

## TP3

## UTILISER UNE BIBLIOTHÈQUE

## I 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 :

```
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(...)

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

## Constantes usuelles

Constante	Syntaxe Python
nombre e	np.e
nombre pi	np.pi

```
# Pour calcul le nombre e on écrit :
np.e
# On peut aussi passer par exp(1)
np.exp(1)
```

**Exercice 1** Réaliser les calculs suivants :

$$\frac{\sqrt{5+e}}{\pi}; \quad \ln(|1 - \exp(6)|); \quad \left\lfloor \frac{17\pi}{5} \right\rfloor.$$

**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.

## II 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.

```
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.

```
# Exécutez plusieurs fois pour observer l'aléatoire  
rd.random()
```

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

**Exercice 3** 1. Simuler le lancer d'un dé équilibré à 6 faces.

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

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.*

**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.

### III 3 - Pour finir

---

```
# À exécuter si vous l'osez....  
import antigravity
```

**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.