

numpy3

外部ライブラリを使う

■ダンラム (random)

乱数を発生させるには、random モジュールをインポートし、その上で random() を使う必要があります。¹

```
import random as r
rand=r.random()
print(rand)
```

◆ランダムな整数 (randint)

整数 A 以上、整数 B 以下のランダムな整数を返す。たとえば、「サイコロの目」を作るときなど。

```
import random
random.randint(1,6)
```

◆import のいろいろ

前期に教科書で学びました。今一度、復習のつもりで。

・別名をつける

```
import モジュール名 as 別名
```

「numpy」などのような長い名前を「np」などと省略することで書きやすくする。

・モジュール名なしで使えるようにする

```
from モジュール名 import *
```

¹ JavaScript では、random () は Math オブジェクト

「*」(アスタリスク、ワイルドカード)をつけることで、モジュール内のすべての機能をモジュール名なしで使えるようになる。

・インポートした機能に別名をつける

```
from モジュール名 import 機能名 as 別名
```

```
from random import randint as ri  
ri(1,6)
```

◆シーケンス(文字列、リスト、タプルなど)からランダムに選択する方法 (choice)

```
import random as rd  
flavor=['vanilla','chocolate','strawberry']  
rd.choice(flavor)
```

◆要素の順序をランダムに変更する (shuffle)

```
import random as rd  
flavor=['vanilla','chocolate','strawberry']  
rd.shuffle(flavor)  
print(flavor)
```

実施の結果、リスト内の順番が入れ替わっている

```
['vanilla', 'strawberry', 'chocolate']
```

■時刻の取得

time モジュール

```
import time  
time.time()
```

結果は、エポック秒が表示される

◆ローカルタイム (localtime)

```
import time  
time.localtime()
```

結果は、

```
time.struct_time(tm_year=2023, tm_mon=11, tm_mday=13,  
tm_hour=0, tm_min=7, tm_sec=25,  
tm_wday=0, tm_yday=317, tm_isdst=0)
```

※結果を見やすくするため改行を入れています

◆指定時間待つ (sleep)

```
import time  
x=time.time()  
time.sleep(3)  
tt=time.time()-x  
print(tt)
```

3 秒間待った後、取得済みのエポック時間との差を計算する

■タプルを使う意味

比較的少数で固定されたデータは「タプル」を使うのが良い。(たとえば、赤・青・黄、S・M・L、商品名など)

タプル ('burger', 180, 234)

リスト ['burger', 180, 234]

ただし、タプルは要素の「追加」や「削除」ができない。

タプルは、リストに比べ検索などの要素操作の際に処理速度が上がる。
また、「集合」を使えば、さらに早く処理が行われる。

■Numpy で連立方程式の解を求める

■numpy の基本計算

◆計算(1)

$$A = \begin{pmatrix} 1 & 4 & -3 \\ -2 & 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & -3 & 5 \\ -1 & 4 & 1 \end{pmatrix}$$

```
import numpy as np
A=np.array([[1,4,-3],[-2,1,2]])
B=np.array([[2,-3,5],[-1,4,1]])

C=2*A-3*B
print(C)
```

◆計算(2)

$$A = \begin{pmatrix} 2 & 7 & 3 \\ 4 & 3 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 5 & 9 \\ -2 & 3 & 7 \\ 3 & -8 & 6 \end{pmatrix}$$

```
import numpy as np

z1=2-3j
z2=complex(2,-3)
z3=3+0j
print(z1, z2)
print(type(z3))

print(z1.real, z1.imag)
print(type(z1.real),type(z1.imag))
print(z1.conjugate())

A=np.array([[2+3j, -1+2j, 2+1j], [-3 -1j, 3+4j, 2-1j]])
B=np.array([[1+1j, 2j],[-1j, 3+2j], [5-2j, 6+ 1j]])

C=A@B
D=B@A

print(C,D,sep='\n')
print(np.conjugate(C.transpose()))
print(C.T)
```

Python では電気工学で使われる「i」と混同を避けるため、「j」をつかいます。
complete は複素数型の数値を生成するコンストラクタ。

◆連立方程式の解法

$$\begin{cases} x + 3y + 5z = 14 \\ -2x + 5y + z = -6 \\ 3x + 2y + 4z = 16 \end{cases}$$

```
import numpy as np
import numpy.linalg as LA

A=np.array([[1, 3, 5], [-2, 5, 2], [3, 2, 4]])
b=np.array([14, -6, 16])

#A=np.array([[1,3,2,-1],[2,-1,-5,2],[3,2,-3,1],[1,-4,-7,3]])
#b=np.array([5,-2,3,7])

r=LA.matrix_rank(A)
print('rank(A)=',r)
try:
    x=LA.solve(A,b)
    print(x)
except Exception as message:
    print(message)
```

■numpy で外部ファイルを読み込む

Numpy で「csv」ファイルを読み込む

配列=numpy.loadtxt (ファイル名、delimiter=区切り文字、encoding=文字コーディング)

csv ファイルなら delimiter に「,」と指定。空白で区切られている場合には delimiter は省略可。
ASCII のファイルの場合は encoding は省略可。(※日本語が含まれてはいけない)

◆国数英の 3 教科の入っている csv ファイルの読み込み

```
import numpy as np
score=np.loadtxt('seiseki.csv', delimiter=',')
print(score)
```

表示結果：

```
[[ 1. 80. 85. 75.]
 [ 2. 84. 54. 54.]
 [ 3. 84. 84. 84.]
 [ 4. 87. 78. 88.]
 [ 5. 88. 84. 54.]
 [ 6. 98. 88. 84.]
 [ 7. 85. 54. 65.]
 [ 8. 56. 56. 75.]
 [ 9. 84. 54. 65.]
[10. 76. 88. 85.]]
```

1 列目は出席番号。

2 列目の科目から「平均」(mean) を求める。

```
np.mean(score[:,1])
```

※JupyterNoteBook を使っている場合は、続けて打てば、ひとつ前の csv は有効になったまま。

◆行の指定、列の指定

score[:,0]・・・0 列目 score[:,1]・・・1 列目

score[0, :]・・・0 列目 score[1, :]・・・1 列目

◆指定した軸での表示

```
np.mean(score,axis=0)
```

<表示結果>

```
array([ 5.5, 82.2, 72.5, 72.9])
```

各列別の平均が表示される

■統計量

最大値 :

numpy.max(配列)

最小値 :

numpy.min(配列)

平均値 :

numpy.mean(配列)

中央値 :

numpy.median(配列)

分散 :

numpy.var(配列)

標準偏差 :

numpy.std(配列)

■csv ファイルをデータフレームに読み込む

Pandas

<特徴>

- ・ 数値以外のデータも扱える
- ・ 列名を指定して列を操作できる

※Pandas は少量のデータを抽出したり、並べ替えたりするのには非常に便利。それに対して Numpy はビッグデータなどのデータが数の多い場合に使われる。

◆Pandas でのデータ表示

- ・ Pandas のデータは 1 列目が「自動連番」になっている
- ・ 先頭行に、列名が指定できる

※したがって、先ほどのデータは Pandas 用のデータに書き換える必要がある。

```
import pandas as pd
score=pd.read_csv('seiseki2.csv', encoding='utf-8')
print(score)
```