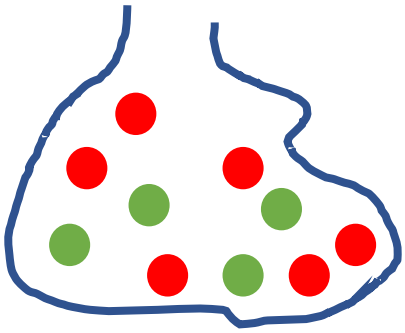
Unité 3 : Débuter la programmation en Python	Compétence 2 : La boucle bornée For
Dans cette seconde leçon de l'unité 3, vous allez découvrir comment répéter un processus ou un ensemble d'instructions en utilisant une boucle bornée **FOR**.	**Objectifs :** • Appliquer une fonction. • Découvrir et mettre en œuvre la boucle bornée FOR. • Utiliser la boucle FOR dans des exemples simples.

Constituer un échantillon :

Un sac opaque contient six jetons rouges et quatre jetons verts. On tire au hasard un jeton du sac, on note sa couleur puis on le replace dans le sac.

Programmer une fonction **couleur()** simulant une expérience aléatoire de variable.

- Quelles sont les valeurs possibles prises par la variable x ?
- On souhaite écrire un script qui permette de distinguer les boules rouges des vertes à l'aide d'un test.



- Commencer un nouveau script et le nommer « **ECHANTIL** »
- Comme vous travaillez sur des nombres aléatoires, le chargement de la bibliothèque « random » est nécessaire. Pour cela, appuyer sur l'onglet **F1** (**Fns…**) et choisir le menu **Modul** puis **2 : random…**
- Entrer le script ci-contre dans l'éditeur en veillant à respecter l'indentation.

```
ÉDITEUR : ECHANTIL
LIGNE DE SCRIPT 0001
from random import *
def couleur():
  x=randint(1,10)
  if x<=6:
    c="Rouge"
  else:
    c="Vert"
  return c

Fns...| a A # |Outils| Exéc |Script
```

- Afficher le résultat de la fonction dans la console.
- Exécuter le script plusieurs fois en appelant la fonction **couleur( )**.

```
PYTHON SHELL
>>> couleur()
'Rouge'
>>> couleur()
'Rouge'
>>> couleur()
'Rouge'
>>> couleur()
'Vert'
>>> couleur()
'Rouge'
>>> |
Fns...| a A # |Outils|Éditer|Script
```

Conseil à l'enseignant : ce script peut être modifié pour être appliqué sur un nombre quelconque de jetons. Dans ce cas-là, on pourra définir **couleur(n,a)** où n est le nombre de jetons et a le nombre de jetons rouges.

Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

Appliquons nos connaissances : Échantillonnage et prise de décision :
On a réalisé un sondage à la sortie du nouveau spectacle proposé par un artiste. Ce sondage réalisé dans une grande ville montre que les deux tiers des personnes ayant vu le spectacle l'ont aimé. L'agent de l'artiste pense que toute la population française est dans le même état d'esprit. Il commande une enquête auprès d'un institut de sondage pour le vérifier.

Étude de la population :
Pour des besoins de l'enquête statistique, l'institut de sondage doit créer une fonction qui simule la réponse à la situation.
Votre travail consiste à créer cette fonction en respectant le cahier des charges suivant :

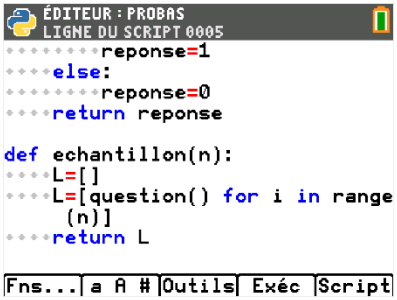
- Le spectacle a été apprécié, avec une probabilité $p = \frac{2}{3}$.
- Le spectacle n'est pas apprécié, avec une probabilité $p = \frac{1}{3}$.

