

## BIBLIOTHÈQUE RANDOM

### 1 Bibliothèque random

#### ↳ Exemple 1

```
import random
random.random()

for test in range(20):
    random.random()

for test in range(30):
    random.randint(1,6)
```

La bibliothèque `random` permet de charger des fonctions permettant de réaliser des simulations et d'obtenir des nombres pseudo-aléatoires ; néanmoins, avant toute utilisation de ces fonctions, il faut les charger grâce à `import random` parce que les fonctions ne sont pas chargées nativement au lancement de Python.

### 2 Simulations

#### ↳ Exemple 2

1. On simule 100 lancers d'une pièce équilibrée, et on compte le nombre de « face ». Compléter le module Python en conséquence.
2. Modifier le module précédent afin qu'il simule le lancer d'une pièce truquée qui a la probabilité de 0,7 de faire « face » et qui renvoie la fréquence de « pile » après 1 000 lancers.

```
import random
compteur=0
for lancer in range(?):
    aléa = random.random()
    if aléa<?? :
        compteur=???

print(????)
```

#### ↳ Exemple 3

1. Expliquer le rôle de ce module python.
2. Modifier la fonction précédente afin qu'elle simule le lancer de quatre dés équilibrés à 8 faces numérotées de 1 à 8, et donne la fréquence de « 10 » obtenu à l'issue de 1 000 lancers.

```
import random
nb_6=0
for lancer in range(400):
    dé_1=random.randint(1,6)
    dé_2=random.randint(1,6)
    if dé_1+dé_2==6 :
        nb_6=nb_6+1

print(nb_6/400)
```

### 3 Fluctuation

#### ↪ Exemple 4

On lance 400 une pièce truquée qui a la probabilité 0,6 de faire « pile », et on décide de faire 100 fois cette expérience aléatoire en déterminant pour chaque série de 400 lancers la fréquence de « pile ». Le module donné permet d'afficher 100 fréquences obtenues après simulation.

1. Quel phénomène est ainsi illustré ?
2. Déterminer un intervalle de fluctuation de la proportion de « pile ». En quoi a-t-il été illustré par ce module ?

```
import random
Nb_séries=100
nb_lancers=400
Liste=[]

for série in range(Nb_Séries) :
    nb_pile=0
    for lancer in range(nb_lancers) :
        if random.random()<0.6 :
            nb_pile=nb_pile+1
    Liste.append(nb_pile/nb_lancers)
```

### 4 Application

#### → Exercice 1

Proposer un module Python de 200 simulations de l'expérience aléatoire suivante : une urne contient 15 boules indiscernables au toucher, 5 magenta et 10 cyan ; on prélève une boule et on observe sa couleur : si elle est magenta, on a gagné.

#### → Exercice 2

1. Proposer une fonction Python, qui à un nombre  $n$  de lancers, renvoie les fréquences de « 10 » et de « 11 » après  $n$  lancers de trois dés équilibrés à 6 faces numérotées de 1 à 6.
2. La probabilité d'obtenir 10 et celle d'obtenir 11 semblent-elles égales ?

#### → Exercice 3

Pour deux euros, on joue au jeu suivant :

- on lance une pièce équilibrée et on prélève un jeton dans une urne contenant 9 jetons magenta et 1 cyan ;
- si on obtient « face », on gagne un euro ;
- si on tire le jeton cyan, on gagne quatre euros ;
- si on tire le jeton cyan et si on obtient « face », on gagne dix euros.

Proposer une fonction Python, qui à un nombre de parties, renvoie le gain algébrique (qui peut être négatif) obtenu à l'issue des parties.

#### → Exercice 4

En France, 10 % des habitants ont les cheveux blonds.

Proposer une fonction Python, qui à  $N$  échantillons de  $n$  individus, renvoie la liste des  $N$  fréquences d'individus blonds dans chaque échantillon simulé.