

コード自動生成・修正 (Automatic_code_generation) プログラムの使用方法

※このプログラムは VSCode と GitHub Copilot を使用します。学生は申請後、無料で使用可能です。

<このプログラムについて>

このプログラムは、エディタである **VSCode** と LLM である **GitHub Copilot** を用いて、コード生成から、バグ修正までを自動で行うプログラムです。バッチファイルを起動させ、自然言語で要求（プロンプト）を入力すると、コードを生成開始します。**対応言語は Python** です。

このプログラムでは、生成したコード内に使用できないモジュールがあった時、そのパッケージを自動でインストールします。ただし、pip のパッケージ名とモジュール名が違うものは手動でインストールする必要があります。

< 設定方法 >

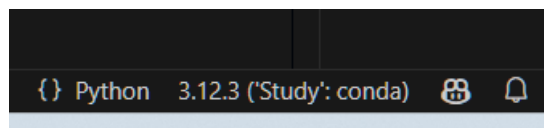
1. 配置

このプログラム (Automatic_code_generation フォルダ) は C ドライブ直下に配置してください。
その他の場所に配置した場合、Settings.py からパス (program_path) を変更してください。
初期設定: "C:¥Automatic_code_generation¥program"

(1. 2. 仮想環境設定)

動作確認済みの環境は、Python3.12.3 です。

Automatic_code_generation ¥program フォルダを VSCode で開いて、右下のインタプリタからバージョンを設定してください。



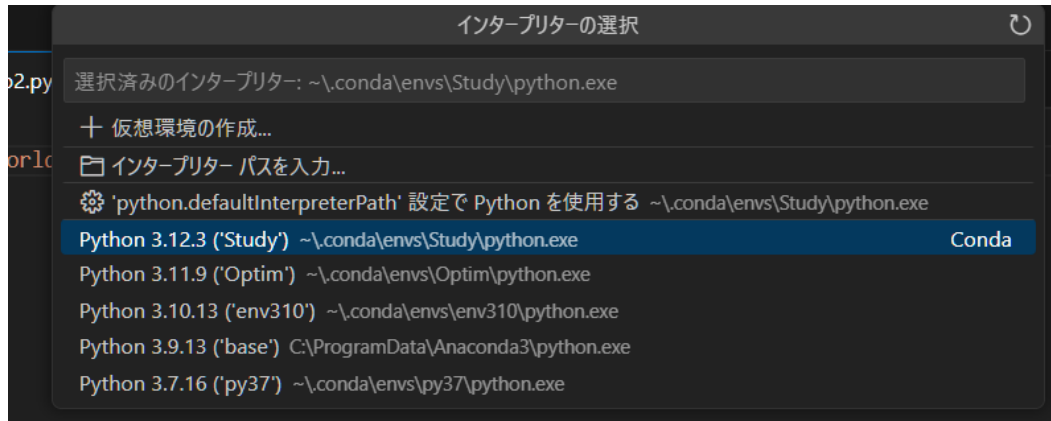


図1 インタプリタ設定

その後、Automatic_code_generation ¥program フォルダを VSCode で開きなおしてください。ターミナル (Ctrl+@) に `python --version` と入力して、右下のインタプリタに表示されているバージョンと一致しているか確認してください。ターミナルとインタプリタの Python バージョンが一致しない場合、不具合が出ることがあります。一致しない場合は、右上の実行ボタンを押して、再度 `python --version` で確認してください。

2. モジュールインストール

`pip install -r requirements.txt` で必要なモジュールをインストールしてください。

3. 使用環境の設定

バッファファイル `main.bat` を開き、図の 8 行目「Study」に当たる部分を、使いたい仮想環境の名称に変更してください。デフォルトでは、conda による仮想環境を使用しています。

```
main.bat
1 @echo off
2
3
4 REM Anacondaの仮想環境を有効化します。
5 call C:\ProgramData\Anaconda3\Scripts\activate.bat
6
7 REM Studyは仮想環境名です。適宜変更してください。
8 call activate Study
9
10 REM Pythonスクリプトを実行します。
11 python "C:\Automatic_code_generation\program\main.py"
12
13 REM 仮想環境を無効化します。
14 call conda deactivate
15
16
17 pause
```

図2 仮想環境名の変更

4. ディスプレイ設定

Windows 画面解像度：1920×1080、画面拡大率：125% 推奨です。値が異なる場合は、PC のディスプレイ設定から解像度と拡大率を変更してください。

このプログラムは、画面解像度：1920×1080、画面拡大率 100%、125% に対して対応可能なことを確認しています。(150% 以上は不具合が確認されているため非推奨。また、画面解像度変更も非推奨。) 拡大率を 125% にしたくない場合や上手くプログラムが動かない場合は、pdf 末尾の**座標設定**に従って、プログラムに使用する座標とプログラムに使用する画像を修正してください。

