

## Simuler le lancer de 3 dés

## Énoncé

On lance 3 dés à 6 faces parfaitement équilibrés et on se demande quelle est la probabilité de réaliser au moins un 6.

1. A l'aide du langage Python, définir la fonction **Lancer3des** qui simule le lancer de 3 dés à 6 faces et renvoie 1 si au moins un des dés réalise un 6.
2. A l'aide du langage Python, définir la fonction **Frequence3des**, qui prend en paramètre le nombre  $n$  d'expériences réalisées et qui renvoie la fréquence  $f$  de réussite (obtenir au moins un 6). Donner la fréquence en pourcentage obtenue pour 1000 expériences.
3. Et en théorie, quel pourcentage devrait-on avoir ?

## 1. Définir la fonction Lancer3des

Il s'agit d'utiliser la notion de compteur en algorithmie pour différentes situations. Ici, si on obtient un 6 à l'aide de la fonction `randint(1,6)`, on augmentera notre variable **compteur** de 1.

On crée un script **LANCERDE** de type **Simulation Aléatoire**. (la librairie **random** sera automatiquement importée). On importe ensuite la librairie **math** à l'aide de `Fns...`, onglet **Modul** et enfin **math...**



```
ÉDITEUR : LANCERDE
math module
Math Const Trig
1:from math import *
2:fabs()
```

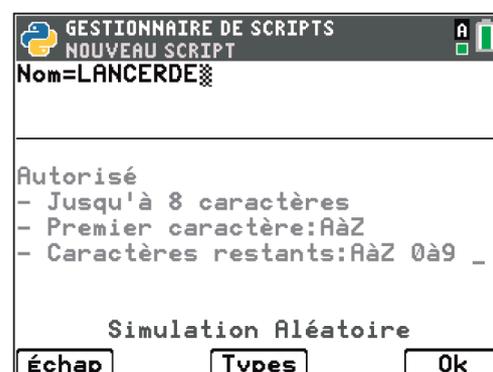
On définit ensuite la fonction **Lancer3des** selon le script ci-contre. On utilise la variable **compteur** (initialisée à 0) pour enregistrer le nombre de fois où l'on a obtenu un 6 à l'aide de la fonction `randint(1,6)`.

L'instruction `for i in range(3)` permet de réaliser nos 3 lancers de dés.

On a préféré l'instruction `compteur+=1` à `compteur = compteur + 1`. La première semble mieux convenir aux élèves pour percevoir son utilité, à savoir augmenter la variable **compteur** de 1. Dans la deuxième instruction la présence de **compteur** de part et d'autre du signe égal pose parfois quelques soucis de compréhension car la variable de droite contient l'ancien contenu et la variable de gauche le nouveau alors que lors de la résolution d'équations, les variables sont identiques dans leur contenu.

A la fin de la boucle **for**, si **compteur** est supérieur ou égal à 1, la fonction **Lancer3des** renvoie 1 sinon elle renvoie 0. On aurait pu choisir une typologie différente pour le retour de la fonction comme **True** ou **False** mais l'objectif étant par la suite de comptabiliser les réussites pour en établir les fréquences, il est plus facile de choisir dès maintenant un retour de type nombre entier.

On mettra à profit pour la saisie du script l'ensemble des menus de la calculatrice qui facilitent la saisie et l'indentation du code.



```
GESTIONNAIRE DE SCRIPTS
NOUVEAU SCRIPT
Nom=LANCERDE

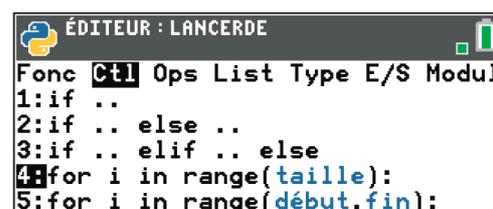
Autorisé
- Jusqu'à 8 caractères
- Premier caractère:AàZ
- Caractères restants:AàZ 0à9 _

Simulation Aléatoire
Échap Types Ok
```



```
ÉDITEUR : LANCERDE
LIGNE DU SCRIPT 0004
# Simulation Aléatoire
from random import *
from math import *

def Lancer3des():
    compteur = 0
    for i in range(3):
        if randint(1,6) == 6:
            compteur += 1
        if compteur >= 1:
            return 1
    else:
        return 0
Fns... a A # Outils Exéc Script
```



```
ÉDITEUR : LANCERDE
Fonc Ctrl Ops List Type E/S Modul
1:if ..
2:if .. else ..
3:if .. elif .. else
4:for i in range(taille):
5:for i in range(début,fin):
```

