).to.eql([
      {start: 13, end: 17}
    ]);
  });
});

toggle-case-sensitivity-mode.test.js
```

## Interface generator

Quickly generate interface definitions from typescript class



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left lists files: 'TS Class1.ts' (open), 'TS Class2.ts', and 'SAMPLEFILE...'. The Editor tab 'TS Class1.ts' displays the following TypeScript code:

```
TS Class1.ts x
TS Class1.ts > ExtendsOnly > PrivateProp1
1 // tslint:disable
2 import { InterfaceDeclaration } from 'typescript';
3 import { IEmptyInterface, EmptyClass } from './class2';
4
5 0 references
6 export class ExtendsOnly<A, B extends string> extends EmptyClass {
7   0 references
8     public prop1: number = 2;
9     0 references
10    public PublicProp1?: string = '';
11    0 references
12    private PrivateProp1: string = '';
13
14    0 references
15    private privateMethod() { ... }
16
17    /** ...
18     * @public */
19    public PublicMethod(): string {
20      return '';
21    }
22
23    0 references
24    PublicMethodWithoutPublic(arg: InterfaceDeclaration) { ... }
```

# Themes

---

## UI

If you're feeling brave, there are at least two custom UIs for VS Code:

- [Essence](#)
- [Pragmatic Essence](#)

## Syntax

### An Old Hope Theme by Dustin Sanders

VSCode theme inspired by a galaxy far far away...



### Ariake Dark by wart

Dark VSCode theme inspired by Japanese traditional colors and the poetry composed 1000 years ago.



### Atom One Dark Theme by Mahmoud Ali

One Dark Theme based on Atom.



### Ayu by teabyii

A simple theme with bright colors and comes in three versions — dark, light and mirage for all day long comfortable work.



## [Captain Sweetheart by ultradracula](#)

Tuff but sweet theme.



## [City Lights by Yummypum](#)

□ Yummypum's Official City Lights suite



## [Cobalt2 Theme Official by Wes Bos](#)

⌚ Official theme by Wes Bos.



## [Dracula Official by Dracula Theme](#)

Official Dracula Theme. A dark theme for many editors, shells, and more.



## [Edge by Bogdan Lazar](#)

A simple theme with bright colors in three variants — Night Sky, Serene and Ocean for all day long comfortable work.



## [Eva Theme by fisheva](#)

A colorful and semantic coloring code theme.



## [Fairy Floss by nopjmp and sailorhg](#)

A fun, purple-based pastel/candy/daydream fairyfloss theme made by sailorhg.



## [GitHub Theme by Thomas Pink](#)

GitHub Theme for Visual Studio Code.



## [Material Palenight Theme by whizkydee](#)

An elegant and juicy material-like theme for Visual Studio Code.



## [Material Theme by Mattia Astorino](#)

The most epic theme now for Visual Studio Code.



## [Mno by u29dc](#)

Minimal monochrome theme.



## [Monokai Oblique by pushqrxd](#)

Monokai inspired theme for [Visual Studio Code](#) and [Visual Studio IDE](#).



## [Monokai Pro by monokai \(commercial\)](#)

Beautiful functionality for professional developers, from the author of the original Monokai color scheme.



## [Night Owl by Sarah Drasner](#)

A VS Code theme for the night owls out there. Works well in the daytime, too, but this theme is fine-tuned for those of us who like to code late into the night. Color choices have taken into consideration what is accessible to people with color blindness and in low-light circumstances. Decisions were also based on meaningful contrast for reading comprehension and for optimal razzle dazzle. 



## [Plastic by Will Stone](#)

A simple theme.