5. 環境変数設定

PC ウィンドウ下部のタスクバー検索欄から「システム環境変数の編集」を検索し、開きます。「環境変数」を開き、「Path」をクリックした後、「編集」を開き、「新規」をクリックします。「C:¥Automatic_code_generation¥program」を入力して、すべてのウィンドウで「OK」を押します。これでパスが通ったため、Windows + R でバッチファイル実行のショートカットが可能になります。

< 使用方法 >

1. コードを生成したい Python ファイル（以降「コード生成先ファイル」と呼ぶ）を VSCode で開きます。VSCode ウィンドウは全画面で表示してください。また、ターミナル (Ctrl+@) も開いてください。 ※コード生成先の環境は、このプログラムの環境（推奨：Python3.12.3）に合わせる必要はありません。
2. Windows + R でバッチを呼び出し、「main.bat」と入力し、Enter を押します。

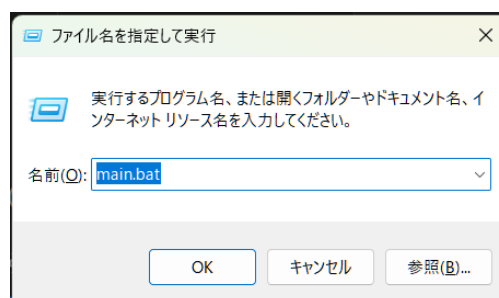


図3 実行ショートカット

3. コマンドプロンプトが開くので、必要ならばc キーを押してプロパティ(数値等)を変更します。通常は変更の必要がないので Enter を押して次に進んでください。プロパティ変更する場合には、Settings.py という設定ファイルが開くので、それを編集し、保存 (Ctrl+S) してください。バッチが再起動します。※バッチを再起動させないと変更が反映されません。

4. コード生成先のフォルダパスを設定します。変更がある場合は c キーを、ない場合は Enter を押してください。VS Code のターミナルに表示されているパス（図 4 の青くなっている部分）をコピー & ペーストしてください。※”PS”や空白、”>”は含めないように注意してください。指定の際、「|」や「|」をつける必要はありません。変更した場合、その設定が保存され、次回にも引き継がれます。

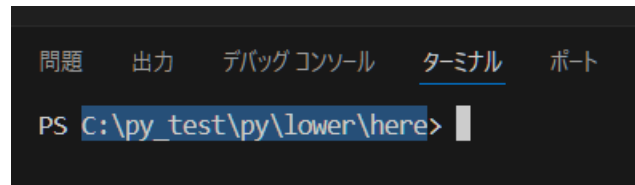


図 4 ディレクトリパスの例

5. 要求（プロンプト）を入力して Enter を押すと、プログラムが開始します。 ※ 図 5 のように、VS Code を全画面で開き、バッチのウィンドウ（コマンドプロンプト）を左側に寄せた状態で開始してください。緑枠で示すように、右端のスペースを空けておいてください。

生成が完了するまでマウスやキーボードを操作しないようにしてください。

プログラムを途中で停止・再起動させる場合は、ESC キーを押してください。

※プログラムを開きながら Word など他のファイルの操作をしていた場合、ESC キーが反応することがあります。その場合、マウスポインタを画面の左上角に移動させると停止します。

