



VISUAL STUDIO CODE NINJA GUIDE

ATSUSHI MORIMOTO @74TH 

Visual Studio Code Ninja Guide

01

Atsushi Morimoto (@74th) 著

目次

はじめに『秘伝の VSCode 操作術』	6
1. エクスプローラービューとエディター	7
1.1 ファイルをすばやく開く方法	
1.2 ジャンプした後に戻ってくる	
1.3 対応する括弧にジャンプする	
1.4 選択領域を広げる	
1.5 指定した行番号に移動する	
1.6 次のエラーの行に飛び、すぐにクイックフィックスを適用する	
1.7 エディターを分割表示したり、プリセットのレイアウトを使う	
1.8 ファイルツリーから特定のファイルを非表示にする	
1.9 ファイルコピー時に数値をインクリメントする	
1.10 同じディレクトリのファイルを開く	
2. 検索	19
2.1 検索から特定のファイルを除外する	
2.2 特定のディレクトリ内で検索する	
2.3 置換対象のファイルの選択に、除外するファイルの方を選ぶ方法	
2.4 選択した行のみを対象に一括置換を行う	
2.5 テキストを選択した状態で、クイック検索を開いた時の挙動を変える	
2.6 検索で Smart Case を有効にする	
3. Git	23
3.1 変更を部分的にステージに追加する、またはもとに戻す	
3.2 次の変更行に飛ぶ	
3.3 フォルダー内のファイルを一括してステージに追加する、またはもとに戻す	
3.4 pull の時に未コミットのファイルをマージする	
3.5 コミット後に自動で push、sync を行う	
4. 自動系	26
4.1 保存時にフォーマットする	
4.2 デバッグ実行前にビルドを自動で実行する	
4.3 ワークスペース起動時にバックグラウンドタスクを実行する	
4.4 保存時にビルドやリントを実行する	

4.5 HTML の閉じタグを自動で入力する	
4.6 ファイルパスを補完する	
5. 環境、設定系	31
5.1 異なるマシン間で設定や拡張機能を同期する	
5.2 Setting Sync で設定を同期しつつ、ホストや OS ごとに異なる設定を行う	
5.3 ワークスペースで共通の設定を用意する	
5.4 ワークスペース設定として拡張機能のインストールを促す	
5.5 言語ごとに異なる設定を使う	
5.6 リモート SSH 機能で拡張機能を自動インストールする	
6. 見た目系	36
6.1 使っているカラーテーマを一部分だけ変更する	
6.2 ワークスペースごとに色を変える	
6.3 日本語と英語で異なるフォントを設定する	
6.4 フォントの太さを変える	
6.5 フォントのリガチャーを有効にする	
6.6 テキスト中のカラーコードを視覚的に表示する	
7. キーボードショートカットとスニペット	39
7.1 サイドバー（左タブ）の切り替え	
7.2 左右のタブグループ、サイドバーにフォーカスを移す	
7.3 パネルのタブを切り替える	
7.4 ページ半分の量をスクロールする	
7.5 カーソルの位置のテストを実行する	
7.6 よく入力する項目をスニペットにする	
8. その他	45
8.1 TODO コメントを管理する	
8.2 スペルミスをチェックする	
8.3 CSV やログファイルを見やすくする	
8.4 JSON や YAML で補完を有効にする	
8.5 デバッグ実行をしたい時の設定のヒント	
9. 筆者が活用している設定	50
9.1 見た目、カーソル周りの設定	
9.2 エディター内の表示の設定	
9.3 VSCode の挙動に関する設定	
9.4 git の設定	

9.5 自動処理の設定	
9.6 プログラミング言語固有の設定	
9.7 拡張機能固有の設定	
9.8 ワークスペース設定としてよく使う設定	
9.9 操作で変更される設定	
著者 VSCode 関連著作紹介	59
商業誌: Visual Studio Code 実践ガイド	
商業誌: Visual Studio Code デバッグ技術	
同人誌: Visual Studio Code Remote Dev & Cloud Code Guide	
筆者について	63

はじめに『秘伝の VSCode 操作術』

今や、拡張機能や LSP によってソフトウェア開発のほとんどが VSCode 上で可能となりました。筆者は、ほとんど^{*1}の業務を VSCode を使って開発しています。

VSCode の機能はマニュアルなしに直感的に使えるように工夫されていますが、VSCode の開発が進んだ現在では意外と知られていない機能も多いのではないでしょうか。また、VSCode を自分好みにカスタマイズして、手に馴染むように強化している人も多いと思います。本書では、VSCode の入門書では扱われていないような、細かすぎるテクニックを解説します。これらによって、読者の VSCode の操作が 1 秒でも速くなることを目指しています。

