

VSCode Journey

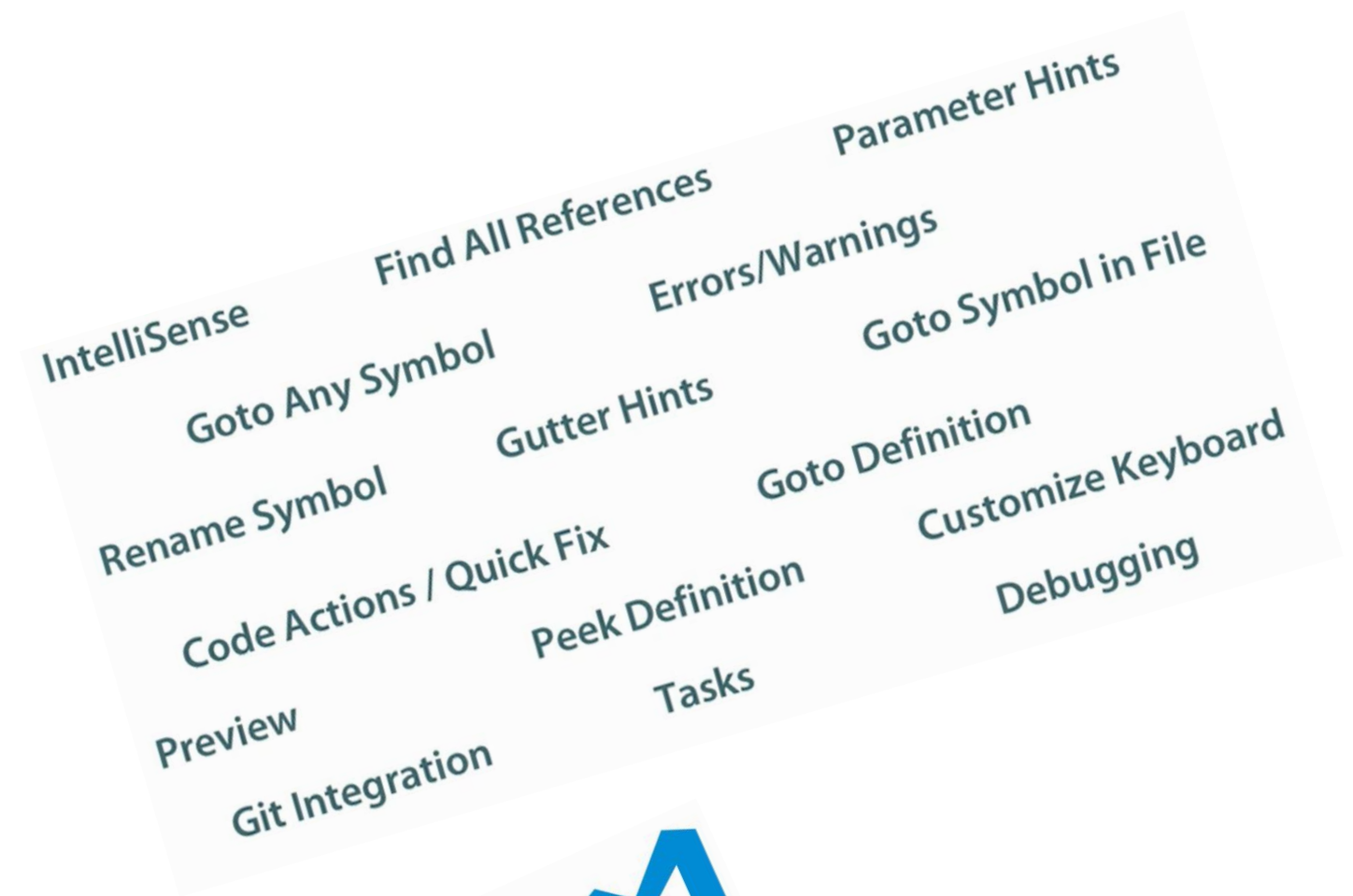
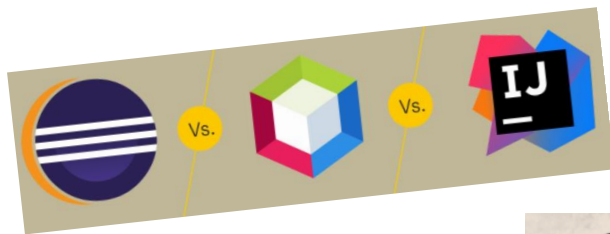
Azat Satklichov

azats@seznam.cz,

<http://sahet.net/htm/java.html>

Agenda

- ☐ Why VSCode
- ☐ Getting Started
- ☐ Some Refactorings
- ☐ Debugging Experience
- ☐ Themes, Preferences
- ☐ Keyboard Shortcuts
- ☐ Font Ligatures
- ☐ Handy Extensions



Why VSCode

- ❖ Fast, Lightweight and Powerful and Cross Platform (Mac and Win, ..)
- ❖ Coding Fast, multi-instance, customizable

How VSCode designed - 3 main parts

- ✓ UI Framework ([Electron](#)[Atom Shell], e.g. Atom builds on it)
- ✓ Editor (Monaco) (also used in IE, OneDrive, ..)
- ✓ Intellisense, Tooling, Debugging [for lang. features] (Typescript server ..)

Comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Visual studio Code a **new choice** of tool that **combines** the **simplicity** of a code editor with what **developers need** for their code-editor-debug cycle.

- **Eric Gamma (GoF Author)**

2011

"Monaco" Workbench



2013

Visual Studio Online "Monaco"



2014

From the Browser to the Desktop



Electron

2015

//build [May] VS Code Preview



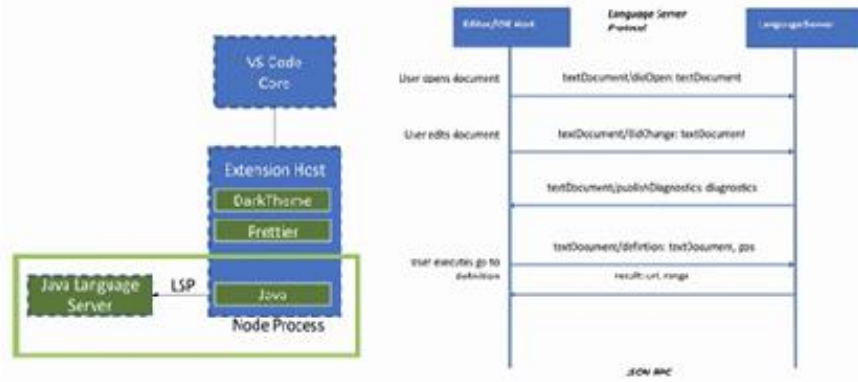
2015

Connect() [November] Extension API



2016

LSP - Language Server Protocol



Getting Started

>code .
>code-insiders .

Single and Multi Instances

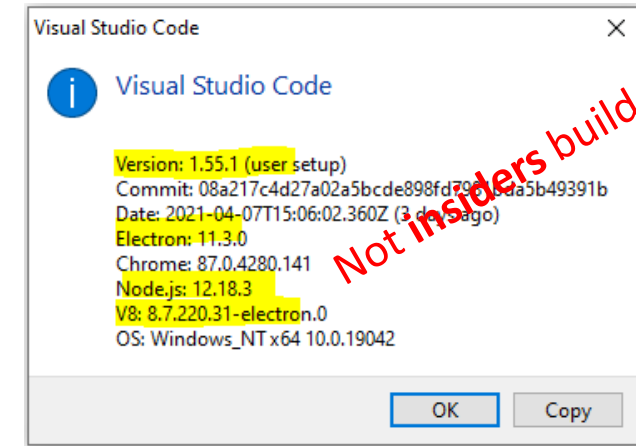
- code .
- code about.txt -r
- code -g caw.cbl:163:23
- code -n --goto C:\workspace\che-che4z-lsp-for-cobol\clients\cobol-lsp-vscode-extension\Readme.md:112:70
- code index.html style.css doc\readme.md //like in Linux
- code --telemetry, code --help (lang, disable extension, diff, ...)
- code C:\workspace-JavaNew\ilki -n

Tweet us your feedback.

How was your experience?