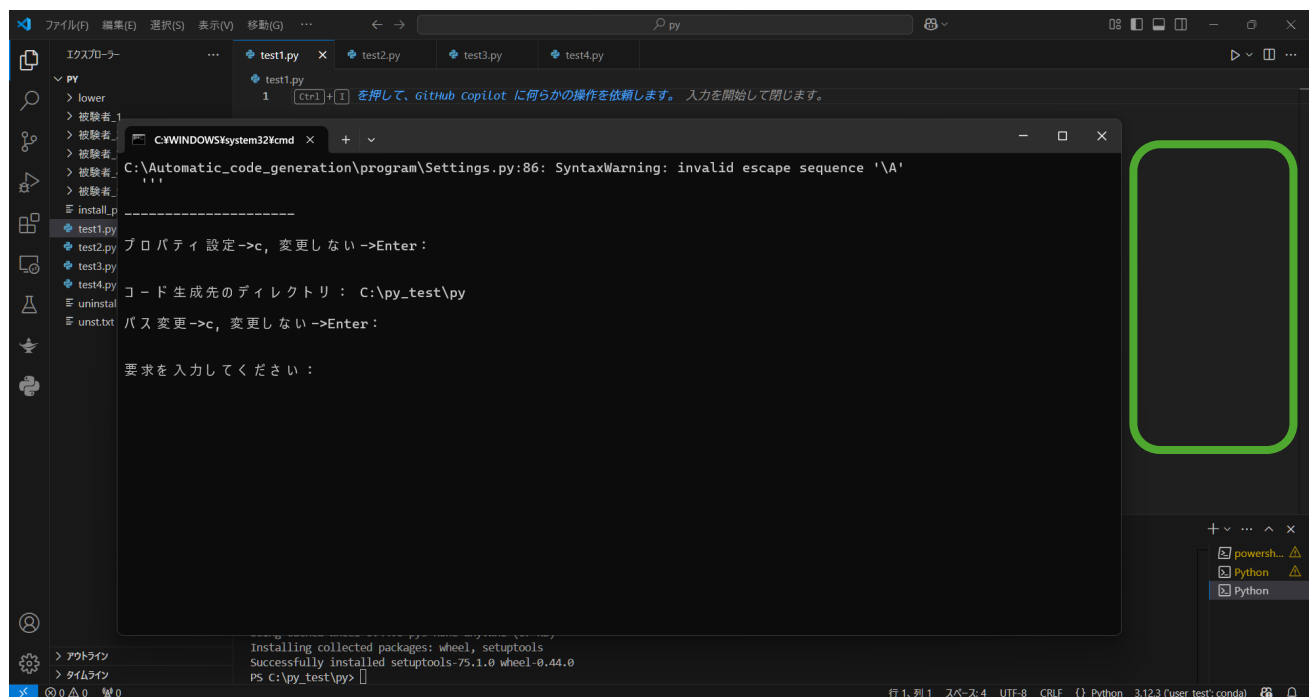


図 5 プログラム開始時の画面

6. プログラムが終了するとウィンドウが開きます。このウィンドウは、Enter / Space キーで閉じることが出来ます。反応しない場合は、ウィンドウをクリックしてから Enter / Space キーを押すか、直接×ボタンを押してください。 ※ このウィンドウを閉じないとプログラムが終了しないので注意してください。



図7. 生成完了時のウィンドウ

(座標設定) -----

Settings.py を開いてください。画面拡大率 100% のときは、①画面拡大率の値を修正し、②座標を以下の値に修正して試してください。(画面拡大率 125%、または、画面拡大率 100%で下記設定後にうまく動作しない場合は、**座標修正**を参照してください。)

- ① screen_rate の値を 100 に変更してください。(初期値：125 (画面拡大率 125%))

```
7  # Windowsの画面拡大率 (%)
8  screen_rate = 125
```

- ② 各画面解像度と拡大率における座標の設定例です。以下のように指定してください。(実際にはより正確な値へ修正する必要があるかもしれません。)

解像度：1920×1080、拡大率：100% のとき

```
x_blanc_space = 1750
y_blanc_space = 400
x_execute = 1830
y_execute = 50
x_terminal = 1600
y_terminal = 1003
```

解像度：1920×1080、拡大率：125% のとき（初期設定）

```
x_blanc_space = 1800
y_blanc_space = 400
x_execute = 1805
y_execute = 64
x_terminal = 1200
y_terminal = 975
```

（座標修正）

上記を試したり、画面拡大率 125%にもかかわらず、プログラムが上手く動かない場合は以下に従って修正してください。

Tkinter.py を開いて、PC の画面拡大率を指定してください。

```
8  # 解像度と拡大率は、PCのディスプレイ設定を確認してください。
9  # 画面解像度1920x1080（変更非推奨）
10 screen_width = 1920
11 screen_height = 1080
12 # 画面拡大率を設定して下さい。
13 scale_factor = 1.25 # （初期設定）拡大率125%の場合
```

図 8．Tkinter の設定

Tkinter.py を実行し、表示された座標に従って、Settings.py の以下 6 箇所を修正してください。（詳細は図 9 を見てください。）

x_blanc_space / y_blanc_space	コードを記述するスペース内の座標です。 バッチファイルを実行した後に、プログラムが VSCode のウィンドウをクリックできるように、右側のスペースを指定してください（図 3）。
x_execute / y_execute	VSCode 右上にある実行ボタンの座標を指定してください。
x_terminal / y_terminal	ターミナルの右下を指定してください。y 座標は、ターミナル内のできるだけ下の部分を指定してください。