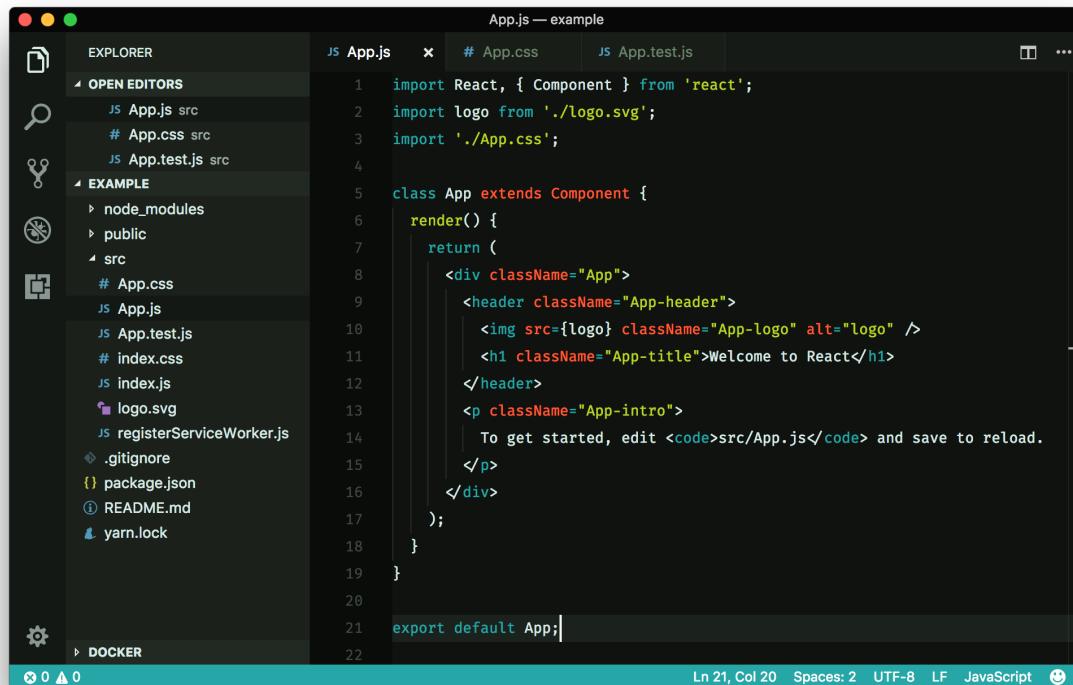
## [Nord by arcticicestudio](#)

An arctic, north-bluish clean and elegant Visual Studio Code theme.



## Rainglow by Dayle Rees

Collection of 320+ beautiful syntax and UI themes.



## Relaxed Theme by Michael Kühnel

A relaxed theme to take a more relaxed view of things.



## Shades of Purple by Ahmad Awais

⚡ A professional theme with hand-picked & bold shades of purple ❤️ to go along with your VS Code. A custom VS Code theme with style.



## Slime Theme by smlombardi

A dark syntax/workbench theme for Visual Studio Code - optimized for SCSS, HTML, JS, TS, Markdown, and PHP files.



## Niketa Theme by Dejan Toteff

Collection of 18 light themes separated in 4 groups by background's brightness.



# People to Follow

---

A list of Twitter accounts for various people in the VS Code Community

- [@code](#) - The official VS Code Twitter
- [@auchenberg](#) - VS Code Program Manager
