



VS Code 完全ガイド

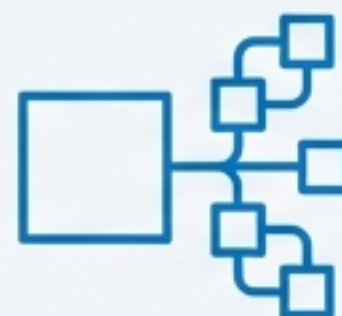
快適なコードエディタ生活をはじめのための全知識

なぜVS Codeなのか？



パワフル & 高速 (Powerful & Fast)

プログラミングを快適にするための重量級の機能を備えた、軽量なエディタ。



豊富な拡張機能 (Rich Ecosystem)

あらゆる言語やワークフローに合わせて、自分だけの環境を構築可能。



Microsoftによる開発 (Developed by Microsoft)

活発な開発と強力なコミュニティによるサポート。



無料 & オープンソース (Free & Open Source)

プロフェッショナルな標準ツールが、誰でも無料で利用可能。

5分で完了：インストールと日本語化

1



公式サイトからダウンロード

URL: `code.visualstudio.com`

お使いのOS (Windows, Mac, Linux) に合ったバージョンを選択。

2



インストーラーを実行

重要な設定：「エクスプローラーのコンテキストメニューに[Codeで開く]アクションを追加する」にチェックを入れると便利です。

3



日本語化パックを導入

拡張機能 (Extensions) サイドバーで「`Japanese Language Pack for Visual Studio Code`」を検索してインストール。

基本画面構成：VS Codeの全体像を把握する

4. ツールバー & コマンドパレット (Toolbar & Command Palette)