## 2. Définir la fonction Frequence3des

Il s'agit de poursuivre l'utilisation de la notion de compteur pour évaluer une probabilité en répétant un grand nombre de fois l'expérience « lancer 3 dés » et observer les cas où l'on obtient au moins une fois le chiffre 6.

On va donc réutiliser la fonction **Lancer3des** précédente et compléter notre script avec la fonction **Frequence3des** qui prendra en paramètre  $n$  le nombre de répétitions souhaitées.

Lorsqu'on est en mode console et que l'on souhaite de nouveau modifier notre script, il suffit de sélectionner l'onglet **Editer** à l'aide de la touche  de la calculatrice.

On complète alors notre script avec la fonction **Frequence3des**. Vous remarquerez qu'en allant à la ligne, en repartant de la fonction précédente, l'indentation est conservée (marquée par les points gris). Il faut donc les supprimer à l'aide de la touche  pour faire comprendre à l'environnement Python que l'on souhaite démarrer une nouvelle fonction. Une fois l'opération réalisée, vous pouvez saisir comme d'habitude votre fonction.

L'idée consiste à comptabiliser les résultats (0 ou 1) de **Lancer3des** dans la variable **compteur** et ce  $n$  fois. On retourne la fréquence des réussites.

Dans la console, des appels successifs à la fonction **Frequence3des** pour notamment 1000 puis 100 000 lancers laissent penser que la probabilité de faire au moins un 6 en lançant 3 dés se situe aux alentours de 42%.

```

PYTHON SHELL
>>> from LANCERDE import *
>>> Lancer3des()
1
>>>
>>> Lancer3des()
0
>>> Lancer3des()
1
>>> Lancer3des()
1
>>> |
Fns... a A # |Outils|Editer|Script

```

```

ÉDITEUR : LANCERDE
LIGNE DU SCRIPT 0019
def Frequence3des(n):
    compteur = 0
    for i in range(n):
        compteur += Lancer3des()
    return compteur/n
Fns... a A # |Outils|Exéc|Script

```

```

PYTHON SHELL
>>> # L'exécution de LANCERDE
>>> from LANCERDE import *
>>> Frequence3des(10)
0.4
>>> Frequence3des(100)
0.42
>>> Frequence3des(1000)
0.413
>>> Frequence3des(10000)
0.4246
>>> |
Fns... a A # |Outils|Editer|Script

```

```

PYTHON SHELL
>>> Frequence3des(100000)
0.42165
>>> Frequence3des(100000)
0.42076
>>> Frequence3des(100000)
0.42166
>>> Frequence3des(100000)
0.42237
>>> Frequence3des(100000)
0.4218
>>> |
Fns... a A # |Outils|Editer|Script

```

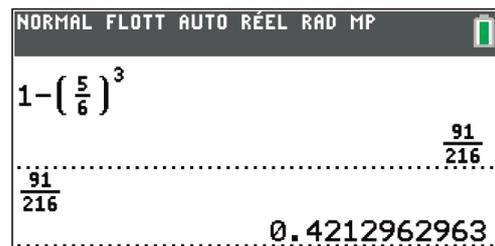
## 3. Et en théorie ?

Lors du lancer de 3 dés à 6 faces parfaitement équilibrés, on note  $A$  l'événement « obtenir au moins un 6 ».

Ainsi  $\bar{A}$  est l'événement « ne pas obtenir de 6 ».

Les 3 lancers sont indépendants on a  $p(\bar{A}) = \left(\frac{5}{6}\right)^3$ .

On obtient donc  $p(A) = 1 - p(\bar{A}) = 1 - \left(\frac{5}{6}\right)^3 = \frac{91}{216} \approx 42\%$ .



$$1 - \left(\frac{5}{6}\right)^3 = \frac{91}{216} \approx 0.4212962963$$