- [@BenjaminPasero](#) - VS Code Dev
- [@chrisdias](#) - VS Code Program Manager
- [@\\_clarkio](#) - Developer Advocate @ Azure. Creator of VS Code release highlight videos
- [@eamodio](#) - GitLens creator
- [@ErichGamma](#) - VS Code Dev
- [@IsidorN](#) - VS Code Dev
- [@joaomoreno](#) - VS Code Dev
- [@johannesrieken](#) - VS Code Dev
- [@lannonbr](#) - Creator of vscode.rocks & JS Parameter Annotations extension
- [@maeschli](#) - VS Code Dev
- [@mattbierner](#) - VS Code Dev
- [@MrAhmadAwais](#) - JS/WordPress Core Dev. Creator of VSCode.pro course & Shades of Purple theme
- [@ramyanexus](#) - VS Code Dev. Maintainer of Go extension
- [@Tyriar](#) - VS Code Dev. Creator of xterm.js

## Resources for extension developers

---

### Documentation

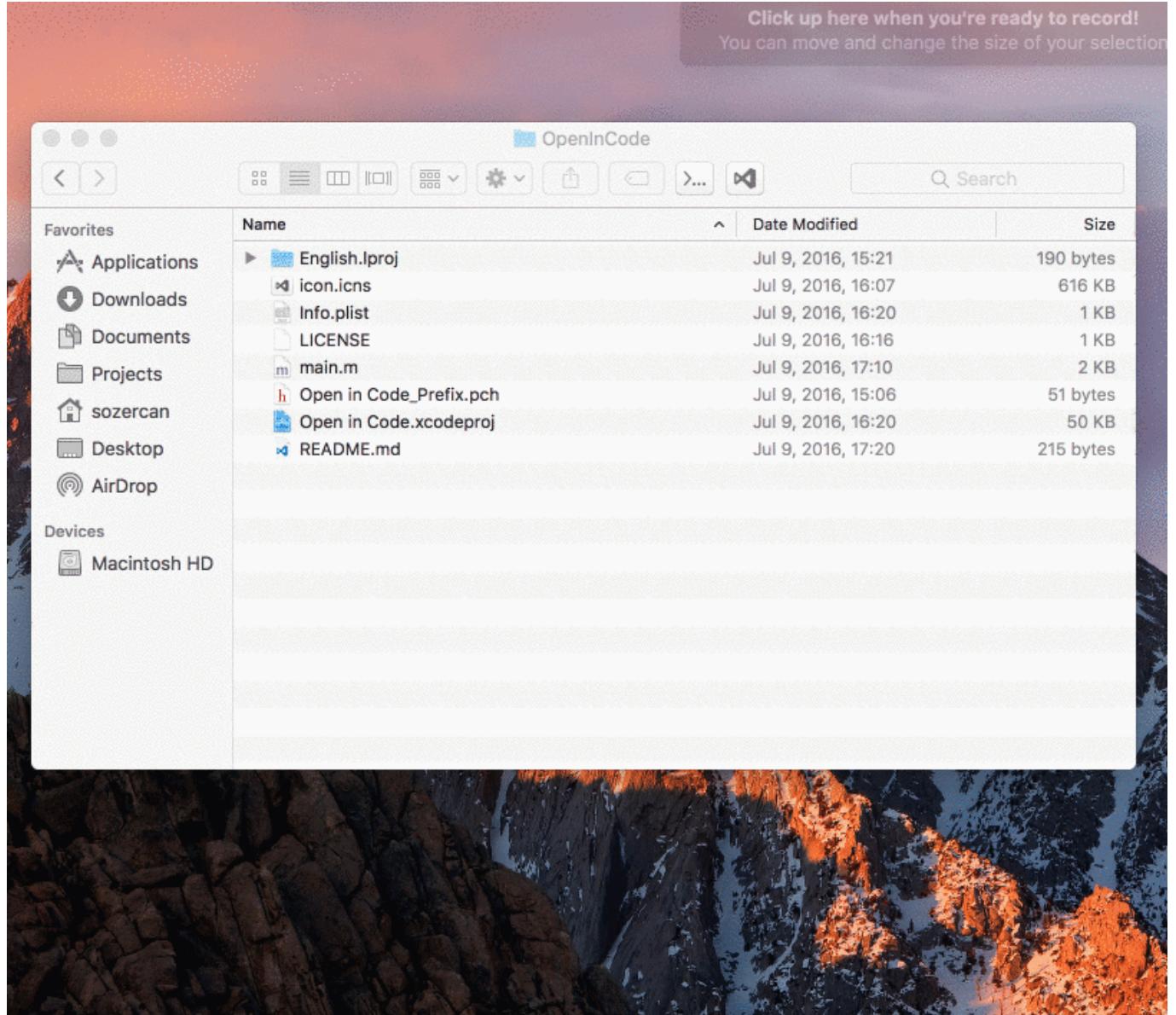
- [Better Comments](#) - The Better Comments extension will help you create more human-friendly comments in your code.