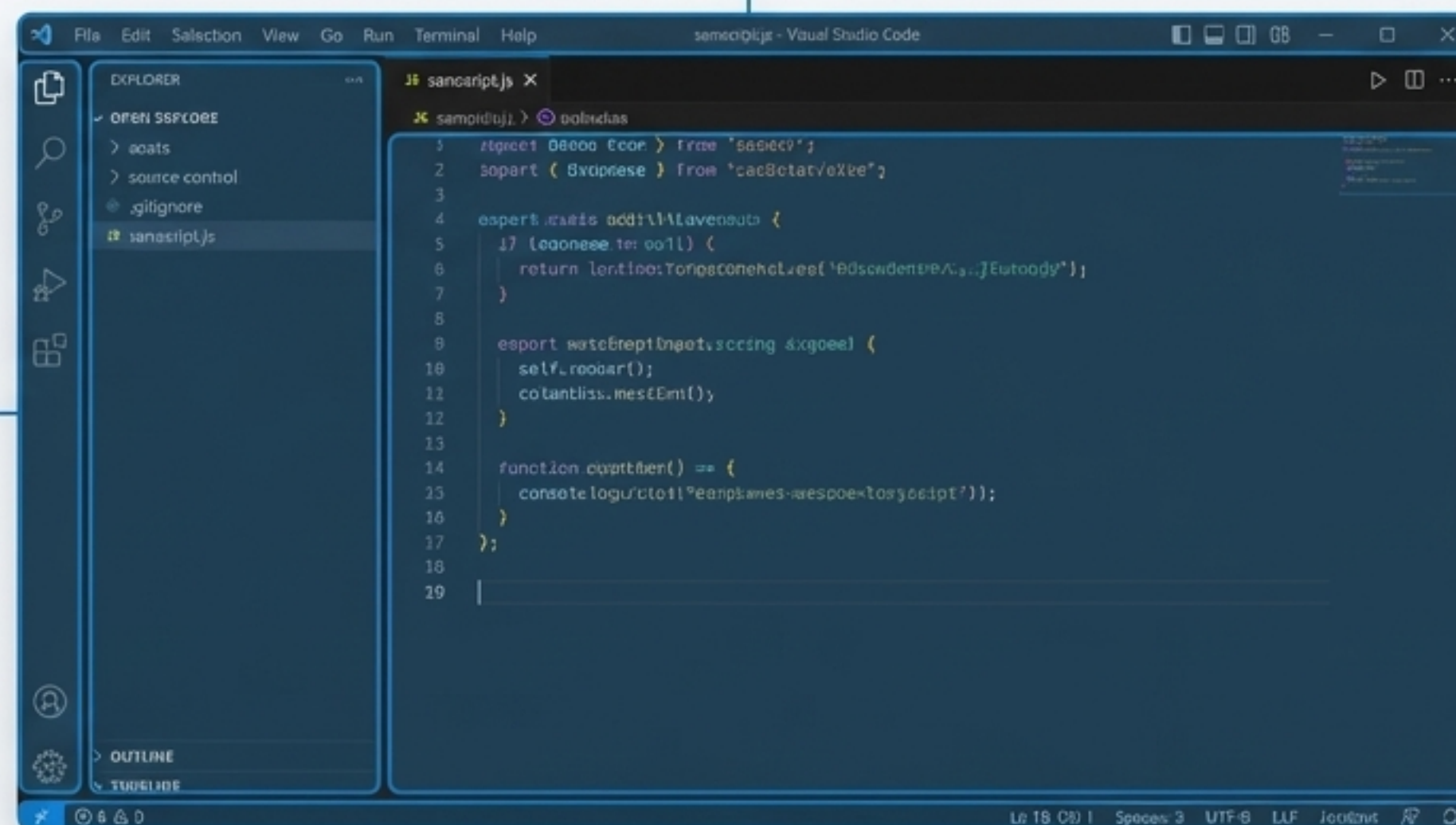
全ての機能と設定へのアクセスポイント。

1. サイドバー (Sidebar)

ファイル管理、検索、ソース管理(Git)、拡張機能の拠点。

2. エディタ (Editor)

実際にコードを記述し、編集するメインエリア。複数のタブでファイルを切り替え可能。



3. ステータスバー (Status Bar)

ファイル情報、Gitブランチ、エラー通知などが表示される。

コマンドパレット：VS Codeの頭脳を使いこなす

これは何か？ (What it is)

VS Codeのあらゆる機能、コマンド、設定にアクセスできる統一インターフェース。

起動方法 (How to open)

Windows: `Ctrl+Shift+P`

Mac: `Cmd+Shift+P`

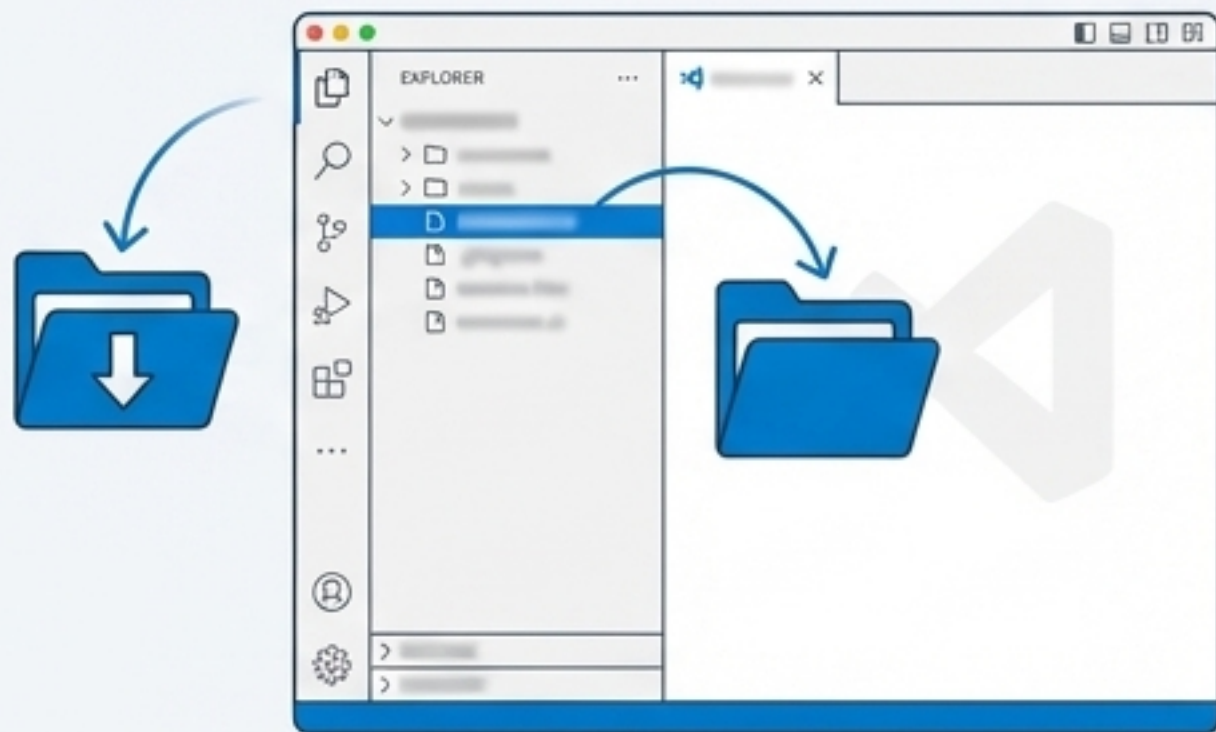
なぜ重要か？ (Why it's essential)

