山本悠士_yuji96

無題

yuji的には、許容範囲内であればコメントを添えて満点で返却するが、その後の問題で変化が見られない場合は再提出を求めるスタイルでやっていました。
以下のような場合は、GitHubのissueを活用してください。Slackで質問を受けた場合も他の人から同じ質問をされる予想ことを予想できる場合は、issueで質問し直してもらうことをお勧めします。

- 全体に対して説明したい
- シンタックスハイライトを使って説明したい

```
print("こんな感じの書き方")
```

全体

1. コーディング規約が正しいか。
2. for文に書き換え可能なのにwhile文を使用しているコードは非推奨。
3. `for i in range(len())`のような書き方は、ほとんどの場合非推奨。稀に修正しない方が読みやすい場合がある。
4. 不要な型変換をしていないか。例えば、`for i in list(map(...)):`はmap関数の返り値をlist型に変換する必要がない。
5. コードに重複部分が無いか。あった場合はfor文などを用いてまとめたコードを再提出してもらう。
6. 出力の仕方が簡潔かどうか。

```
NG: print(list[0], list[1], list[2], ...)
OK: print(*list)
NG: print("a = " + str(a) + ", b = " + str(b))
OK: print(f"a = {a}, b = {b}")
```

1章

D: 便利な演算子である`//, %`を使っていたらナイス。

2章

D: 以下のような重複したコードを見たら即指摘。

```
W, H, x, y, r = input().split()
W = int(W)
H = int(H)
...
```

3章

A: while文で実装されていたら即指摘。

B: `enumerate`を使っていたらナイス。
D: `range`関数の初期値を上手く設定して実装してほしい。

4章

A; Pythonユーザーに対する初見殺し問題。これに関しては答えに近いアドバイスをしてと良い思う。
B: `pi`がmathライブラリからimportされてなければ即指摘。

6章

この章の問題は複雑な解答が提出されることが多い。①レビュワーがコードを理解するのに時間がかかり、②改善の余地がある場合はヒントのアドバイスをして再提出をしてもらった方が良い。
C: リスト内包表記が使った方が簡潔に実装できることが多い。

7章

`combinations`を使っていたらナイス。（`itertools`は自分でfor文で実装するより処理が遅くなるというデメリットもあるらしいので強要はしなかった。）

11章

難易度が上がるので先にnumpy, pandasの課題をやるのがおすすめ。

numpy

基本的に最後のグラフ出力で赤線と緑線がほぼ重なっていたら問題無い。
今まで、そもそもエラーが発生したり理論的に誤っている箇所を指摘することが多かった。

pandas

問1: 『顧客ID（customer_id）が“CS018205000001”かつ、売上金額（amount）が1,000以上』または『売上数量（quantity）が5以上』という誤った解答をたまに見る。
問2: 『若い順にソート』
問7: formatを指定せずにto_datetimeすると誤った変換がされる。