ワークショップ

AIを使ったプログラミング GitHub Copilot編

2025/10/24

株式会社ホープ

AI活用まとめ

- Aさん　対象：アラーム　言語：Rust
  - datetime型で時間を設定しようとするとエラー→解決策を問い合わせのループ
- Bさん　対象：アラーム　言語：Rust
  - 音声ファイルを再生しようとしたが、ライブラリがうまく呼べなかった
  - AIが書いたコードをAIにレビューさせたら、このコードは正しくないと言ってくる。。。
- Cさん　対象：回線速度計測　言語：Haskell
  - 環境構築→ビルドOK
  - AI出力のコードでエラー→AIにエラー解決問い合わせ→環境パスのアスキーコード問題だった（解決）
  - 別のエラー→言語仕様（importの位置）→解決→別のエラー（ここまで）
- Dさん　対象：回線速度計測　言語：Go
  - ハルシネーション→プロンプトを工夫→ハルシネーション…を繰り返し、最終的には、GitHubにあるpythonスクリプトをGo言語の外部コマンド呼び出しで実行するコードを提案された。
  - pythonがインストールされておらず、ここでタイムアップ…

- 所感
  - 環境構築など、ある程度確立されている手順などはかなり正確に出力できると感じた
  - 図らずもヒューマンインザループ（HIL）のプロセスを体験できたことはよかった
  - 知らない言語ということもあり、抽象的なプロンプトになっているような気がする
  - コーディングに特化したAIエンジンだとどうなるか気になるところ

GitHub Copilotの活用

https://www.choge-blog.com/programming/github-copilot/

- GitHub Copilotでできること（抜粋）
  - コメント駆動開発
  - 関数名から実装を生成
  - テストコードの自動生成
  - リファクタリング支援
  - ドキュメント生成
  - エラー処理の追加

GitHub Copilotの活用

- 知っておくべき注意点
  - APIキーやパスワードをハードコーディング
  - 機密情報を含むコメントの記述
  - 社内専用コードでの無断使用（会社のポリシーを確認）

- お約束
  - コードレビューは必須（あくまで提案されたものという認識で！）
  - ライセンスの考慮

GitHub Copilotの活用

- Copilot Chatについて
  - チャット形式でコーディングの質問ができる
    - エラーの解決
    - 実装方法の相談

環境設定

- GitHub アカウントの作成（Sign up）
  - https://github.com/
  - 会社のメールアドレスでOK、パスワードは個別に決めてください

環境設定

- Visual Studio Code
  - インストール

    https://code.visualstudio.com/download

    https://code.visualstudio.com/docs/setup/raspberry-pi

    - 参考）GitHub Copilot Freeについて
      https://docs.github.com/en/copilot/concepts/billing/individual-plans

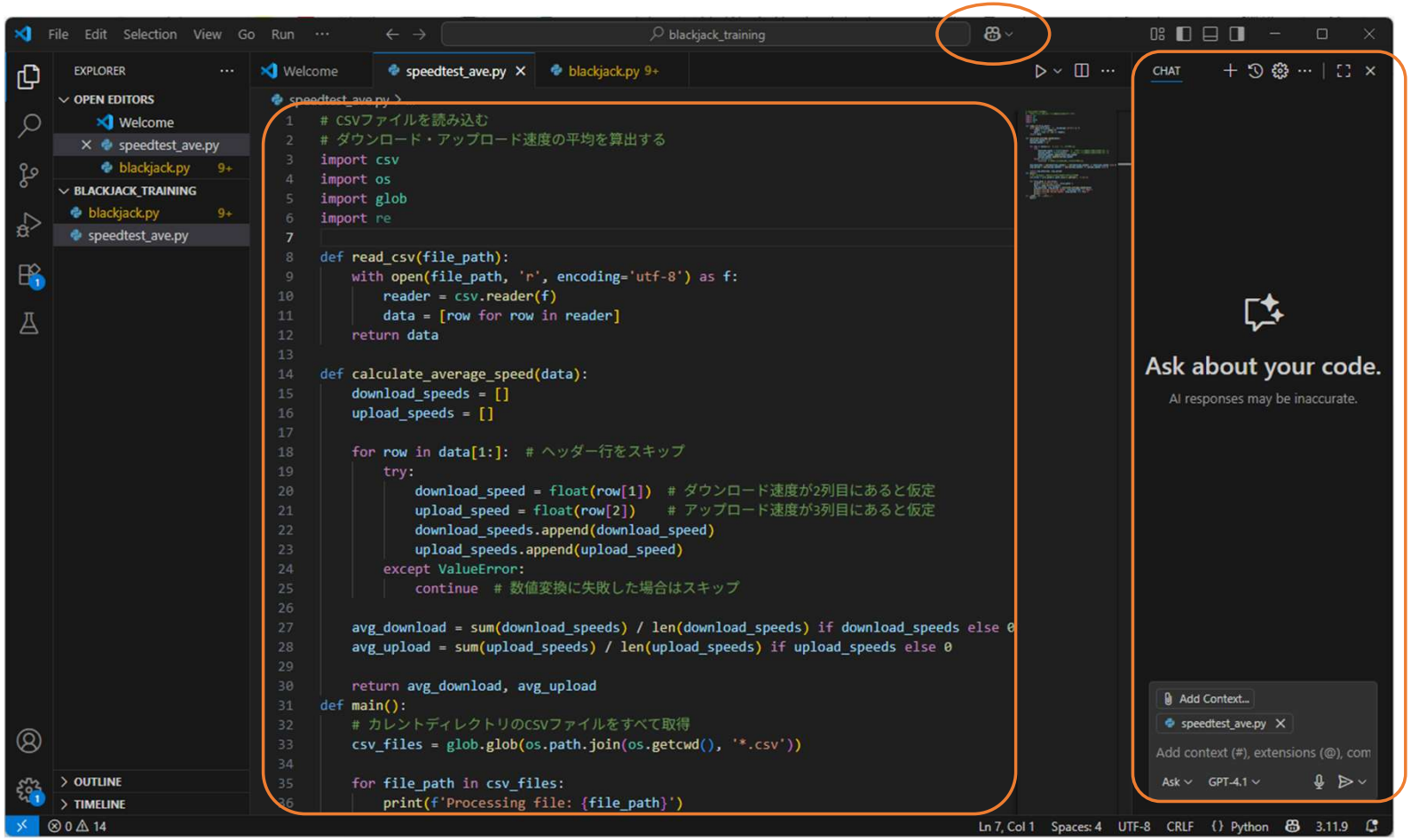
  - 機能を追加（GitHub copilot, GitHub copilot chat)

