

Exercice du cours :

1. Écrire un algorithme sous la forme d'une fonction simulant le lancer de 100 dés bien équilibrés et calculant la fréquence d'apparition de la face numéro 6.
2. Comment modifier la fonction précédente pour l'appliquer à un nombre quelconque de tirages ?

En français	En python
<pre>fonction frequence6() : nb6=0 Pour n allant de 1 à 100: face est un entier aléatoire compris entre 1 et 6 si face = 6 alors : nb6=nb6+1 freq6=nb6/100 retourner freq6</pre>	<pre>from random import * def frequence6() : nb6=0 for n in range(1,101): face=randint(1,6) if face==6: nb6=nb6+1 freq6=nb6/100 return freq6</pre>

On utilise une boucle « Pour » car on sait combien de fois on devra effectuer une répétition (lancer)

Ici on pourrait mettre `for n in range(100):` (car `n` prendrait les valeurs allant de 0 à 99 donc il prendrait 100 valeurs)

- `random()` est une fonction qui donne un nombre réel de l'intervalle `[0;1[`
- `randint(a,b)` est une fonction qui renvoie un entier de l'intervalle `[a;b]`

`nb6` sera le nombre de fois où la face N°6 sera apparue.

A la fin de la boucle, `nb6` sera l'équivalent de `=NB.SI(A1:A100)` du tableur.

Remarque :

`randint(1,6)` pour être créé avec la fonction `random()`

`random()*6` → nombre aléatoire réel de l'intervalle `[0;6[`

`int(random()*6)` → nombre entier aléatoire de l'intervalle `[0;5]`

donc `randint(1,6)` équivaut à `int(random()*6)+1`

2. On ajoute un argument à la fonction `frequence6` : `nbtirages` qui prendra le rôle de la valeur 100.

On écrira alors : `def frequence6(nbtirages) :`

Dans le programme, il suffit alors de remplacer 100 par `nbtirages` et 101 par `nbtirages+1`

```
from random import *  
def frequence6(nbtirages) :  
    nb6=0  
    for n in range(1,nbtirages+1):  
        face=randint(1,6)  
  
        if face==6:  
            nb6=nb6+1  
  
    freq6=nb6/ nbtirages  
    return freq6
```

L'appel `frequence6(500)` simulera 500 tirages et la fréquence d'apparition de 6 sur ces 500 tirages.

Dernier exercice du cours :

Construire une fonction qui réalisera 500 fois cette expérience et qui déterminera le nombre de fois où f est dans l'intervalle $\left[\frac{1}{15}; \frac{4}{15}\right]$.

Pour répondre à ce problème, nous allons utiliser la fonction `frequence6` qui nous donnera la fréquence d'apparition de la face 6 lors d'une expérience. Ici il faudra en faire 500, donc on devra de nouveau construire une boucle.

Le principe de l'algorithme repose sur :

- on effectue 500 répétitions (donc une boucle « pour ») ;
- au cours de chaque répétition, on lance 100 fois notre dé et on calcule la fréquence d'apparition de la face 6. C'est juste une application de la fonction `frequence6` ;
- à chaque fois, on vérifie si la fréquence obtenue est dans l'intervalle $\left[\frac{1}{15}; \frac{4}{15}\right]$.

Voici l'algorithme (je nomme la fonction `echantillon`). Cette fonction sera écrite après la fonction `frequence6`.

En français	En python
<pre>fonction echantillon() : nbCas=0 Pour n allant de 1 à 500: si frequence6() est dans $\left[\frac{1}{15}; \frac{4}{15}\right]$ alors : nbCas=nbCas+1 retourner nbCas</pre>	<pre>def echantillon() : nbCas=0 for n in range(1,501): if 1/15<=frequence6()<=4/15: nbCas=nbCas+1 return nbCas</pre>

`nbCas` va compter le nombre de fois où la fréquence sera bien dans l'intervalle.

J'ai utilisé la première forme de la fonction `frequence6`, si j'utilisais la deuxième, j'écrirais juste `frequence6(100)`.