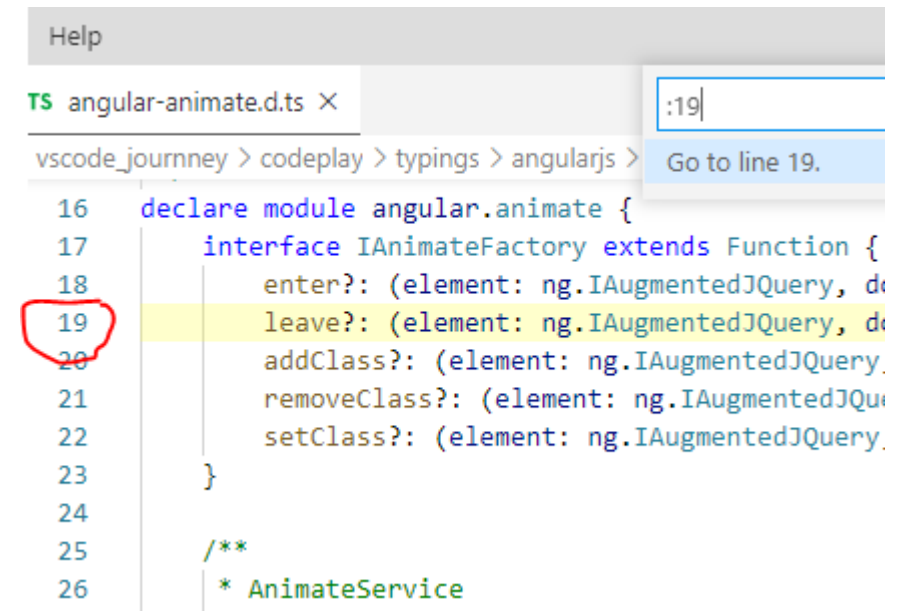
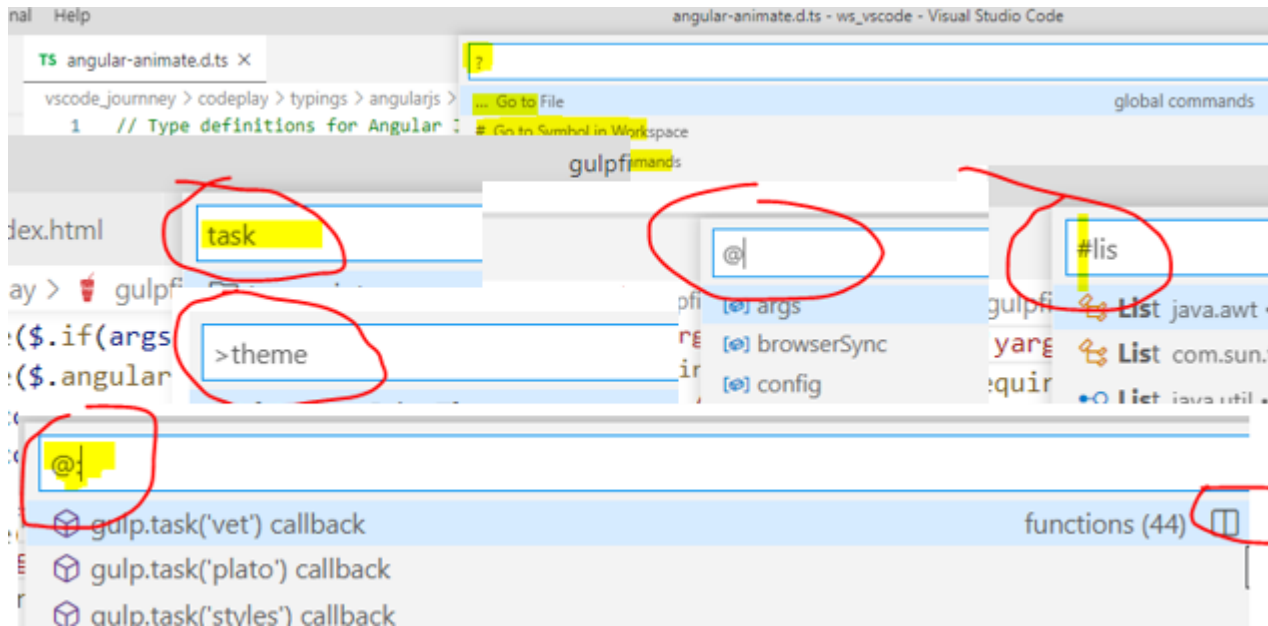


Ot
Su
n-



Getting Started

- ❑ **Keystrokes** - Just remember File Navigation (CMD + P), (CMD+PP), type **?, :, #, @, @:, !, >, >xzy**, or (CMD + SHIFT + P)
@ or @:<editor>. CMD and ,+. <num>



Some Refactorings

- Language Features (Intellisense, coloring, linting, outline, Emmets)
- Layout, Explorer, Editing, Saving, Searching, File Nav., Command P.

Bracket matching

Selection

Cursors

Multi cursors (ctrl+F2, alt+click, progressive
selection: ctrl+D, skip: ctrl+K) [shift+alt] vert

Intellisense

Parameter hints

Emmet

[Emmet](#) (e.g. html,css) **html:5**, **div>ul>li*5**, | **div.color**, | **div#idm**, (**JSish**, **React**), **image-tricks**, custom emmets

Snippets

emmets not for Angular, COBOL, .. then use Snippets. Also **share it**: ~\User\snippets | local)

Go to definition or symbol

Gutter indicators

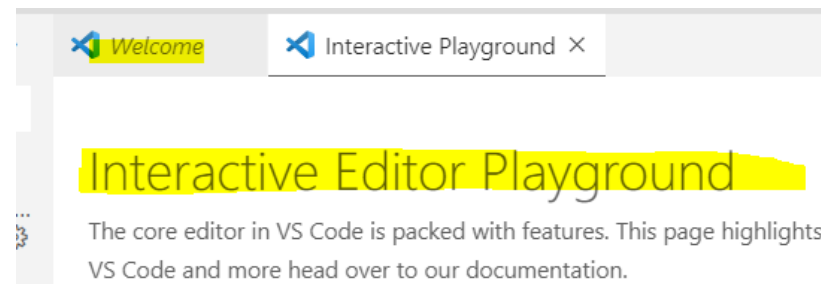
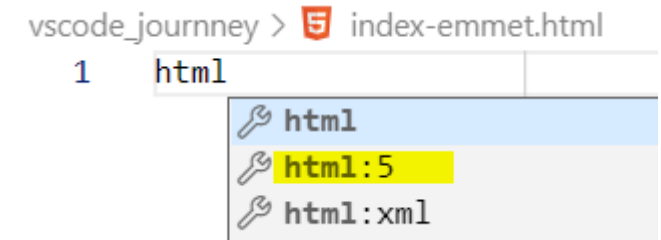
Peek

