情報処理概論

九州大学　工学部地球環境工学科　講義資料　担当：木村

06 ユーザ定義関数／ファイル入出力

1）自分で新しい関数を定義する

(要求)　プログラムの中に同じ処理が何度も出てくる
記述が重複、コードが無駄に大きくなり読みにくい

(解決策)　「ユーザ定義関数」としてまとめ、必要に応じて呼出し

2）プログラムのデータ入出力にファイルを用いる

(要求)　入出力データが膨大
プログラムへのデータ入力をキーボードではなくファイルで行う
計算結果をファイルに保存する

(解決策)　ファイルを開いてデータ読込／ファイルを開いてデータ書込み

関数を定義する

関数定義のフォーマット

```
def 関数名(引数1, 引数2):
        処理1
        処理2
        return 返り値  #返り値がある場合
```

コロン: を
忘れずに付ける

(例) 絶対値を求める関数myabs

```
def myabs(x):
        if x < 0:
                x *= -1
        return x

print(myabs(-5))
```

実行結果
5

1. 引数はf(x)のx. 型は任意. ただし, 要らない場合もある.
2. 関数名(引数 = デフォルト): でデフォルト値を指定できる.
3. 返り値は関数の処理結果を返す. 要らない時もある.
4. **関数を使う前にその関数が定義されていなければならない.**

例1　引数なし

```
#----------ユーザ定義関数ここから
def plus():
    x=input("xを入力")
    y=input("yを入力")
    x=int(x)
    y=int(y)
    print("x + y = %d" %(x+y))
#----------ユーザ定義関数ここまで
#----------ここからプログラム本体
#          （処理はここからスタート）
plus()
```

def plus()の中身が実行される

例2　引数あり　計算結果を返す

```
#--------------------ユーザ定義関数ここから
def multi(x,y):
    z=x*y
    return z
#--------------------ユーザ定義関数ここまで
#--------------------ここからプログラム本体
x=input("xを入力")
y=input("yを入力")
x=int(x)
y=int(y)
print(multi(x,y))
```

実行結果
xを入力3
yを入力2
6

例3　引数にデフォルト値を設定する場合

```
def hello(arg1="good",arg2="Morning"):      #デフォルト値
    print(arg1+arg2)
hello()                          #引数を与えなかった場合はデフォルト値が採用される
hello(“bad”)                  #引数を変更（何番目の引数か指定なし→1番目の引数が該当）
hello(arg2 = "Evening")       # 2 番目の引数を変更
hello(1,3)
```

