

```

s = experience()
n = 1
L = [s] # moyenne su
while n < nExperiences:
    n = n+1
    s = s + experience
    L.append(s/n) # o
plt.plot(list(range(1,
plt.plot([1,nExperienc

```

LA COLLECTION D'IMAGES

Toto collectionne les images qu'il trouve dans les paquets de ses biscuits préférés.

On suppose que les images ont été disposées au hasard dans les paquets et que chaque paquet contient exactement une image.

La collection comporte 10 images différentes numérotées de 1 à 10.



L'objectif est d'étudier le nombre de paquets de biscuits qu'il faudra acheter pour obtenir la collection complète.

On crée une liste contenant 10 zéros.

Lorsque Toto obtient une image qu'il ne possédait pas, l'élément correspondant de la liste passe de 0 à 1.

1) En exécutant un programme, on obtient l'affichage suivant :

```

>>>
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0]
[0, 0, 0, 0, 1, 0, 1, 1, 0, 0]
[0, 0, 0, 0, 1, 1, 1, 1, 0, 0]
[0, 0, 1, 0, 1, 1, 1, 1, 0, 0]
[0, 0, 1, 0, 1, 1, 1, 1, 0, 0]
[0, 0, 1, 0, 1, 1, 1, 1, 0, 1]
[0, 0, 1, 0, 1, 1, 1, 1, 0, 1]
[0, 1, 1, 0, 1, 1, 1, 1, 0, 1]
[0, 1, 1, 0, 1, 1, 1, 1, 0, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 0, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 0, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 0, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 0, 1, 1, 1, 1, 1, 1]
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
21

```

- Pour cette simulation, combien de paquets de biscuits faudra-t-il acheter pour obtenir la collection complète ?
- Combien d'images différentes possède Toto à l'ouverture du 7^{ème} paquet ?
- Combien de paquets aura-t-il été nécessaires pour obtenir l'image n°9 ?

2) Syntaxe pour générer un nombre entier aléatoire :

Langage naturel	TI	CASIO	Python
Générer un nombre entier aléatoire de l'intervalle [1,6]	randInt(1,6)	RanInt#(1,6)	randint(1,6)

En Python : La commande **randint** nécessite l'appel d'un module au début du programme.
Pour ce faire, taper : **from random import***

On a écrit dans différents langages de programmation un algorithme permettant de simuler l'obtention aléatoire d'images jusqu'à avoir la collection complète :

TI	CASIO	Python (*)
<pre> :ClrList L1 :10→dim(L1) :0→N :0→I :While I<10 :N+1→N :randInt(1,10)→X :If L1(X)=0 :Then :1→L1(X) :I+1→I :End :Disp L1 :End :Disp N </pre>	<pre> 10→Dim List 1↵ 0→N↵ 0→I↵ While I<10↵ N+1→N↵ RanInt# (1,10)→X↵ If List 1[X]=0↵ Then 1→List 1[X]↵ I+1→I↵ IfEnd↵ List 1↵ WhileEnd↵ N↵ </pre>	<pre> from random import* L=10*[0] n=0 i=0 while i<10: n=n+1 x=randint(0,9) if L[x]==0: L[x]=1 i=i+1 print(L) print(n) </pre>

(*) En Python, les éléments de la liste ne sont pas numérotés de 1 à 10 mais **de 0 à 9**.

- Quel est le rôle de la variable n (ou N) ?
- Quel est le rôle de la variable i (ou I) ?
- A l'aide d'une calculatrice ou d'un logiciel, tester de nombreuses fois un des programmes.

Calculer le nombre moyen de paquets nécessaires pour obtenir la collection complète. On pourra mettre en commun les résultats de la classe.

3) On souhaite maintenant pouvoir renouveler l'expérience précédente 500 fois de suite et calculer le nombre moyen de paquets nécessaires obtenus à chaque expérience.

Adapter et tester le programme précédent en conséquence.



Hors du cadre de la classe, aucune reproduction, même partielle, autres que celles prévues à l'article L 122-5 du code de la propriété intellectuelle, ne peut être faite de ce site sans l'autorisation expresse de l'auteur.

www.maths-et-tiques.fr/index.php/mentions-legales