Hover

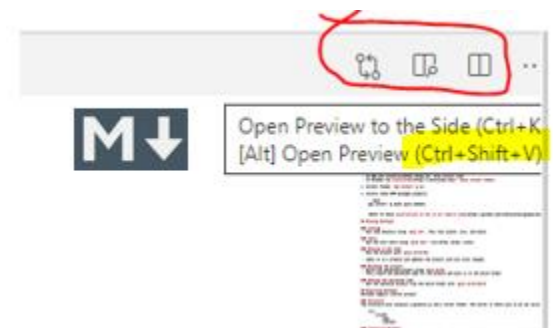
Renaming

Code actions

Errors / Warnings



Context Sensitive buttons



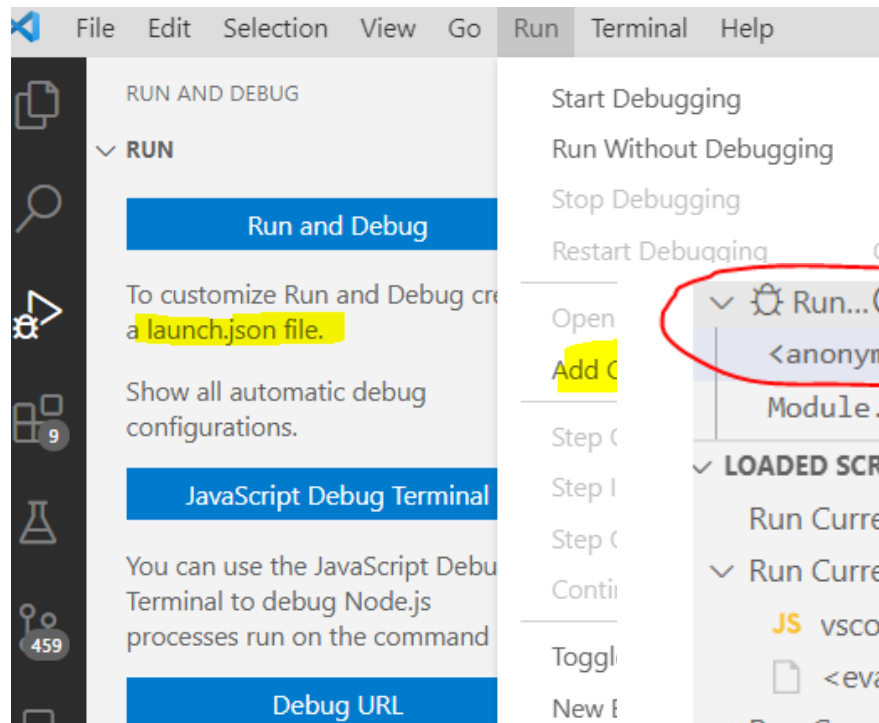
Debugging Experience

VSCode default TS Compiler: `@ts-check`: or per-prj `implicitProjectConfig.checkJS: true`

```
index.js
1 // @ts-check
2
3 let rick = () => {
4   var messages = [
10
11   for (i = 0; i
12     |
13   }
    cons
```