マウス操作から解放され、思考のスピードで作業できるようになります。設定変更や機能検索で迷ったら、まずここを開きましょう。



基本操作：ファイルと内蔵ターミナル

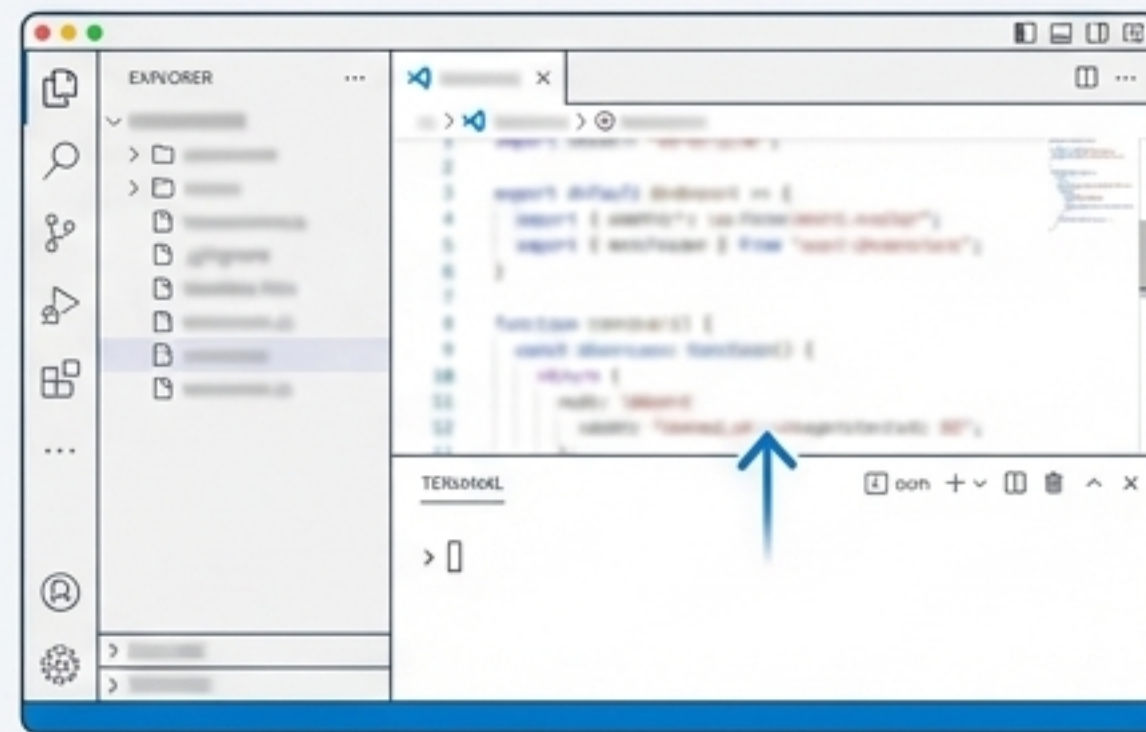
ファイル・フォルダの操作



開く (Open): ファイルやフォルダをエディタにドラッグ&ドロップするのが最も直感的です。

作成・変更・削除 (Create, Rename, Delete):
サイドバーのエクスプローラーでファイルを右クリックするだけで簡単に行えます。

内蔵ターミナル



エディタから離れずにコマンドを実行できます。

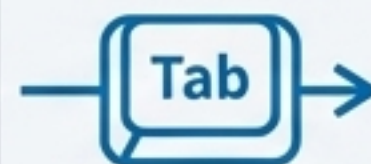
表示/非表示 (Toggle):
Windows: **Ctrl+@** , Mac: **Cmd+J**

