

FileIO

write

Point

ex: open(path, mode)

- ファイル操作には`open`関数を使う.
- 第1引数pathは保存したいファイルパスを指定する. (例 "hoge.csv")
- 第2引数modeはファイル操作のmodeを指定する. 書き込みは"w".

研究でプログラムを扱う際に計算結果の出力, 計算に必要な数値の読み込みを行うことが多い. ここでは, ファイルを用いた入出力について学んでいく.

次の例は, Pythonでwrite.txtファイルに書き込む方法である.

```
with open("write.txt", "w") as f:  
    f.write("noma is cool")
```

`open`には2つの値を渡す必要がある. 1つ目は保存するファイルパス, 2つ目はモードの指定である. 書き込みは"w", 読み込みは"r"である.

書き込みが成功するとwrite.txtが生成され, 以下の内容が保存されているだろう.

write.txt

```
noma is cool
```

次は2行にかけて書き込みを行う.

```
with open("kaigan_miss.txt", "w") as hoge:  
    hoge.write("noma is cool")  
    hoge.write("mizutani is very cool")
```

しかし, kaigan_miss.txtには以下のように1行で出力されたはずだ.

kaigan_miss.txt

```
noma is coolmizutani is very cool
```

改行は文字列の最後に"\n"をつけると改行される.

```
with open("new_kaigan.txt", "w") as fuga:
    fuga.write("noma is cool\n")
    fuga.write("mizutani is very cool\n")
```

正しく動作すると以下ようになる.

new_kaigan.txt

```
noma is cool
mizutani is very cool
```

writeに渡せる値はstr型のみである. intなどはstrにtype castingする必要がある.

```
with open("num.txt", "w") as file:
    for i in range(0, 5):

        # これはダメ! write はstrだけ!
        # file.write(i)

        # castingする
        nya = str(i) + "\n"
        file.write(nya)
```

type castingも正しく行い, 書き込みを行うと以下のように出力されているだろう.

num.txt

```
0
1
2
3
4
```

csvへ複数列データの書き込み

海岸工学の研究を想定すると複数列あるデータをcsvに書き出せるのは必須である.

```
nums = []
for i in range(0, 3):
    line = []
    for j in range(5, 10):
        line.append(j)

    nums.append(line)
```

```
with open("nums.csv", "w") as f:
    for l in nums:
        str_l = []
        for num in l:
            str_l.append(str(num))

        f.write(",".join(str_l)+"\n")
```

`",".join()`は、`str`型を保持した`list`型の値をカンマ(,)で連結している。`["1", "2", "3"]`の場合、`"1,2,3"`このように変換される。

出力された`nums.csv`を確認してみるとカンマでjoinされていることが分かるだろう。

nums.csv

```
5,6,7,8,9
5,6,7,8,9
5,6,7,8,9
```

read

```
nums = []
with open("nums.csv", "r") as f:
    while True:
        str_line = f.readline().strip().split(",")
        if str_line == [""]:
            break

        num_line = []
        for str_num in str_line:
            num_line.append(float(str_num))

        nums.append(num_line)

for line in nums:
    print(line)
```

出力

```
[5.0, 6.0, 7.0, 8.0, 9.0]
[5.0, 6.0, 7.0, 8.0, 9.0]
[5.0, 6.0, 7.0, 8.0, 9.0]
```

課題

1. `test_score.csv`に5科目10人分の試験の成績とIDがID, 国語, 算数, 理科, 社会, 英語の順で記録されている。この成績が保存されたファイルをコピーし、このデータから10人それぞれの合計点と平均点, さらに科目毎に合計点と平均点を求めてファイルに書き出さない。
 - 10人それぞれの合計点と平均点は, `test_score.csv`の新しい列に追加すること。
 - 科目毎の合計点と平均点は, `test_score.csv`の新しい行に追加すること。

出力

```
73,80,62,84,74,373,74.6
91,89,88,100,89,457,91.4
68,81,60,83,70,362,72.4
56,62,51,61,54,284,56.8
64,72,56,72,58,322,64.4
55,55,47,59,47,263,52.6
61,61,47,65,45,279,55.8
55,52,41,54,45,247,49.4
48,51,46,53,45,243,48.6
86,88,77,95,77,423,84.6
657,691,575,726,604
65.7,69.1,57.5,72.6,60.4
```

ヒント

解答例