Debugger Extensions

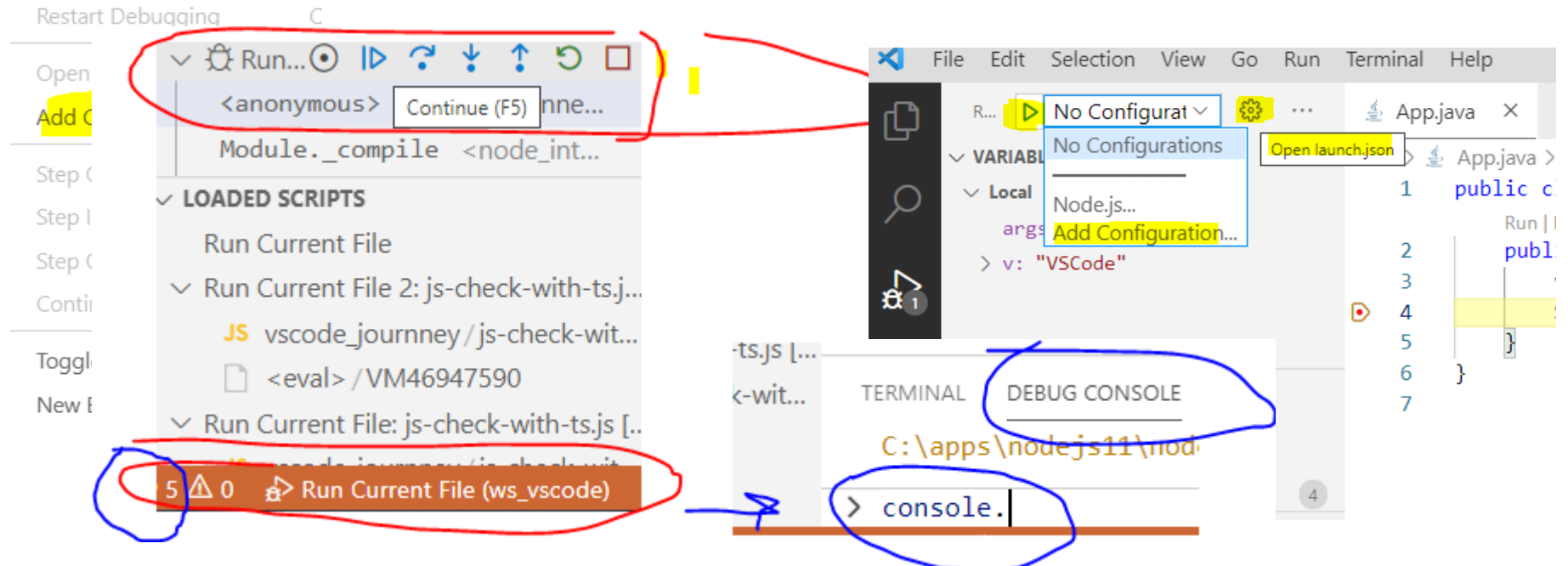
- Built-in debugging support for the [Node.js](#) runtime and can debug JS, TS, or any other that gets transpiled to JS
- To debug other languages and runtimes either search extensions or from menu **Run -> Install Additional Debuggers**



Launch Configuration (compound, .. DevTool-F12, ..)

Create "launch.json" either by 'gear-icon' or Run->Add Conf.

- E.g. - Add "**stopOnEntry**": true
- Drag&drop for Execution point



Themes, Preferences

Themes

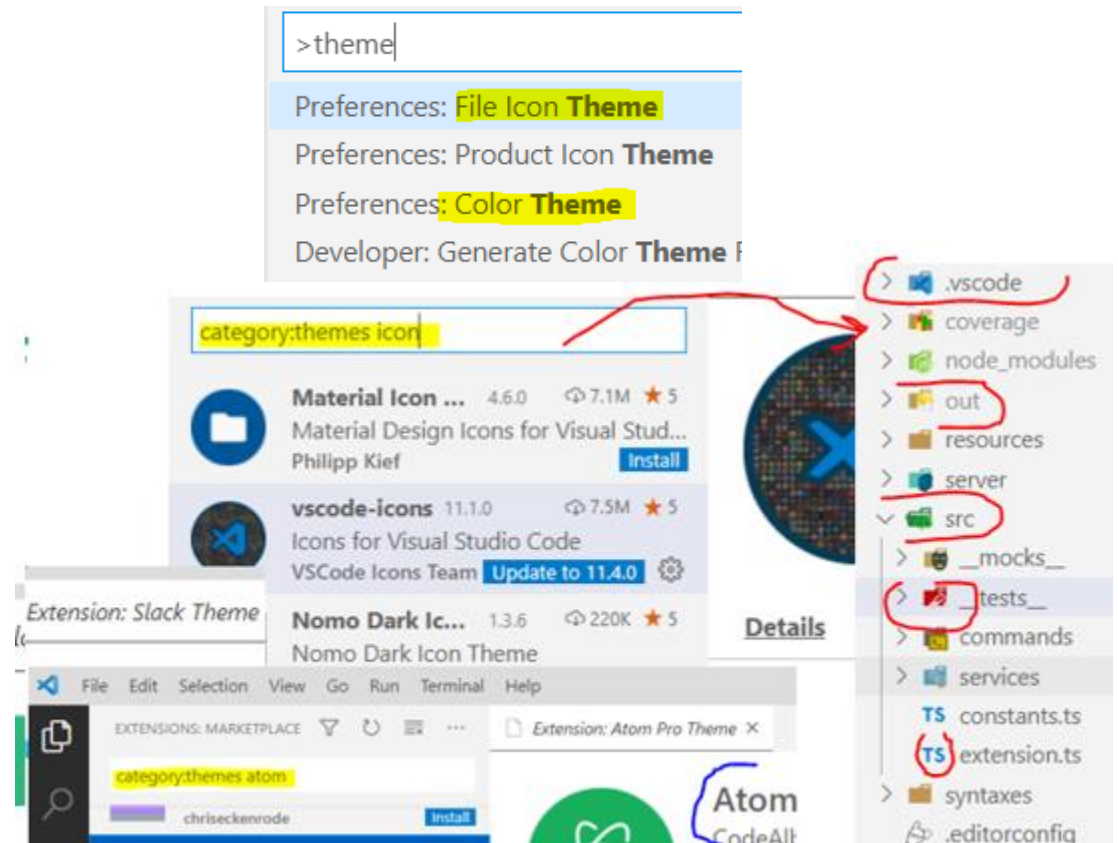
(**CMD + SHIFT + P**) – Themes, icon themes
or just install from [themes](#) (on installed theme)