編集テクニック：効率を劇的に上げる3つの機能

1. Emmet

HTML/CSSを高速生成。例えば
`div.container>ul>li\*5` + `Tab` キーでリスト構造を一瞬で作成できます。

```
<div.container>ul>li*5
```



```
<div class="container">
  <ul>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
    <li></li>
  </ul>
</div>
```

2. マルチカーソル (Multi-Cursor)

`Alt` キー (Mac: `Option`) を押しながらクリックすることで、複数箇所を同時に編集できます。

```
<p class="item">Text 1</p>
<p class="item">Text 2</p>
<p class="item">Text 3</p>
```

3. 定義へ移動 (Go to Definition)

`F12` キーを押すだけで、関数や変数の定義元へ一瞬でジャンプします。

A 使用箇所 (Usage)

```
const userData = getUserData();
```

F12

B 定義元 (Definition)

```
function getUserData() {
  ...
}
```


プロジェクト管理術：ワークスペース機能

ワークスペースとは？ (What is a Workspace?)

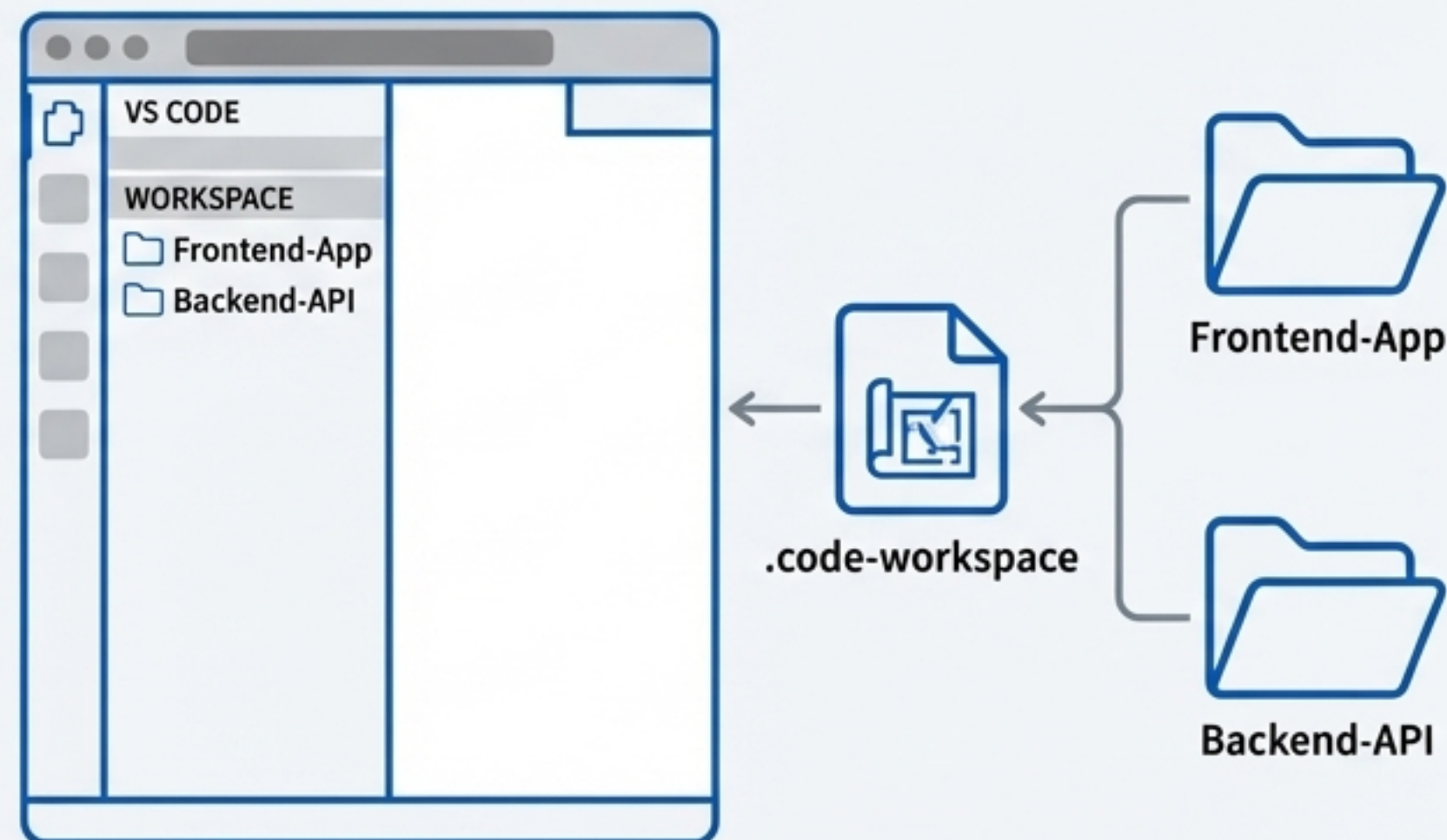
複数のプロジェクトフォルダを1つのウィンドウで管理するための仕組みです。フロントエンドとバックエンドを同時に開発する場合などに最適です。

