



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 :

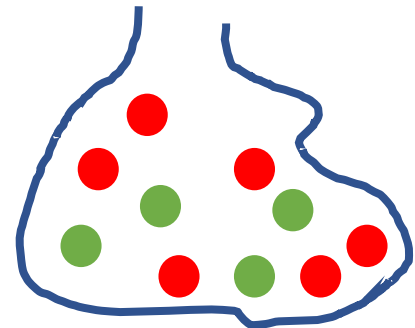
- Appliquer une fonction.
- Découvrir et mettre en œuvre la boucle bornée **FOR**.
- Utiliser la boucle **FOR** dans des exemples simples.

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 cette expérience aléatoire de variable X .

- 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 **[menu]** puis **6 Nombres aléatoires**.
- Entrer le script ci-contre dans l'éditeur en veillant à respecter l'indentation.

```

1.1 | *Classeur | RAD | 8/8
from random import *
def couleur():
    x=randint(1,10)
    if x<=6:
        c="Rouge"
    else:
        c="Vert"
    return c

```

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

```

1.1 | 1.2 | *Classeur | RAD | 10/10
Shell Python
>>>#Running Echantil.py
>>>from Echantil import *
>>>couleur()
'Vert'
>>>
>>>couleur()
'Vert'
>>>couleur()
'Rouge'
>>>

```

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.



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 les 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'a pas été 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 **ctrl** **R** puis en tapant dans la console le nom de la fonction sans arguments : **question()**.

```
1.1 1.2 1.3 *Classeur RAD 3/8
*Probas.py
from random import *
def question():
    s=randint(0,2)
    if s==0 or s==1:
        reponse=1
    else:
        reponse=0
    return reponse
```

Astuce : on peut utiliser la touche **var** puis flèche du haut pour répéter l'exécution, (c'est bien 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**.

```
1.1 1.2 1.3 *Classeur RAD 13/13
*Probas.py
s=randint(0,2)
if s==0 or s==1:
    reponse=1
else:
    reponse=0
return reponse
def echantillon(n):
    L=[]
    L=[question() for i in range(n)]
    return L
```

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.





10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

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

UNITE 3 : COMPETENCE 2

NOTES DU PROFESSEUR

```

1.2 1.3 1.4 *Classeur RAD 6/6
Shell Python
>>>#Running Probas.py
>>>from Probas import *
>>>echantillon(20)
[1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1]
>>>

```

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 :

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

$Nf \leftarrow 0$

Pour *i* allant de 1 à nbre ech faire

$f \leftarrow \text{freq_echantillon}(\text{taille_ech})$

 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.

```

1.2 1.3 1.4 *Classeur RAD 25/27
*Probas.py
def freq_echantillons(n):
    L=[]
    L=[question() for i in range(n)]
    return sum(L)/len(L)

from math import sqrt
def interv_fluctu(te,ne):
    Nf=0
    for i in range(ne):
        f=freq_echantillons(te)
        if 2/3-1/sqrt(te)<=f<=2/3+1/sqrt(te):
            Nf+=1
    return f

```





10 Minutes de Code

TI - NSPIRE™ CX II & TI - PYTHON

- Appuyer sur `var` afin de rappeler la fonction `interv_fluctu`.
- 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 ?

UNITE 3 : COMPETENCE 2

NOTES DU PROFESSEUR

```
1.2 1.3 1.4 *Classeur RAD 9/9
Shell Python
>>>interv_fluctu(100,100)
0.6600000000000001
>>>interv_fluctu(100,100)
0.6800000000000001
>>>interv_fluctu(100,100)
0.7
>>>interv_fluctu(100,100)
0.75
>>>|
```