Customize Preferences (Settings)

When overriding, find default values by GUI

Settings precedence: Note that user-settings overrides default, and workspace-settings override user-setting.

Default	User	Workspace
Come with Code	All instances of Code	Per workspace
	~\settings.json	.vscode\settings.json



Note: VS Code extensions can also add their own custom settings and they will be visible under an **Extensions** section. E.g. **LSP** ;)

Keyboard shortcuts

File -> Preferences -> Keyboard Shortcuts, or search

Keybindings: Default, and Custom

Note: Check **conflicts** and use those extra (not binded) ones

Rules: (key, when(debug, in Development, ...), command)

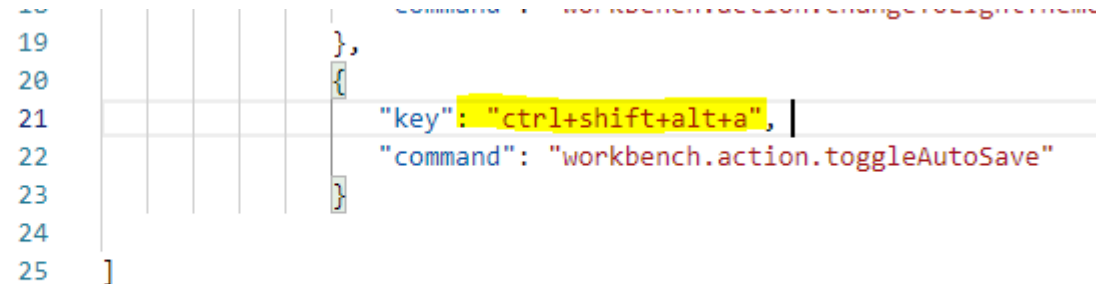
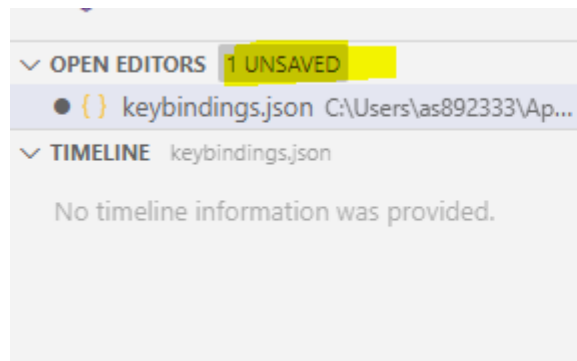
ctrl+shift+u

ctrl+shift+l

```
{ "key": "F5",  
  "when": "inDebugMode",  
  "command": "workbench.action.debug.play"}
```

```
{ "key": "ctrl+shift+alt+a",  
  "command": "workbench.action.toggleAutoSave" }
```

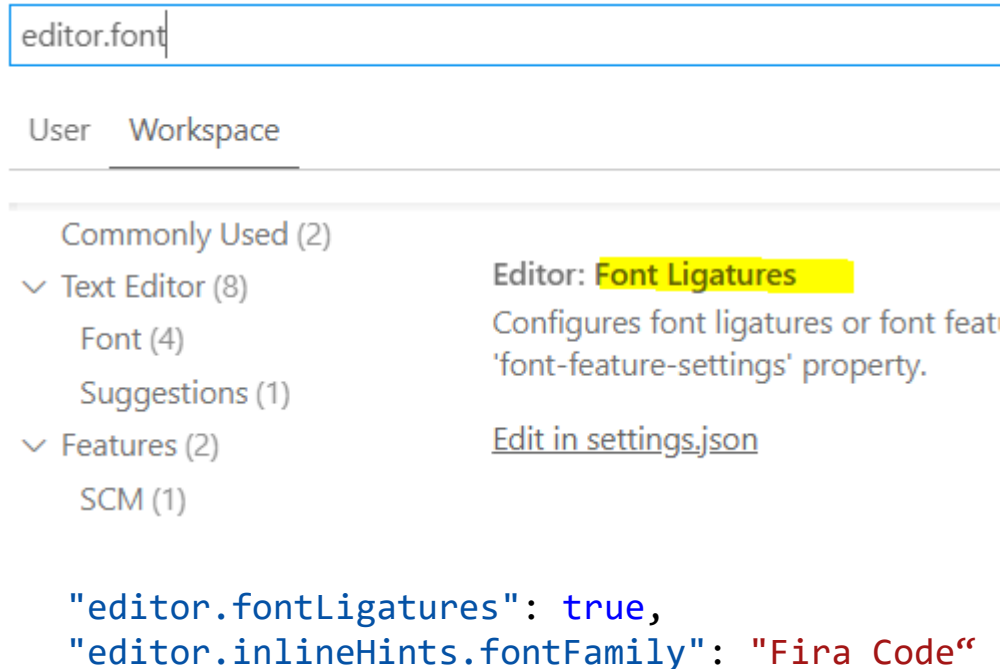
C:\Users\as892333\AppData\Roaming\Code\User\keybindings.json



