

Simulation Lièvre et Tortue sous Python : différents scripts

Simulation d'un lancer d'un dé ou d'une pièce

```
import random as rd
def de6(): #lancer de dé à 6 faces
    l=rd.randint(1,6)
    return l
```

```
import random as rd
def de(p): #lancer de pièce pile le lièvre a
gagné
    a=rd.random() #flotant aléatoire entre 0
et 1
    if a<=p:
        return 1 #pile
    else:
        return 0
```

Simulation d'une course (deux modèles possibles) :

```
def partie(n) : #nombre de cases (loi binomiale)
```

```
    T=0 #position de la tortue sur le parcours
```

```
    L=0
```

```
    for i in range(n):
```

```
        if de6()<6:
```

```
            T=T+1
```

```
        else:
```

```
            L=L+1
```

```
    if n==L :
```

```
        return 1 #le lièvre a gagné
```

```
    else:
```

```
        return 0
```

```
def course(n) : #nombre de cases (loi géométrique tronquée)
```

```
    for i in range(n):
```

```
        if de6()==6:
```

```
            return 1#le lièvre a gagné
```

```
    return 0
```

```
def course2(n):
```

```
    i=0
```

```
    while de6()<6 and i<n:
```

```
        i=i+1
```

```
    if i==n:
```

```
        return 0
```

```
    return 1#le lièvre a gagné
```

Calcul de fréquence et affichage nuage de points (n, fréquence(lièvre))

```
def FrequenceL(k,n):
    nbL=0
    for i in range (k):
        if partie(n)==1:
            nbL=nbL+1
    return nbL/k
Max=2000
Lk=[k for k in range(2,Max,1)]
LF=[FrequenceL(k,4) for k in Lk]
plt.axis([0,Max,0,1])
plt.plot(Lk,LF,'*')
```

Prolongement

```
def coursep(n,p):
    i=0
    while de(p)==0 and i<n:
        i=i+1
    if i==n:
        return 0
    return 1

def freqpL(k,n,p): #p: proba de pile.
    nbL=0
    for i in range(k):
        nbL=nbL+coursep(n,p)
    return nbL/k
import matplotlib.pyplot as plt
Lx=[k/100 for k in range(101)]
Ly=[freqpL(100000,4,p) for p in Lx]
plt.plot(Lx,Ly)
```