

Python-ECG1-03-bibliothèques-cor

September 29, 2021

1 TP3 - Utiliser une bibliothèque

1.1 1- La bibliothèque numpy

La bibliothèque numpy sera la bibliothèque que nous allons utiliser le plus fréquemment.

Elle contient notamment **toutes les fonctions usuelles**.

Pour charger la bibliothèque, on écrit ceci :

```
[1]: import numpy as np
      # import nom_bibliothèque as alias
```

Fonctions usuelles

Fonction	Syntaxe Python
exp	np.exp(...)
ln	np.log(...)
racine carrée	np.sqrt(...)
valeur absolue	np.abs(...)
partie entière	np.floor(...)

```
[2]: # Exemple : calcul de exp(3)
      np.exp(3)
```

```
[2]: 20.085536923187668
```

Constantes usuelles

Fonction	Syntaxe Python
nombre e	np.e
nombre pi	np.pi

```
[3]: # Pour calcul le nombre e on écrit :
      np.e
```

```
[3]: 2.718281828459045
```

```
[4]: # On peut aussi passer par exp(1)
np.exp(1)
```

```
[4]: 2.718281828459045
```

```
[5]: np.pi
```

```
[5]: 3.141592653589793
```

1.1.1 Exercice 1

Réaliser les calculs suivants :

$$\frac{\sqrt{5+e}}{\pi}$$

$$\ln(|1 - \exp(6)|)$$

$$\left\lfloor \frac{17\pi}{5} \right\rfloor$$

```
[6]: np.sqrt(5+np.e)/np.pi
```

```
[6]: 0.8843220287568424
```

```
[7]: np.log(np.abs(1-np.exp(6)))
```

```
[7]: 5.99751817063104
```

```
[8]: np.floor(17*np.pi/5)
```

```
[8]: 10.0
```

1.1.2 Exercice 2

Réécrire la fonction `second_degre(a,b,c)` donnant les solutions de l'équation $ax^2 + bx + c = 0$ et utilisant la fonction racine carrée.

```
[9]: def second_degre(a,b,c):
    Delta = b**2 - 4*a*c
    if Delta > 0 :
        x1 = (-b-np.sqrt(Delta))/(2*a)
        x2 = (-b+np.sqrt(Delta))/(2*a)
        return x1,x2
    elif Delta == 0 :
        x0 = -b/(2*a)
```

```
    return x0
else:
    return "Pas de solution"
```

1.2 2 - Bibliothèque numpy.random

La bibliothèque **numpy.random** est une bibliothèque faisant partie de numpy et permettant de simuler des phénomènes **aléatoires**.

On commence par charger la bibliothèque avec un alias.

```
[10]: import numpy.random as rd
```

Aujourd'hui nous allons voir comment générer un nombre aléatoire de l'intervalle $]0, 1[$ et un entier aléatoire.

- **rd.random()** : renvoie un nombre aléatoire de $]0, 1[$
- **rd.randint(a,b)** : renvoie un entier aléatoire de $[a, b[$. Attention, b est exclu.

```
[11]: # Exécutez plusieurs fois pour observer l'aléatoire
rd.random()
```

```
[11]: 0.796271666623374
```

```
[12]: # Générer un entier aléatoire compris entre 5 et 10
rd.randint(5,11)
```

```
[12]: 5
```

1.2.1 Exercice 3

- 1) Simuler le lancer d'un dé équilibré à 6 faces.

```
[13]: rd.randint(1,7)
```

```
[13]: 5
```

- 2) Simuler le lancer d'un dé équilibré à 16 faces.

```
[14]: rd.randint(1,17)
```

```
[14]: 10
```

- 3) Déterminer avec Python le *gain final* (mise de départ comprise, donc positif ou négatif) lors de ce jeu :

Une personne joue avec un dé à 16 faces. Il mise en début de jeu une somme S (la mise est une somme donnée donc perdue). Si le résultat est pair et compris entre 1 et 5, il récupère 3 fois sa mise, si le résultat est pair et compris entre 6 et 10, il récupère 2 fois sa mise, si le résultat est pair et compris entre 11 et 16 alors il récupère sa mise et si le résultat est impair, il ne récupère rien.

```
[15]: S = 100 # mise de départ. Tester tous les cas de figure.
de = rd.randint(1,17)
print('dé =', de)

if de == 2 or de == 4:
    Gain = 3*S - S
elif de == 6 or de == 8 or de == 10:
    Gain = 2*S - S
elif de == 12 or de == 14 or de == 16:
    Gain = S - S
else :
    Gain = 0 - S

print('Gain final :', Gain)
```

```
dé = 5
Gain final : -100
```

1.2.2 Exercice 4

Écrire un script python qui simule un tirage dans une urne contenant 5 boules jaunes et 3 boules rouges.

Votre script doit afficher “rouge” ou “jaune” selon le tirage obtenu et doit respecter les probabilités de cette urne.

```
[16]: # rouge = 1,2,3,4,5
# jaune = 6,7,8
numero = rd.randint(1,9)
if numero <= 5:
    print('rouge')
else:
    print('jaune')
```

```
rouge
```

1.3 3 - Pour finir

```
[17]: # À exécuter si vous l'osez...
import antigravity
```

1.3.1 Exercice 5

Un exercice sans bibliothèque pour terminer.

Écrire un script qui précise lequel des trois nombres a , b et c est le plus grand. On traitera aussi les cas d'égalité.

Exemples de réponse : “ a est le plus grand”, “ a,b et c sont égaux”, “ a et b sont égaux et plus grands”, etc.

```
[18]: a = 10
      b = 10
      c = 8

      if a==b and a==c :
          print('a=b=c')
      elif a==b and a > c :
          print('a et b sont égaux et les plus grands')
      elif a==c and a > b:
          print('a et c sont égaux et les plus grands')
      elif b==c and b > a:
          print('b et c sont égaux et les plus grands')
      elif a>b and a>c:
          print('a est le plus grand')
      elif b>a and b>c:
          print("b est le plus grand")
      elif c>a and c>b:
          print("c est le plus grand")
```

a et b sont égaux et les plus grands