主な利点 (Key Benefit)

プロジェクトごとに設定（推奨拡張機能、整形ルール等）を分離・共有できます。これにより、チーム開発での環境統一が容易になります。

作成方法 (How to Create)

「ファイル」メニュー → 「ワークスペースにフォルダーを追加...」を選択します。



厳選ショートカット：これだけは覚えたい5選

機能 (Function)	Windows	Mac
コマンドパレット (Command Palette)	Ctrl+Shift+P	Cmd+Shift+P
ファイル検索	Ctrl+P	Cmd+P
コード整形 (Format Document)	Shift+Alt+F	Shift+Option+F
行をコメントアウト (Comment Line)	Ctrl+/ 	Cmd+/
サイドバー表示/非表示 (Toggle Sidebar)	Ctrl+B	Cmd+B

拡張機能：VS Codeを自分だけの最強ツールに

おすすめ拡張機能スターターパック

バージョン管理 (Version Control)



GitLens, Git Graph

Gitの変更履歴をエディタ内で視覚的に確認できます。

Web開発 (Web Development)



Live Server

ファイル保存時にブラウザを自動でリロードします。

コード品質 (Code Quality)



Error Lens

エラーや警告を該当行に直接表示してくれます。



Path Intellisense

ファイルパスの入力を自動補完します。



Code Spell Checker

コード内のスペルミスを検知します。



Prettier - Code formatter

コードを統一されたスタイルに自動整形します。

カスタマイズ①：見た目を自分好みに

配色テーマ (Color Themes)

コマンドパレットで`配色テーマ`と入力して選択。長時間作業しても目が疲れにくいダーク系テーマが人気です。



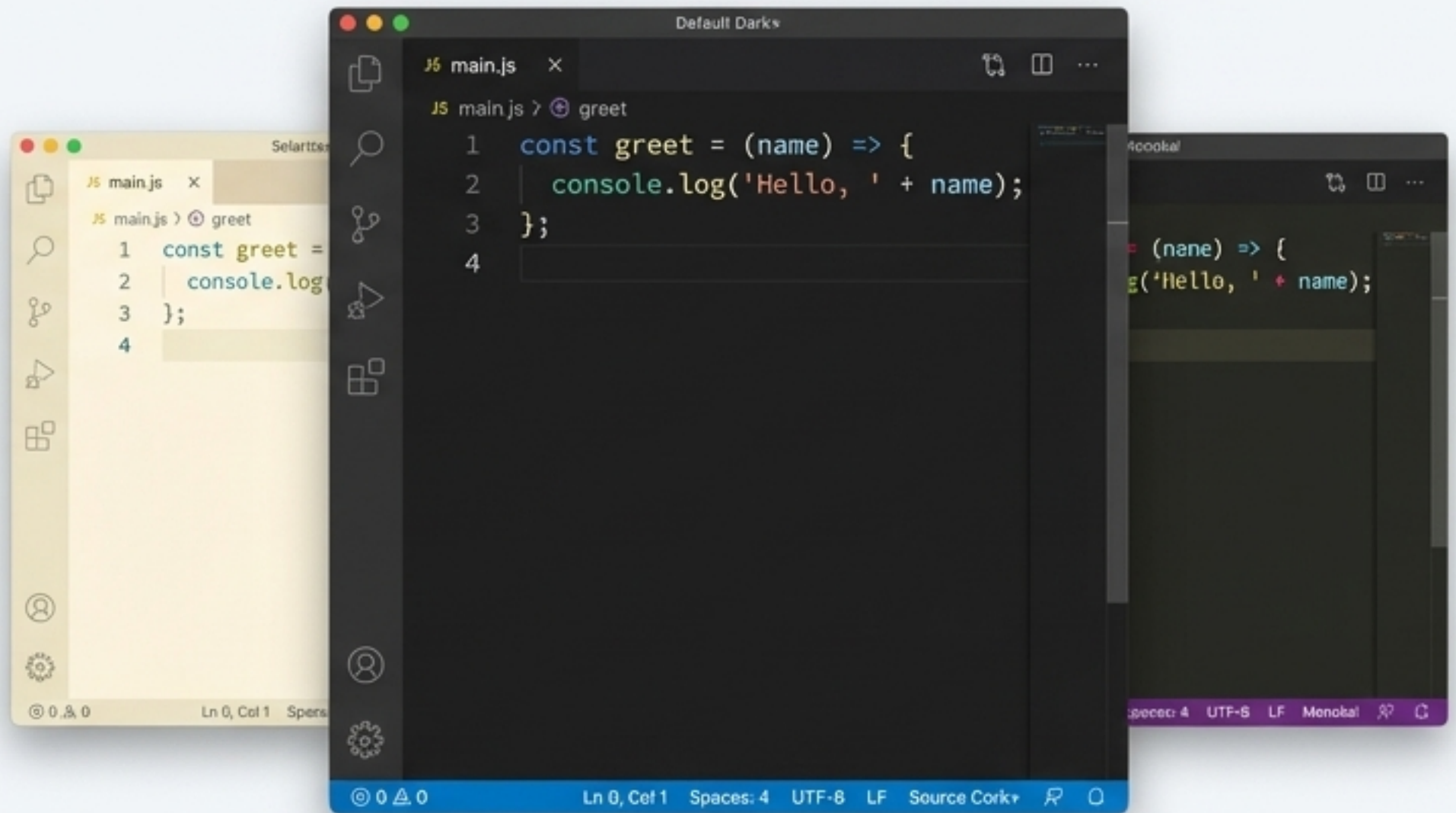
アイコンテーマ (Icon Themes)

