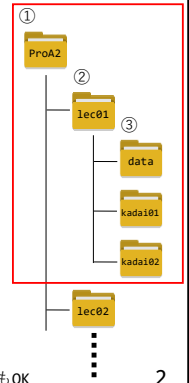


プログラミングA2 第1回

演習課題

準備

1. デスクトップなどの任意の場所にプログラミングA2用のフォルダを作成する
2. 1.の下に第1回授業用のフォルダを作成する
3. 2.の下にデータファイル用のフォルダ、課題01、課題02用のフォルダを作成する



※必ずしも、このようなディレクトリ構造でなくてもOK

2

注意点

- 採点の都合上、指示された要件を満たすコードを実装すること
- 授業時間中に提出された場合は2点満点
- 授業時間外に提出された場合は1点満点
- 授業時間中に1問も提出できなかった場合は、学修支援センターで指導・チェックを受けること
- 授業時間外の質問は、Slackにて伏見が受け付ける(24時間365日営業)

3

課題チェックの流れ

1. 受講生：課題ができたらMoodleにアップロードする
2. 受講生：課題チェック依頼シートにて、待ちキューが少ないTASAの列に学籍番号を入力する
3. TASA：Moodleにアップされた課題をチェックする
4. TASA：チェックが済んだら、
 - Moodleに点数を入力する
 - スプレッドシートの学籍番号を赤字に変更する【チェック完了の合図】
5. 受講生：自分の学籍番号が赤字になったら帰る

4

【課題 01 : greater_average.py】

配布した「base_stats.txt」を読み込み、行ごとの合計値（ポケモンごとの種族値の合計）が、全ポケモンの平均値より高い行だけ標準出力するコードを実装せよ。ちなみに、該当する行数（ポケモン数）は、127 である。

採点の都合上、以下の要件を満たすこととする：

- ファイルからデータを読み込む関数を実装する；
 - － 読み込みファイルのパスを引数とし、読み込んだデータの 2 次元リストを戻り値とする；
 - － with 構文を用いる；
- リストの平均値を計算する関数を実装する；
 - － リストを引数とし、平均値を戻り値とする；
- 読み込みファイル名をコマンドライン引数で指定する；
- 結果は、1 始まりの行番号（ポケモン番号）と種族値リストのタプルを要素としたリストを内包表記で定義する；
- 出力の形式は、pprint 関数を用いて図 1 のようにする；

```
PS C:\Users\admin\Desktop\ProA2> python .\lec01\kadai01\greater_average.py .\lec01\data\base_stats.txt
[(3, [80, 82, 83, 100, 100, 80]),
 (6, [78, 84, 78, 109, 85, 100]),
 (9, [79, 83, 100, 85, 105, 78]),
 (18, [83, 80, 75, 70, 70, 91]),
 (20, [55, 81, 60, 50, 70, 97]),
 (22, [65, 90, 65, 61, 61, 100]),
 (24, [60, 85, 69, 65, 79, 80]),
 (26, [60, 90, 55, 90, 80, 100]),
 (28, [75, 100, 110, 45, 55, 65]),
 (31, [90, 82, 87, 75, 85, 76]),
 (34, [81, 92, 77, 85, 75, 85]),
```

図 1 課題 01 の実行例（一部）：全出力結果は 127 行となる

【課題 02 : count_pairs.py】

配布した「poke_names.txt」を読み込み、3 列目に書かれたタイプのペアをカウントし、出現回数が多いペアから順に出力するコードを実装せよ。当該ファイルは、各行において番号、名前、タイプがタブで区切られており、3 列目のタイプは半角スペースで区切られている。課題 02 では、3 列目のタイプが 2 つある行が対象となる。ちなみに、タイプのペア数は 45 である。

採点の都合上、以下の要件を満たすこととする：

- ファイル読み込みには、with 構文を用いる；
- タイプのタプルをキーとした辞書を用いる；
- ソートにはラムダ式を用いる；
- 読み込みファイル名をコマンドライン引数で指定する；
- 出力の形式は、図 2 のようにする；

```
PS C:\Users\admin\Desktop\ProA2> python .\lec01\kadai02\count_pairs.py .\lec01\data\poke_names.txt
[ (('ひこう', 'ノーマル'), 10),
  (('くさ', 'どく'), 9),
  (('いわ', 'じめん'), 8),
  (('どく', 'むし'), 7),
  (('ひこう', 'むし'), 5),
  (('いわ', 'みず'), 5),
  (('みず', 'エスパー'), 4),
  (('ひこう', 'ほのお'), 3),
  (('ノーマル', 'フェアリー'), 3),
  (('どく', 'ひこう'), 3),
  (('どく', 'みず'), 3),
  (('こおり', 'みず'), 3),
  (('どく', 'ゴースト'), 3),
```

図 2 課題 02 の実行例（一部）：全出力結果は 45 行となる