- [Extending Visual Studio Code](#) section of [Official Documentation](#)

### Libraries

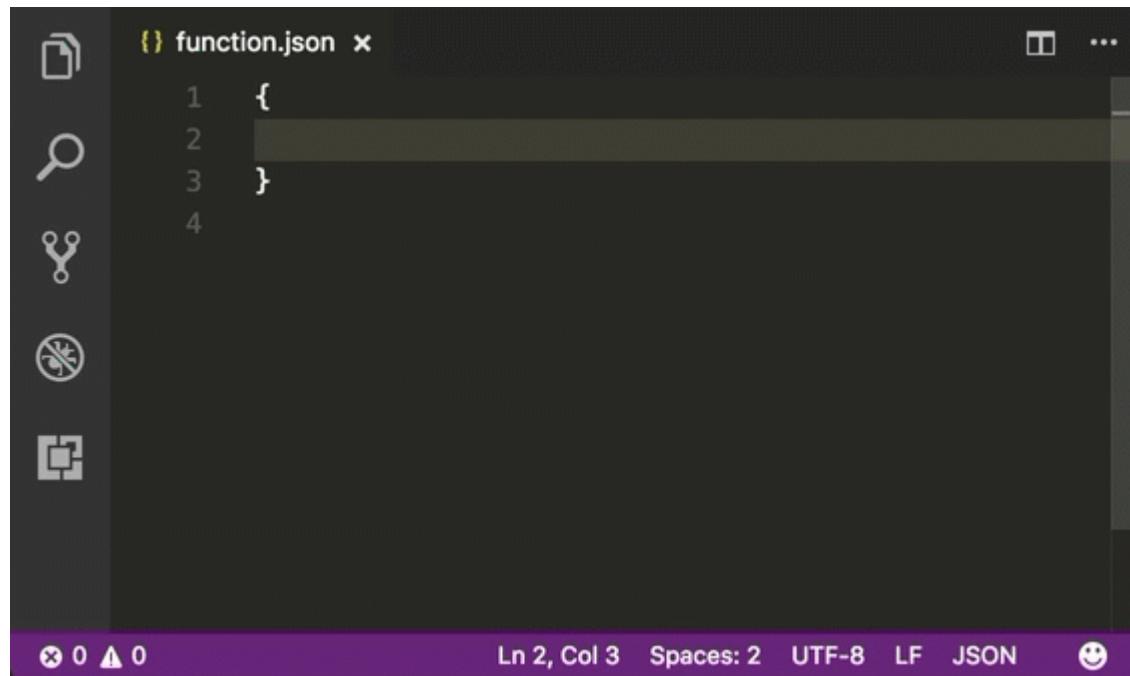
- [vscode-extension-vscode](#) - The vscode.d.ts node module
- [vscode-set-text](#) - An easy-to-use VS Code `#setText()` method.
- [vscode-test-content](#) - A method to set/get editor content, and its selection. Especially useful for unit tests.

### Tools

- [Live Share](#) - Visual Studio Live Share enables you to collaboratively edit and debug with others in real time, regardless what programming languages you're using or app types you're building.
- [Online TextMate Themes Editor](#) - since VS Code supports TextMate themes, you can create them in this online editor and then create a new VS Code package using [Yo Code](#) tool
- [Yo Code - Extension Generator](#)
- [Open in Code](#) - macOS Finder toolbar app to open current folder in Visual Studio Code



- [Themer](#) - Easily create your own theme for VS Code (and matching ones for your other tools).
- [Azure Tools for Visual Studio Code](#) - This extension for Visual Studio Code gives Azure developers some convenient commands for creating or accessing resources directly in the editor.



- [Mark down preview](#) - Markdown Preview Enhanced is an extension that provides you with many useful functionalities such as automatic scroll sync, math typesetting, mermaid, PlantUML, pandoc, PDF export, code chunk, presentation writer, etc. A lot of its ideas are inspired by Markdown Preview Plus and RStudio Markdown.

## Online Courses

---

### Visual Studio Code Power User Course (commercial)

After 10 years with Sublime Text, [Ahmad Awais](#) switched to VSCode, built [Shades of Purple theme](#), and spent over a 1,000 hours perfecting his setup. He has launched a VSCode Power User Course to help you switch today. You can bring all your custom settings from the previous editor and learn the power user workflows for HTML/CSS, Git/GitHub & Open Source, supercharged Markdown, and everything from JavaScript to PHP, Go, Python, C++, C#, ROR. In this course, you'll also learn to install/setup 50+ Extensions.

- [VSCode.pro](#) - 📺 Huge five hours 65 videos VSCode course.
- [Ahmad Awais](#) - 🤝 Creator of Shades of Purple theme. Core Developer for WP/JS. OSS Dev Advocate.

## Contribute

---

Contributions welcome! Read the [contribution guidelines](#) first.

## License

---

I am providing code and resources in this repository to you under an open source license. Because this is my personal repository, the license you receive to my code and resources is from me and not my employer (Microsoft).



To the extent possible under law, [Valerii Iatsko](#) has waived all copyright and related or neighboring rights to this work.