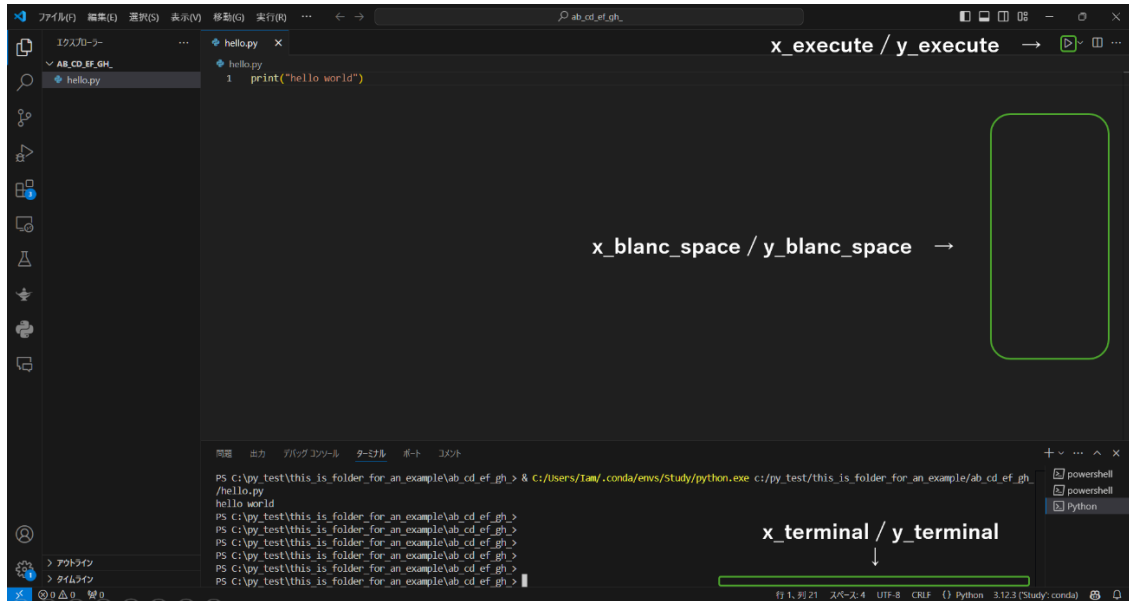


図 9. 各座標指定

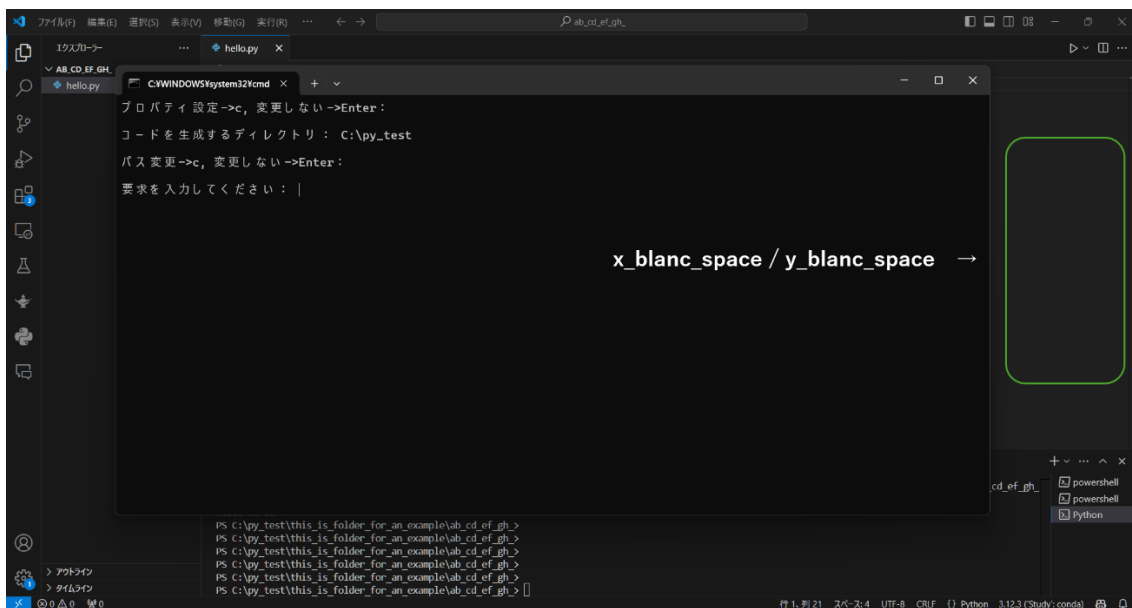


図 1 0. blanc_space の（バッチを開いたときに干渉しない場所の）座標指定

修正後、再度実行し、正常に動作するか確認してください。