Font Ligatures (custom fonts)

Search **FiraCode**: <https://github.com/tonsky/FiraCode>, <https://fonts.google.com/specimen/Fira+Code>

In [writing](#) and [typography](#), a **ligature** occurs where two or more [graphemes](#) or letters are joined as a single [glyph](#).



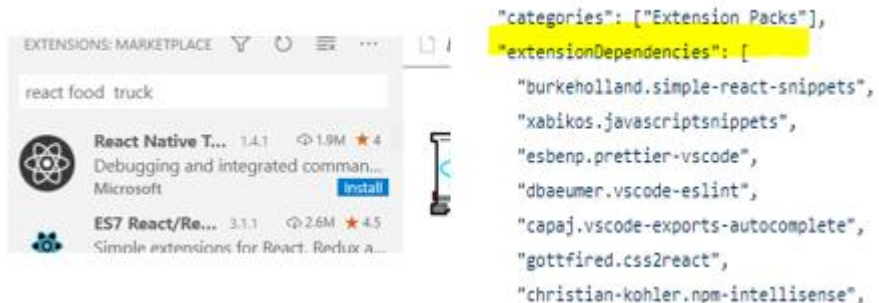
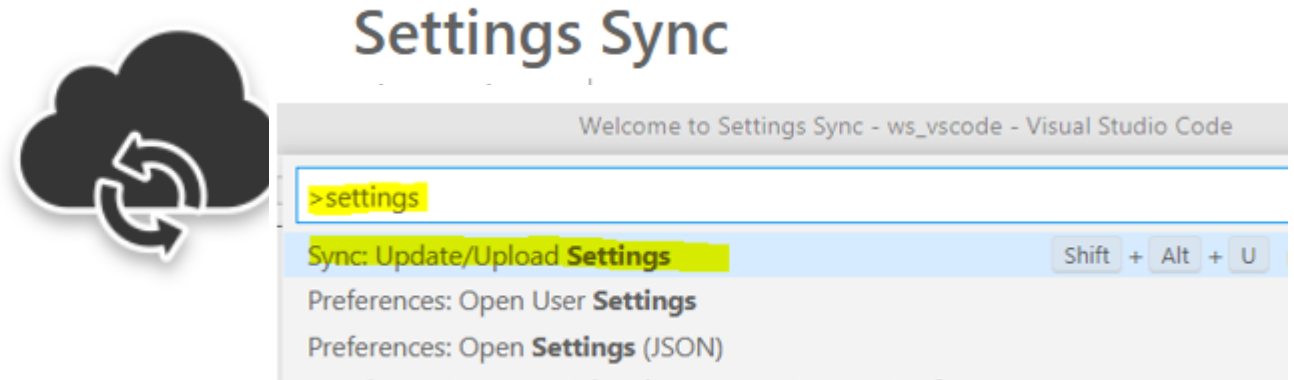
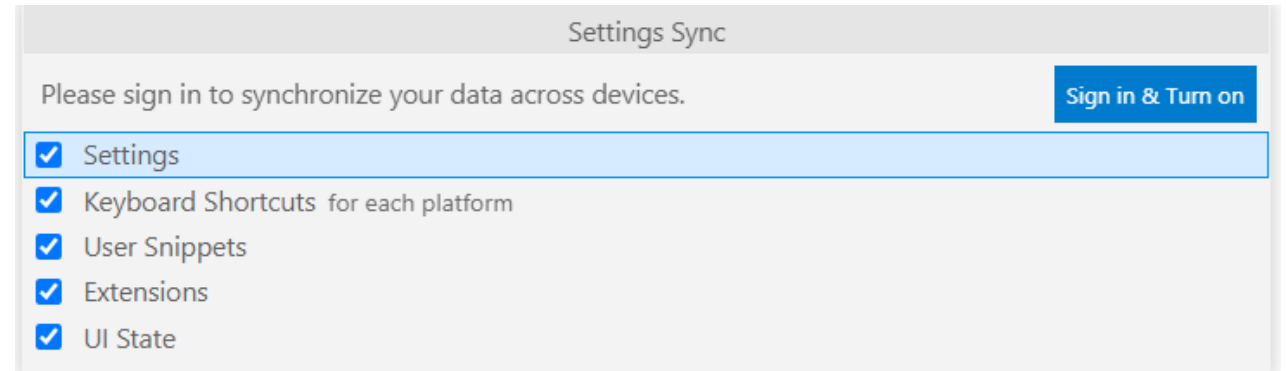
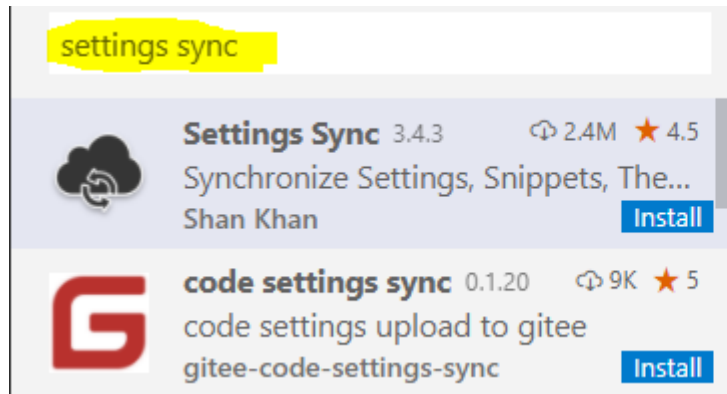
e.g. fat arrows, 2equals, != (**≠**), == (**≡**) ...

```
67 |         return result["items"].map((i: any) => i.member);
68 |     } catch (error) {
69 |         throw convertError(error);
70 |     }
71 | }
72 |
73 | public async createSession(profileName: string) {
74 |     const profile = (await new BasicProfileManager(this
75 |     if (profile.password === "") {
76 |         throw new ZoweError("No password", Type.NoP
77 |     }
```