1. Commencer un nouveau script et le nommer **PROBAS**.
2. Écrire cette fonction dans l'éditeur et la tester plusieurs fois en appuyant sur **F4** (Exec) puis en tapant dans la console le nom de la fonction sans arguments : **question()**

Astuce : on peut utiliser la touche **VAR** puis flèche du haut pour répéter l'exécution, (c'est peut-être plus rapide que de taper le nom de la fonction…)

Simulation d'un échantillon de taille *n* :
L'institut de sondage souhaite simuler des échantillons de taille variable.
Vous devez donc créer dans le script courant une fonction **échantillon(n)** permettant de poser la question à un échantillon de taille *n*.

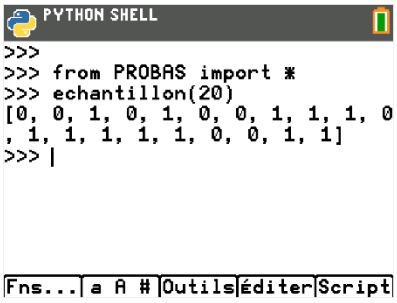
- Pour cela, créer une liste vide **l**.
- Remplir cette liste en utilisant la fonction **question()** et en utilisant une boucle **For**.



Conseil à l'enseignant : Le langage Python permet d'utiliser une fonction pour remplir une liste incrémentée par une boucle For, comme arguments de la liste.
La réalisation de cet exemple est possible sans l'utilisation de listes. Les scripts sont alors à modifier légèrement en remplaçant les instructions relatives aux listes par des boucles bornées incrémentant une variable.

Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

- Tester la fonction **echantillon(n)** pour un échantillon de taille 20, en exécutant le script et en tapant dans la console « echantillon(20) ».



Validation de l'échantillon :

L'institut de sondage souhaite déterminer le pourcentage d'échantillons dont la fréquence de personnes qui ont apprécié le spectacle appartient à l'intervalle de fluctuation à 95% de $p = \frac{2}{3}$

A la suite du script précédent, il vous est demandé de créer deux fonctions :

a) **freq_echantillon(n)** qui calcule automatiquement la fréquence des réponses 1 observée dans l'échantillon **L** de taille *n*.

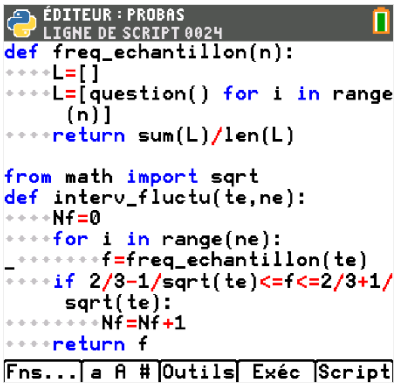
b) **interv_fluctu(te,ne)** pour déterminer si la fréquence d'un échantillon de taille *n* est compris dans l'intervalle de fluctuation à 95%.

Algorithme

Fonction interv_fluctu(taille ech , nbre ech)
  Nf ← 0
  Pour *i* allant de 1 à nbre ech faire
    f← freq_echantillon(taille ech)
    Si *f* appartient à $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}}\right]$ alors
      $Nf \leftarrow Nf + 1$
    Fin si
  Fin Pour

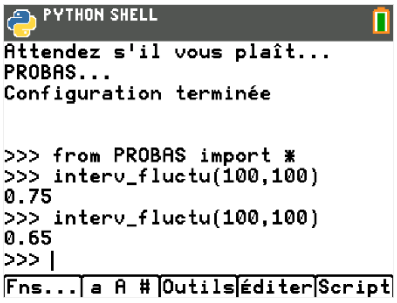
Conseil à l'enseignant : On rappelle que pour un caractère dont la proportion dans une population donnée est *p*.
Pour $n \geq 25$ et $0.2 \leq p \leq 0.8$, la fréquence du caractère dans les échantillons de taille *n* appartient à l'intervalle $\left[p - \frac{1}{\sqrt{n}} ; p + \frac{1}{\sqrt{n}}\right]$ dans 95% des cas.
Cet intervalle est appelé intervalle de fluctuation à 95%.

A partir de la fonction *interv_fluctu* écrit en langage naturel, vérifier que vous obtenez la fonction interv_fluctu en Python.



Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

- Tester le script plusieurs fois pour un échantillon de taille 100.
- En déduire pour un exemple, l'intervalle de fluctuation à 95%.
- Cette étude statistique remet-elle en question l'affirmation de l'agent de l'artiste ?



Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/