

## Simulation de la décroissance radioactive - Python

**Exercice 1** : jeu de dé

```
from random import *  
  
#utilisation de la bibliothèque random  
print(randint(1,6))  
  
#affiche un nombre aléatoire de 1 à 6
```

Tester le programme : il joue aux dés avec un tirage aléatoire de 1 à 6

**Exercice 2** : nombre de jets pour tirer un 6

```
from random import *  
  
# utilisation de la bibliothèque random  
n=0  
d=0  
  
# initialisation des variables : n : nombre de jets et d : résultat du dé  
while d < 6 :  
    n=n+1  
    d = randint(1,6)  
    print(d)  
  
# tant que mon score n'est pas 6, on lance le dé et on compte de nombre de coups.  
print("Le nombre de jets est de ",n)  
  
# Affichage du score final une fois qu'on a tiré un 6.
```

On remarque que sur un seul essai il est impossible de prévoir combien de jets il faudra pour avoir un 6 même si on sait que ce tirage à une probabilité de  $1/6 = 16,7\%$  de se produire à chaque fois.

Par analogie : quand on a un seul atome radioactif, il est impossible de prévoir quand il va se désintégrer.

**Exercices 3** : probabilité lors d'un grand nombre de jets d'avoir un 5

```
from random import *  
  
# utilisation de la bibliothèque random  
Nbrelancers=10  
  
# nombre à modifier par la suite  
Compteur=0  
  
for i in range (0,Nbrelancers) :  
    De = randint(1,6)  
  
    # pour les petits Nbrelancers on peut ajouter print(De)  
  
    if De==5 :  
        Compteur=Compteur+1  
  
print ("Nombre de 5 obtenus = ", Compteur)  
  
Proba = Compteur/Nbrelancers  
print("La probabilité est de ",Proba)
```

Essayer le programme en modifiant ensuite la valeur du nombre de lancers : 100 1000 10000 100000

On retrouve la probabilité de 1/6 (16,6%) pour les grands nombres de lancers.

Par analogie, quand on a un grand nombre d'atomes radioactifs, on peut suivre l'évolution de l'échantillon connaissant la probabilité que l'atome se désintègre.

**Exercices 4** : Simulation de la décroissance radioactive sur un graphique

```
from matplotlib import pyplot  
from random import randint  
  
# utilisation de la bibliothèque random et graphique  
n = 10000  
p = 10  
  
for k in range(p+1):  
    pyplot.plot(k,n,'k.')  
    for i in range(n):  
        d = randint(1,2)  
        if d == 2 :  
            n = n-1  
  
axes = pyplot.gca()  
pyplot.grid()  
axes.set_xlim(0, 10)  
axes.set_ylim(0, 10000)  
pyplot.show()
```