また、筆者は他のユーザがどのように VSCode をカスタマイズし、日常の開発をブーストされているのかに非常に興味があります。そのために、まず筆者自身のテクニックを公開してみようというのが本書執筆の動機です。本書で紹介していない VSCode のテクニックのブログ記事やスライド、同人誌等があればぜひお知らせください。

日本での VSCode コミュニティーを活性化するべく 2019 年末から、VSCode Meetup が開催されています。VSCode の知見を得たい、もしくは共有したい方はぜひ参加してください。これまでのイベントではリモートでも参加できるようストリーミング配信を行っており、イベント後にスライドや動画がアップロードされています。イベントページの URL は <https://vscode.connpass.com/> です。

本書は VSCode の 2020 年 2 月時点のバージョン 1.42.0 でのテクニックとなっています。

本書では、OS ごとに異なるキーボードショートカットを、macOS  , Windows  , Linux  のアイコンを使って示します。

Visual Studio Code は Microsoft 社の製品です。また、本書で紹介する拡張機能のライセンスについては、マーケットプレースの各拡張機能のページを参照下さい。本書は Microsoft 社、および拡張機能の開発者とは無関係のファンブックになります。本書の内容について Microsoft 社、および各拡張機能の開発者に問い合わせることはご遠慮いただき、筆者に問い合わせ下さい。

^{*1} Python、Go、Kubernetes など。Android 開発だけは Android Studio を使っています。

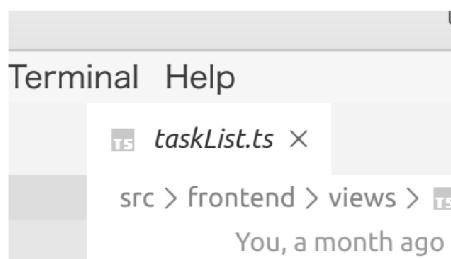
1. エクスプローラービューとエディター

1.1 ファイルをすばやく開く方法

VSCode では目的のファイルを開く方法として、複数の方法が用意されています。状況に応じてすばやくファイルを開くことができると開発に集中できます。

方法 1：エクスプローラービューから

エクスプローラービューのファイルツリーでファイルをクリックすると、そのファイルがエディターで開かれます。ファイルをクリックすると、そのエディターのタブのファイル名は斜体で表示されます。これは、一時的に開いているファイルで、別のファイルを開くとこのファイルは閉じられます。

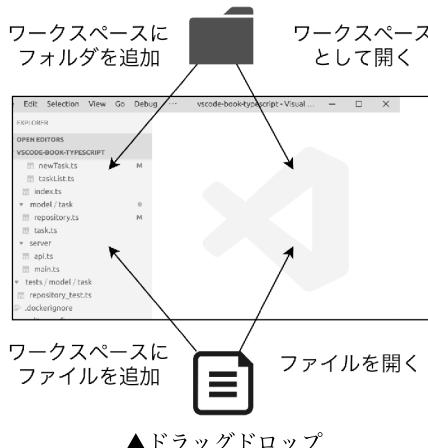


▲一時的に開いているファイルのタブ（ファイル名が斜体で表示）

一時的に開かれているファイルを閉じないようにするには、 :Cmd+S、 / :Ctrl+S で保存するか、ファイルツリーのファイル名をダブルクリック、もしくはエディターのタブのファイル名をダブルクリックします。

1. エクスプローラービューとエディター

方法 2：ファイルからドラッグドロップ



macOS の Finder、Windows のエクスプローラー、Linux の Gnome3 Files から VSCode にファイルをドラッグドロップすることができます。そして、ドロップ先によって機能が異なります。

エディターの中にファイルをドロップしたときには、そのファイルをエディターで開くことができます。また、エディターの中にフォルダーをドロップした場合は、そのフォルダーをワークスペースとして開くことができます。この時、現在開いているワークスペースは一度閉じられることに注意してください。

一方、ファイルやフォルダーをエクスプローラービューにドロップした時には、そのファイルやフォルダーがファイルツリーに追加されます。

また、ファイルやフォルダーをパネルのターミナルタブにドロップすると、ターミナルにフルパスが記入されます。

方法 3 : Go To File のあいまい検索を使う (Quick Open)

