

# Java ランタイムの Visual Studio Code

`java.configuration.runtimes` を Visual Studio Code (VS Code) の設定で使用するには、`settings.json` ファイルで設定する必要があります。この設定を使用すると、VS Code が Java プロジェクトで使用できる複数の Java Development Kit (JDK) バージョンを指定できます。特に、特定の Java バージョンを必要とする異なるプロジェクトで作業する際に便利です。以下はその設定方法のステップバイステップガイドです。

## `java.configuration.runtimes` の設定方法

### 1. VS Code の設定を開く:

- Windows/Linux: ファイル > 設定 > 設定に移動するか、Ctrl + , を押します。
- macOS: コード > 設定 > 設定に移動するか、Cmd + , を押します。

### 2. JSON 設定ファイルにアクセス:

- 設定 UI で `java.configuration.runtimes` を検索します。
- 「Java: Configuration: Runtimes」のようなオプションが表示されます。設定説明の下にある「`settings.json` で編集」というリンク（通常）をクリックして `settings.json` ファイルを開きます。

### 3. `settings.json` を編集:

- `settings.json` ファイルで `java.configuration.runtimes` 配列を追加または変更します。この配列には、VS Code が認識する各 JDK バージョンを表すオブジェクトが含まれます。
- 各オブジェクトには通常以下が含まれます:
  - `name`: Java バージョンの識別子（例: JavaSE-1.8、JavaSE-11、JavaSE-17）。
  - `path`: システム上の JDK インストールディレクトリへの絶対パス。
  - `default` (オプション): この JDK をビルドツール (Maven や Gradle) がないフォルダ (プロジェクト) のデフォルト JDK に設定するには `true` に設定します。

以下は設定例です:

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "C:/Program Files/Java/jdk1.8.0_351",
      "default": true
    },
    {
      "name": "JavaSE-11",
      "path": "C:/Program Files/Java/jdk-11.0.15"
    }
  ]
}
```

```

    {
        "name": "JavaSE-17",
        "path": "C:/Program Files/Java/jdk-17.0.6"
    }
]
}

```

#### 4. JDK パスの確認:

- path が JDK インストールのルートディレクトリを指していることを確認します（例: bin フォルダに java.exe または java が含まれている場所）。
- Windows では、パスに前方スラッシュ (/) またはエスケープバックスラッシュ (\) を使用します。
- macOS/Linux では、適切なファイルシステムパスを使用します（例: /usr/lib/jvm/java-17-openjdk）。

#### 5. 保存および再読み込み:

- settings.json ファイルを保存します。
- VS Code を再起動するか、ウィンドウを再読み込み (Ctrl + R または Cmd + R) して変更を適用します。

#### 6. 設定の確認:

- コマンドパレット (Ctrl + Shift + P または Cmd + Shift + P) を開き、Java: Configure Java Runtime コマンドを実行します。
- この操作により、プロジェクト用に利用可能な JDK が表示されるビューが開きます。設定したランタイムが「プロジェクト JDKs」タブに表示されていることを確認します。

## 仕組み

- **管理されていないフォルダ:** ビルドツールがないプロジェクト（例: 単なる Java ファイル）では、VS Code は java.configuration.runtimes で指定された default JDK を使用します。
- **管理プロジェクト (Maven/Gradle):** ビルドツールがあるプロジェクトでは、JDK バージョンはビルド設定（例: pom.xml または build.gradle）によって決定されますが、VS Code は互換性のためにここにリストされているランタイムを認識します。
- **言語サーバー:** Java Language Server (Java 拡張機能の一部) は実行に JDK 17+ が必要です。必要に応じて java.jdt.ls.java.home で別途指定できます（例: "java.jdt.ls.java.home": "C:/Program Files/Java/jdk-17.0.6"）。

## 複数の JDK を使用した例

Mac で Homebrew を使用して JDK 8 と 17 をインストールしている場合、設定は以下のようになります:

```

{
    "java.configuration.runtimes": [

```

```
        {
          "name": "JavaSE-1.8",
          "path": "/Library/Java/JavaVirtualMachines/jdk1.8.0_351.jdk/Contents/Home",
          "default": true
        },
        {
          "name": "JavaSE-17",
          "path": "/Library/Java/JavaVirtualMachines/jdk-17.0.6.jdk/Contents/Home"
        }
      ]
    }
```

## ヒント

- 正しい name フォーマット (JavaSE-<version>) を使用し、Java バージョンに一致させます（例: Java 11 の場合は JavaSE-11）。
- 1つの JDK しかない場合でも、リストに追加して VS Code が明示的にそれを使用するようにします。
- JDK パスを確認するには、ターミナルで `java -version` を実行し、インストールディレクトリを確認します（例: Windows では `where java`、Unix 系システムでは `which java` を実行し、JDK ルートに戻ります）。

この設定により、VS Code で複数の Java バージョンを柔軟に使用できます！JDK パスの検索やトラブルシューティングでお手伝いが必要な場合はお知らせください。