    https://vscode.dokyumento.jp/docs/copilot/setup

最近はVisual Studio Codeに最初から機能が入っているので、ここは無視できます

環境設定

- イメージ


File Edit Selection View Go Run
blackjack_training
EXPLORER
OPEN EDITORS
Welcome
speedtest_ave.py
blackjack.py 9+
BLACKJACK_TRAINING
blackjack.py 9+
speedtest_ave.py
OUTLINE
TIMELINE
Welcome
speedtest_ave.py
blackjack.py 9+
# CSVファイルを読み込む
# ダウンロード・アップロード速度の平均を算出する
import csv
import os
import glob
import re
def read_csv(file_path):
with open(file_path, 'r', encoding='utf-8') as f:
reader = csv.reader(f)
data = [row for row in reader]
return data
def calculate_average_speed(data):
download_speeds = []
upload_speeds = []
for row in data[1:]: # ヘッダー行をスキップ
try:
download_speed = float(row[1]) # ダウンロード速度が2列目にあると仮定
upload_speed = float(row[2]) # アップロード速度が3列目にあると仮定
download_speeds.append(download_speed)
upload_speeds.append(upload_speed)
except ValueError:
continue # 数値変換に失敗した場合はスキップ
avg_download = sum(download_speeds) / len(download_speeds) if download_speeds else 0
avg_upload = sum(upload_speeds) / len(upload_speeds) if upload_speeds else 0
return avg_download, avg_upload
def main():
# カレントディレクトリのCSVファイルをすべて取得
csv_files = glob.glob(os.path.join(os.getcwd(), '*.csv'))
for file_path in csv_files:
print(f'Processing file: {file_path}')
CHAT
Ask about your code.
AI responses may be inaccurate.
Add Context...
speedtest_ave.py
Add context (#), extensions (@), com
Ask
GPT-4.1
Ln 7, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.11.9

実際にコードを組んでみる

- 課題
  - 会社の回線速度を測定したので、平均値を出してください
- 手順

  Step 1：プロジェクトを準備します　（Shell or Terminal）
  - mkdir speedtest_analysis
  - cd speedtest_analysis
  - code .

  Step 2：プログラムファイルを作成します
  - analyze_speedtest.py を新規作成し、以下のコメントを記述してください。

    # CSVファイルを読み込む
    # ダウンロード・アップロード速度の平均を算出する

  - コメントの下にカーソルを置くと、コードを提案してくれます。

実際にやってみる

- 基本操作
  - Tab　　提案を受け入れる
  - Esc　　提案を拒否
  - Alt +]　　次の提案を表示
  - Alt + [　　前の提案を表示
  - Ctrl + Enter　　提案パネルを開く（複数の候補を確認）

- 使用するSpeedtestログファイル
  - log_speedtest_sample.csv
    - データ列は７列(Down)と８列(Up)です
    - データはbpsですので、Mbpsで表示する場合は125,000で割ってください