ワークスペースからあいまい検索でファイルを検索することができます。

3. Git

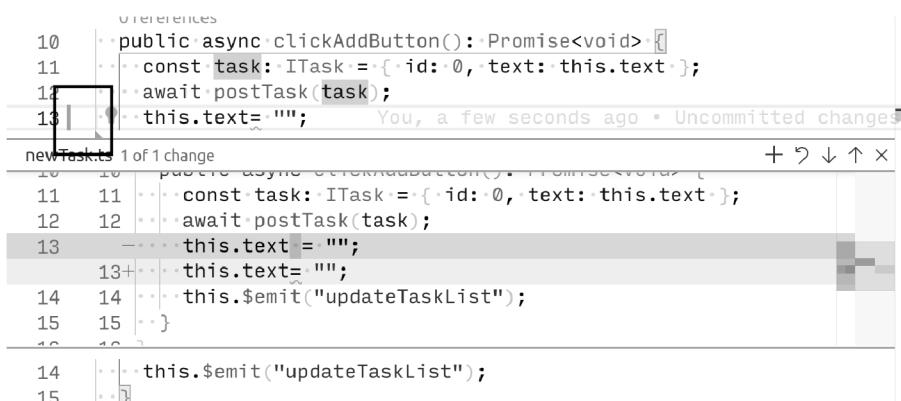
3.1 変更を部分的にステージに追加する、またはもとに戻す

Git では行単位で変更をステージングし、コミットすることができます。VSCode では、ソースコントロールビューでファイルの差分を表示している時に部分的に追加したい行を選択し、右クリックメニューで「Stage Selected Ranges」を選ぶと部分的にステージに追加することができます。また、変更の取消もここから行うことができます。



▲選択した行の右クリックメニュー

また、ステージングされていない変更がある場合、エディターの行番号とテキストの間にマークが付きます。これをクリックすると、変更をエディター内で確認することができます。このウインドウの右上のボタンで、この範囲をステージに追加する、またはもとに戻すことができます。

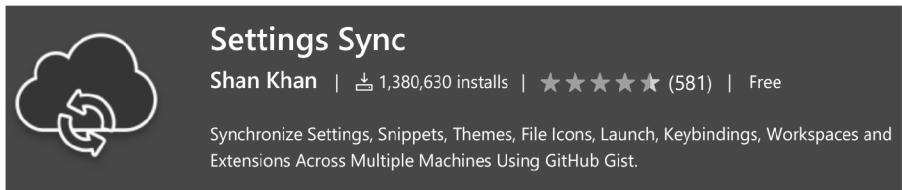


▲マークの位置と、エディター内で開いた変更

5. 環境、設定系

5.1 異なるマシン間で設定や拡張機能を同期する

複数のマシンで設定やインストールした拡張機能を同期させる拡張機能に「Setting Sync *1」があります。



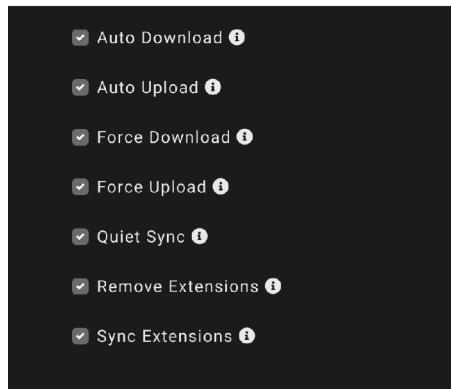
▲Setting Sync

Setting Sync は、gist に設定ファイルをコミットし、その設定ファイルを各マシンで同期させるものです。

gist の設定を済ませた後は、コマンド「Sync: Advanced Options」から、「Sync: Toggle Auto-Upload On Setting Change」と「Sync: Toggle Auto-Download On Startup」を実行すると、自動アップロード、ダウンロードを有効にできます。

また、設定だけではなく拡張機能も同期することができます。コマンド「Sync: Advanced Options」から「Sync: Open Settings」を選択して設定画面を開き、「Sync Extensions」をクリックして有効化します。

*1 <https://marketplace.visualstudio.com/items?itemName=Shan.code-settings-sync>



▲Settings Syncの設定

5.2 Setting Sync で設定を同期しつつ、ホストや OS ごとに異なる設定を行う

Setting Sync にはホスト名、OS ごとに設定を使い分ける機能があります。以下のように`// @sync os=linux`というコメントを付けると、その下の行の設定は特定の OS の場合のみ有効になります。

```
// settings.json
{
  // @sync os=linux
  "python.pythonPath": "/home/linuxbrew/.linuxbrew/bin/python3",
  // @sync os=mac
  // "python.pythonPath": "/usr/local/bin/python3",
  // @sync os=windows
  // "python.pythonPath":
  "C:\\\\Users\\\\nnyn\\\\AppData\\\\Local\\\\Programs\\\\Python37\\\\python.exe",
}
```

その環境で有効でない設定はコメントアウトしておきます。OS によってコメントアウトが切り替えられ、それぞれの環境で有効と無効が切り替わります。

また、同一 OS であるが 2 台以上の PC で異なる設定をしたい場合、ホスト名を設定して設定を切り替えることができます。

コマンド「Sync: Advanced Options」から「Sync: Edit Extension Local Settings」を実行すると、ローカルの Setting Sync の設定ファイルを開くことができます。この設定に「hostName」という設定があり、これが環境名として使われます。

Visual Studio Code Ninja Guide

2020年3月1日 初版発行 (@技術書典8)

筆者 Atsushi Morimoto (@74th)

イラスト もりもり