ファイルの種類をアイコンで視覚的に識別し、プロジェクトの視認性を向上させます。



フォント & サイズ (Font & Size)

設定画面の`Font Size`で調整。読みやすいフォントとサイズは生産性に直結します。



カスタマイズ②：`settings.json`による高度な設定

これは何か？ (What it is)

VS Codeの全設定を管理するJSONファイル。UIでは設定できない詳細な項目もカスタマイズ可能です。

開き方 (How to open)

コマンドパレットで`ユーザー設定を開く (JSON)` (Open User Settings (JSON)) を検索します。

設定例：保存時に自動フォーマット

この設定を追加すると、ファイルを保存するたびにコードが自動で綺麗に整形されます。

```
{"editor.formatOnSave": true}
```

```
function greet(name) {  
  console.log('Hello, ' + name);  
}
```



保存時

```
function greet(name) {  
  console.log('Hello, ' + name);  
}
```


設定例：生産性を極めるための`settings.json`

```
{  
  // --- 見た目の調整 (UI Tweaks) ---  
  "workbench.colorTheme": "Default Dark+",  
  "editor.fontSize": 16,  
  "window.zoomLevel": 0,  
  
  // --- エディタの挙動 (Editor Behavior) ---  
  "files.autoSave": "afterDelay",           // 少し待つと自動保存  
  "files.insertFinalNewline": true,         // ファイル末尾に改行を自動挿入  
  "files.trimTrailingWhitespace": true,     // 行末の不要な空白を自動削除  
  
  // --- 言語ごとの設定 (Language-specific settings) ---  
  // Pythonファイルを開いている時だけ適用される  
  "[python]": {  
    "editor.defaultFormatter": "ms-python.black-formatter"  
  }  
}
```

****Note****: この設定はあくまで一例です。ここから自分好みに育てていくことが、エディタを使いこなす醍醐味です。

あなたのだけの最強の開発環境が完成

まとめ (Summary): これで、VS Codeを効率的に使いこなすための基礎が固まりました。



次のステップ (Next Steps):

- **言語特化の拡張機能 (Language-Specific Extensions):** お使いのプログラミング言語 (Python, JavaScript等) 専用の拡張機能を探してみましょう。
- **デバッグ機能の活用 (Utilize Debugging):** VS Codeの強力なデバッグ機能に挑戦してみましょう。
- **ショートカットの習慣化 (Habituate Shortcuts):** 今日学んだショートカットを少しずつ身体に覚えさせ、作業をさらに高速化しましょう。

Happy Coding!