実行結果

```
goodMorning
badMorning
goodEvening
4
```

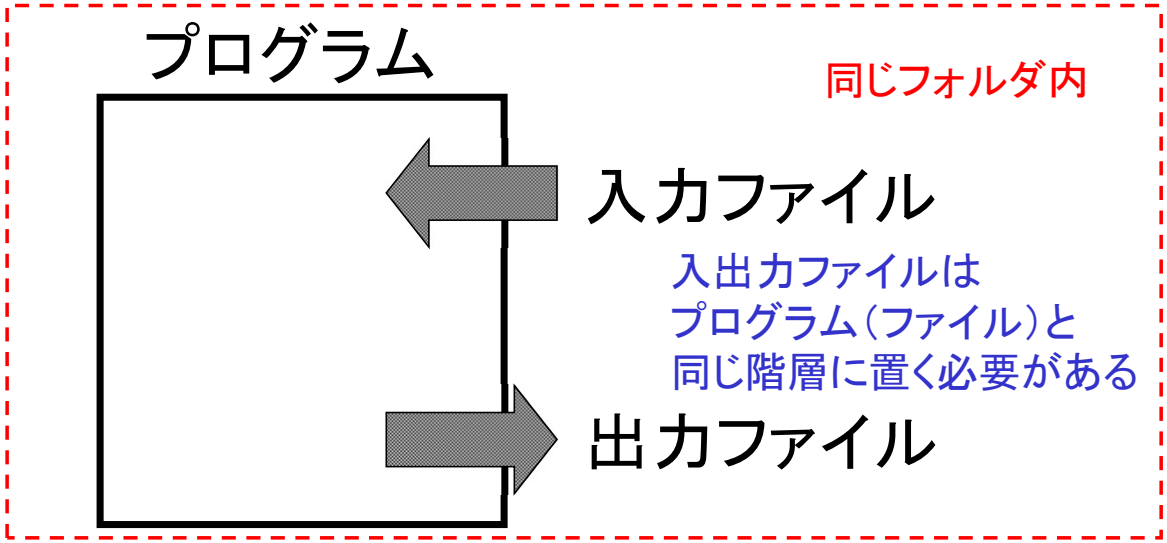
これまでの演習では

プログラム中 or 画面上(input関数)でデータを入力

画面上で結果を出力

入出力データが膨大なとき、

結果を残したいとき、　　どうする？ ➡ ファイル入出力



例題　テキストファイルからデータ読込み

- 次のような名簿ファイル（テキストファイル形式）を読み込んで，１列目の名前と，２列目の住所だけを表示するプログラムを作る

各データは，半角の空白文字（１つまたは複数）で区切られる

Taro Japan
Jiro USA
Hanako UK

この3行のテキストファイルを
“text1.txt”で保存

各自PCのテキストエディタ（メモ帳など）を用いて作成
PC内（デスクトップなど）に保存

ファイル入力　その1

```
filename="test1.txt"
f=open(filename)
data = f.read()
print(data)
f.close()
```

#ファイル名

#ファイル名を指定してファイルを開く

#ファイルの内容をdataに代入

#ファイルを閉じる

実行結果

Taro Japan
Jiro USA
Hanako UK

ファイルをopenしたら、きちんとclose
しないと再度openできなかったり、
リソースを食いつぶしたりしてしまう

※open(filename)は
open(filename,"r") の省略形

"r"は「読み込み操作」を指示

ファイル入力　その２　with – as文

close文が不要なファイル入力

```
filename="test1.txt"
with open(filename) as f:
    data = f.read()
    print(data)
```

コロン：　を
忘れずに付ける

ファイルを開いて以下の
ブロックを実行

ブロックから出るとき自動的にファイルをクローズ

ファイルを操作するブロック

※万一ブロック内でエラーが
生じても自動的にファイル
をクローズしてくれる

実行結果

Taro Japan
Jiro USA
Hanako UK

ファイル出力

“w”は「上書き」を指示
ファイルが無い場合は生成

```
filename="test.txt"
with open(filename,"w") as f:  #ファイルを指定し書き込み
    f.write("Keiko 2100/06/20 France")

with open(filename,"r") as f:
    print(f.read())
```

※指定したファイルが
存在しなければ生成

#ちゃんと生成されているか確認

実行結果

```
Keiko 2100/06/20 France
```

ファイルからデータを読込み、計算結果を別ファイルへ出力

seisu.txt　作成してフォルダ内に置く

```
5    １行目にデータ個数
2    以下、データ
4
6
8
10
```

自動的に作成される

kekka.txt

```
データ個数 5，総和 30，平均値 6.0
```

整数をm個入力し、総和と平均値を求めるプログラム

```
ifname="seisu.txt"
ofname="kekka.txt"
ifn=open(ifname,"r")
ofn=open(ofname,"w")
m=ifn.readline()          # データ個数(1行目のみ読み込む)
m=int(m)                  # 読み込んだデータは文字列→整数に変換
l=[]
for i in range(m):
    data=ifn.readline()   # 各データの読み込み
    data=int(data)
    l.append(data)
s=sum(l)
a=s/m

ofn.write("データ個数 {}，総和 {}，平均値 {}".format(m,s,a))

ifn.close()
ofn.close()
```

文字列中に変数の値を埋め込む書式

ファイル操作関数(1)

- ファイルのオープンとクローズ

  ファイルの読み書きを行う前に，ファイルをオープンしなければならない

  f = open(“ファイル名.拡張子”, 操作)

  ファイルの読み書きが終わったらファイルをクローズしなければならない
  この場合、f.close()　#最後に閉じること（メモリを使っている）

- 操作を指定する

  "r"　　読み込み。ファイルが存在しないときはエラーを出す
  "w"　　書き込み（上書き！）。ファイルが存在しないときは作成
  "a"　　追加（末尾に）書き込み。ファイルが存在しないときは作成
  "r+"　　読み書き両方。ファイルが存在しないときはエラー
  "w+"　　読み書き両方。ファイルが存在しないときは作成　　など

ファイル操作関数(2)

- ファイルの読み込み（“r”）

  f.read()　　ファイルの中身を全て一つの文字列として読み込み

  f.readline()　１行単位の読み込み（改行含む）
  　　読み終えたら空文字列を返す

  f.readlines()　各行の文字列のリストとして読み込み

```
f=open("text1.txt","r")
s = f.readline()
print(s)
s = f.readline()
print(s)
f.close()
```

【注意】改行コードも1要素としてリストに追加される場合がある

【注意】改行コードが1行として認識され、何もない行が出てくる場合がある

これを防ぐため、strip()メソッドを呼ぶと文字列から改行コードを取り除いてくれる

例）　f.readline.strip()

ファイル操作関数(3)