Even if you share your settings, code will be seen normally
Ligatures not impact

Shareables (snippets, themes, font-size, ..) - Ready Development Environment

- Just share path, settings file, copy+paste, ..
- Turn on Settings Sync
- Settings Sync extension
- Special Extension pack dedicated for that



```
"categories": ["Extension Packs"],  
"extensionDependencies": [  
  "burkeholland.simple-react-snippets",  
  "xabikos.javascriptsnippets",  
  "esbenp.prettier-vscode",  
  "dbaeumer.vscode-eslint",  
  "capaj.vscode-exports-autocomplete",  
  "gottfried.css2react",  
  "christian-kohler.npm-intellisense",  
]
```

Handy Extensions

- **awesome-vscode (e-pack)**
- Prettier, Debugger for Chrome
- ESLint, TSLint
- **Peacock**
- Docker
- Vscode-icons
- **Antlr**
- Angular v5 Snippets for VS Code
- Vue extension depends on Vetur
- Wallaby
- Chronicler
- SQLTools
- REST Client
- Settings Sync
- **Live Share**

index.js
"javascript.implicitProjectConfig.checkJs": true
//@ts-check
let rick = () => {
var messages = [



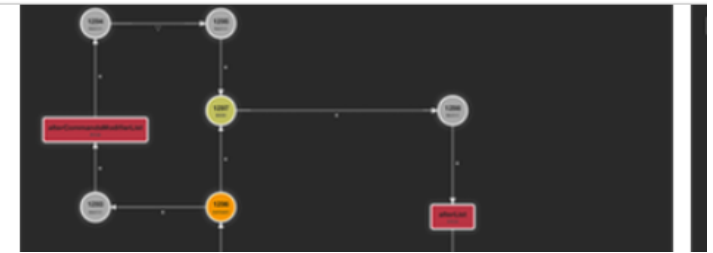
ANTLR4 grammar syntax support

Mike Lischke | 54,296 | ★★★★★ | Repository

Language support for ANTLR4 grammar files

[Set Color Theme](#) [Disable](#) [Uninstall](#) [This extension is enable](#)

[Details](#) [Feature Contributions](#)



Wallaby.js

Wallaby.js | 4,509,238 | ★★★★★

Accelerated Distractio

wallabyjs.wallaby-vscode



Vue V!

sarah.dra

Snippets th

[Install](#)

Contributions



Live Share

ms-vsliveshare.vsliveshar

Microsoft | 4,509,238 | ★★★★★

Real-time collaborative development from the cc

[Update to 1.0.4131](#) [Disable](#) [Uni](#)

[HTTPS_PROXY=http://proxy-ip-address:proxyport](#)

//If you have an authenticating proxy

[HTTPS_PROXY=http://user:password@proxy-ip-address:proxyport](#)



THANK YOU

References

<https://code.visualstudio.com/docs/editor/codebasics>

<https://code.visualstudio.com/docs/getstarted/introvideos>

<https://github.com/johnpapa/pluralsight-vscode-samples>

<https://www.youtube.com/watch?v=hilznKQij7A>

<https://code.visualstudio.com/api/working-with-extensions/publishing-extension>