

Tirage sans remise et recherche d'une probabilité.



Dans cette activité, on va simuler le tirage d'une boule puis de deux boules sans remise dans une urne contenant des boules blanches, vertes et bleues. Puis on déterminera une valeur approchée de l'événement « tirer 2 boules blanches » dans une urne contenant 5 boules blanches, 3 vertes et 1 bleue.

Dans un script LC

1°) Ecrire une fonction `urne1` qui prend comme paramètres a, b, c des entiers naturels (tous non nuls), qui simule le tirage d'une boule dans une urne contenant a boules blanches, b boules vertes et c boules bleues et qui renvoie « blanc », « vert » ou « bleu ».

2°) Ecrire une fonction `urne2` qui prend comme paramètres a, b, c des entiers naturels (tous plus grands ou égaux à 2), qui simule le tirage de 2 boules sans remise dans une urne contenant a boules blanches, b boules vertes et c boules bleues et qui renvoie le tuple des couleurs des boules tirées.

3°) Ecrire une fonction `evalproba` qui prend comme argument un entier naturel non nul et qui renvoie la fréquence de l'événement « tirer 2 boules blanches » dans une urne contenant 5 boules blanches, 3 vertes et 1 bleue.

```
PYTHON SHELL
>>> urne1(5,6,7)
'blanc'
>>> urne1(5,6,7)
'vert'
>>> urne1(5,6,7)
'bleu'
```

```
PYTHON SHELL
>>> urne2(5,6,7)
('bleu', 'bleu')
>>> urne2(5,6,7)
('bleu', 'bleu')
>>> urne2(5,6,7)
('blanc', 'vert')
```

```
PYTHON SHELL
>>> evalproba(10)
0.2
>>> evalproba(100)
0.21
```



Tirage sans remise et recherche d'une probabilité.



Fonction urne1

1°) h représente un entier aléatoire compris entre 1 et $a+b+c$.

Si $h \leq a$ alors on a tiré une boule blanche.

Sinon si $h \leq a+b$ on a tiré une boule verte

Sinon on a tiré une boule bleue.

On n'oubliera pas d'importer la bibliothèque `random` afin d'utiliser la fonction `randint`.

```
ÉDITEUR : SIMUL2
LIGNE DU SCRIPT 0011
from random import *
def urne1(a,b,c):
    h=randint(1,a+b+c)
    if h<=a:
        c="blanc"
    elif h<=a+b:
        c="vert"
    else:
        c="bleu"
    return c
```

Fonction urne2

2°) On va commencer par tirer une boule de l'urne et on affecter sa couleur dans la variable $t1$.

Selon la couleur de la boule on retire 1 à l'effectif des boules de la couleur correspondante.

On a choisi de ne pas faire le test si ce nouveau nombre est positif car on a supposé que l'urne contient suffisamment de boules de chaque couleur pour pouvoir en tirer 2 de la même couleur.

Puis on tire une seconde boule dans l'urne dont les effectifs sont à jour. On stocke le résultat dans la variable $t2$.

On retourne le couple $t1, t2$.

```
ÉDITEUR : SIMUL2
LIGNE DU SCRIPT 0022
def urne2(a,b,c):
    t1=urne1(a,b,c)
    if t1=="blanc":
        a=a-1
    elif t1=="vert":
        b=b-1
    else:
        c=c-1
    t2=urne1(a,b,c)
    return t1,t2
```

Fonction evalproba

3°) Pour déterminer la fréquence recherchée on va effectuer n fois l'expérience « tirer 2 boules sans remise de l'urne ».

On stocke le résultat (qui est un couple « bleu », « blanc » par exemple) dans la variable `tirage`.

Si les deux valeurs `tirage[0]` et `tirage[1]` sont égales à « blanc » alors on ajoute 1 à la valeur k initialisée à 0 au début de la fonction.

k/n correspond à la fréquence recherchée.

On le vérifie par un calcul : la probabilité recherchée est

$$\frac{\binom{5}{2}}{\binom{9}{2}} = \frac{10}{36} = \frac{5}{18} \approx 0,2777 \dots$$

qui est proche de la fréquence obtenue pour de grandes valeurs de n .

```
ÉDITEUR : SIMUL2
LIGNE DU SCRIPT 0031
def evalproba(n):
    k=0
    for i in range(n):
        tirage=urne2(5,3,1)
        if tirage[0]=="blanc" and tirage[1]=="blanc":
            k=k+1
    return k/n
```

```
>>> evalproba(10000)
0.2755
>>> evalproba(100000)
0.27778
```