- ファイルへ書き出し（“w”）

  f.write(“文”)　　書き込み（書き換え）。中身は”文”に。

  f.write(str)　　str型（文字列）を書き出し

  f.writelines()　　str型（文字列）のリストを書き出し

【注意】 print() と異なり、write()やwritelines()メソッドはfloat型やint型の変数を引数にできないので、例えば

```
abc=12.3
f.write( str(abc) )
```

このように文字列へ変換して引数とするか、
あるいは文字列中に変数の値を埋め込む書式format()を使うこと。

- ファイルへ追加書き込み（“a”）

  “w”を”a”にして書き込めばよい。

既存のファイルの末尾から追加される

課題：ファイルを読み込み統計量を計算するプログラム

1） 前回の課題で作ったプログラムを利用し、float型のリストを引数として渡すと
それらの最大値，最小値，**平均，標準偏差，メジアン**を返すユーザ定義関数を
作成せよ。それぞれ別の関数としても良いし、関数を1つにして統計量のリスト
を返り値にしても良い。
2） 1行あたり1つの数字が書かれたテキストファイルdata.txtを読み込み、
上記の統計量を計算
3） 上記の計算結果の統計量を別のテキストファイルresult.txtへ書き出す

以上のプログラムを作成せよ. 【注意】statisticsパッケージを使用しないこと。

プログラムの先頭行にコメント文で自分の氏名と学籍番号を入れておくこと

← この記号より左側の文字列はコメントになる

＊データファイルdata.txtの読み込みにおいて、文字コードに関するエラーが解消しない場合、
データファイルの文字コードをSHIFT JISにして保存してみよ。

ヒント：課題プログラムのひな型

```
¥#----最大値を計算するユーザ定義関数：
#    引数はfloat型のリスト
def calcMax( num_list ):
    max_num = max( num_list )
    return max_num

#----平均を計算するユーザ定義関数：
#    引数はfloat型のリスト
def calcAve( num_list ):
    sumf=sum(num_list)
    average = sumf/len( num_list )
    return( average )

#-------プログラム本体
print("統計量を計算します。data.txtから読み込みます")
num_list=[] #数字のリストを作成 最初は空っぽ
#------ファイルを開いてデータをfloat型リストへ取込み
with open("data.txt","r") as datafile:
    while True:
        # lineStr = datafile.readline()
        lineStr = datafile.readline().strip()
        if lineStr=="":
            break
        print( lineStr )
        num_list.append( float(lineStr) )
#------リストに取り込んだデータの統計量計算
num_max = calcMax( num_list ) # medはfloat型
num_ave = calcAve( num_list ) # aveはfloat型
#------ファイルを開いてデータを書き出す
with open("result.txt","w") as resultfile:
    resultfile.write( "Maximum = ")
    resultfile.write( str(num_max) )
    resultfile.write( "¥n")  #改行
    resultfile.write( "Average = ")
    resultfile.write( str(num_ave) )
    resultfile.write( "¥n")  #改行

```

改行コードが1行として認識され、何もない行が出てくる場合がある

Strip()メソッドで改行コードを取り除く

ファイルの全ての行を読み込むと空行になるので、
そのときbreak文でループを抜ける

読み込んだ行の数字の文字列をfloat型に変換してリストに加える

float型の変数num_maxを文字列型に変換してファイルに出力

float型の変数num_aveを文字列型に変換してファイルに出力

まとめ

ユーザ定義関数　**def 関数名():**　　引数、返り値

ファイルのオープン　**ファイル変数名=open(ファイル名,モード)**

ファイルからのデータ読込：　**ファイル変数名.readline()** 1行単位の読み込み

ファイルへのデータ書き出し：　**ファイル変数名.write(文字列変数)** 文字列の書き出し

Strip()メソッドで
改行コード取り除き

第6回　レポート課題提出方法

課題のプログラムを
下記の課題提出用フォルダへ、課題の番号と提出者が分かるようにファイル名を以下のようにしてアップロードせよ
第6回1TE19xxxZ名前.py
https://share.iii.kyushu-u.ac.jp/public/lRbwAAVITI5A2X4BE45t6TqQIE0UQSQUl5Bap_kZ_sjy

講義資料、および上記フォルダへのリンクは下記ホームページから
http://sysplan.nams.kyushu-u.ac.jp/gen/edu/InfoProcess/2019/index.html

プログラムがエラーで動かないとき

エラーメッセージをよく読もう
エラー箇所の行が表示されているはず（その１行前も怪しい）

文字が全角になっていないか確認すること
カッコ（）やクォーテーション””が全角になっていないか？
空白部分に全角の空白が紛れ込んでいないか？

インデントは「文字数」でそろえないとエラーになる