

# Du tableur vers Python

L'objectif de ce document est de présenter une activité qui permet d'introduire la programmation Python à partir du tableur. Elle a été faite en seconde sur plusieurs séances, en demi-groupe et en salle informatique.

L'activité a été proposée au format numérique sur la plateforme Moodle du lycée. Elle aurait très bien pu être distribuée au format papier. L'essentiel était que les élèves soient autonomes et travaillent à leur rythme pour s'approprier au mieux les notions abordées. Un accès à Internet était permis. Cela leur permettait de chercher comment coder certaines fonctions tableur ou Python.

## Pourquoi passer du tableur à Python ?

Ce sont deux utilisations complémentaires. Comme nous allons le voir, le tableur permet de décomposer les différentes étapes d'un calcul ou d'un programme tout en offrant la visualisation immédiate des résultats des fonctions utilisées. Cela aide les élèves à comprendre l'algorithme qu'ils ont à programmer en Python qui est plus performant que le tableur lorsque l'on veut simuler des échantillons de très grande taille.

**Thème** : Simulation d'une expérience aléatoire

## Notions abordées

Voici une liste non exhaustive de notions abordées :

- La fonction random et la création de nombres pseudo-aléatoires (distribution uniforme)
- La fonction partie entière (sa définition et son utilisation)
- Composition de fonctions
- Création de la simulation d'une expérience aléatoire
- Utilisation d'une simulation pour conjecturer des probabilités
- Notion et observation de fluctuation de fréquences (loi des grands nombres)
- De la moyenne d'une série statistique à la notion d'espérance mathématiques
- Manipulation du tableur : utilisation du \$, fonctions random(), ent(...), somme(...), nb.si(... ;...), ...
- Programmation Python : fonctions Python (procédure et fonctions), liste Python, fonctions floor(...) et random() , structure for (en donnant du sens à range(...)), structure conditionnelle
- ...

## Description

Dans cette activité, on souhaite étudier le nombre de filles dans les familles de 4 enfants.

On fait les hypothèses **(1)** suivantes :

- Chaque naissance a autant de chances d'être celle d'un garçon ou celle d'une fille.
- Le sexe d'un enfant d'une famille ne dépend pas du sexe des enfants précédents.

Population étudiée : Famille de 4 enfants

Caractère étudié sur cette population : nombre de filles

Événements élémentaires (résultats possibles) : le nombre de fille appartient à  $\{0 ; 1 ; 2 ; 3 ; 4\}$

On souhaite simuler un très grand nombre de familles de manière à répondre aux questions posées par les élèves :

- Quel est l'événement élémentaire le plus fréquent ?
- Quel est l'événement élémentaire le plus probable ?
- Quelles sont les probabilités des différents événements élémentaires ?
- Quel est le nombre moyen de fille dans une famille de 4 enfants ?

...

Pour cela, on se donne comme objectif de remplir les tableaux suivants.

Résultats possibles	0	1	2	3	4	Somme
Effectifs	.....	.....	.....	.....	.....	.....
Fréquences	.....	.....	.....	.....	.....	.....

Moyenne	.....
---------	-------

Pour construire cette simulation, on commence, dans un premier temps, par utiliser le tableur pour comprendre comment simuler, pour décomposer les différentes étapes et pour visualiser les résultats intermédiaires et les résultats finaux. Dans un deuxième temps, on utilise Python pour obtenir des échantillons de tailles de très grande taille et des résultats plus précis

Les élèves ont à leur disposition une activité sous format numérique, un fichier tableur (à ouvrir par exemple avec libre Office) et un modèle de fichier Python (à ouvrir par exemple avec Pyscripter que l'on trouve dans la version portable de Python).

## Les différentes étapes

L'activité a été faite sur plusieurs séances, en classe de seconde, en demi-groupe et en salle informatique. Elle peut se décomposer en 6 étapes.

**Etape 1** : Simulation d'une naissance

**Etape 2** : Simulation d'une famille de 4 enfants

**Etape 3** : Calcul du nombre de filles dans une famille de 4 enfants

**Etape 4** : Création d'un échantillon de plusieurs familles

**Etape 5** : Tableau des effectifs et des fréquences

**Etape 6** : Calcul de la moyenne de la série

La suite de ce document est la synthèse des 6 étapes.

## Etape 1 : Simulation d'une naissance

### Avec le tableur :

Dans le premier onglet du tableur fourni (onglet Naissance), on recherche la formule qui permet de simuler une naissance, c'est à dire l'apparition aléatoire des mots « Fille » et « Garçon » en respectant les hypothèses (1) énoncées plus haut.

	A	B	C	D	E	F	G	H	
1	<b>Expérience aléatoire :</b>								
2	On souhaite simuler la naissance d'un enfant.								
3									
4									
5	<b>Exercice 1 :</b> Commençons programmer l'apparition aléatoire d'un 0 ou d'un 1								
6									
7	Dans la cellule A12, créer un nombre pseudo-aléatoire dans ]0;1[ (*)								
8	Dans la cellule B12, transformer le nombre de la cellule A12 en un nombre dans l'intervalle ]0;2[								
9	Dans la cellule C12, transformer le nombre de la cellule B12 en un nombre de l'ensemble {0,1}								
10	Dans la cellule E12, faites la même chose mais dans une seule cellule.								
11									
12	0,63	1,26	1		0				
13									
14									
15	<b>Exercice 2 :</b> On veut simuler la naissance d'un enfant.								
16									
17	Recopier la formule obtenue en E12 dans la cellule A21								
18	En utilisant la structure conditionnelle du tableur (*), simuler la naissance d'une Fille ou d'un Garçon dans la cellule B21.								
19	Dans la cellule D21, faites la même chose mais dans une seule cellule.								
20									
21	0	Fille		Garçon					
22									
23									
24		(*) On trouvera sur Internet la fonction qui permet de réaliser la question.							

La formule à saisir dans la cellule D2 est :

```
=si(ent(2*alea())=0;'Fille';'Garçon')
```

Ceci étant fait, dans le deuxième onglet du fichier tableur, on recopie cette formule dans la cellule B2.

	A	B
1		<b>Naissance 1</b>
2	<b>Famille 1</b>	<b>Fille</b>

### Avec Python :

En utilisant ce qui a été compris dans le tableur, on souhaite coder en Python la fonction Python qui permet de simuler une naissance.

Pour cela on utilise les fonctions `floor(...)` de la bibliothèque `math` et `random()` de la bibliothèque `random`.

```
def naissance():
    nb=floor(2*random())
    naissance=""
    if(nb==0):
        naissance="Fille"
    else:
        naissance="Garçon"
    return naissance

#Création de la fonction qui simule une naissance
# permet d'obtenir 0 ou 1 de manière aléatoire et équiprobable
# Création d'une variable naissance de type chaîne de caractère
# si nb est égale à 0
# alors naissance prend la valeur Fille
#sinon naissance prend la valeur Garçon
# la fonction retourne la valeur de la variable naissance
```

## Etape 2 : Simulation d'une famille de 4 enfants

### Avec le tableur :

Dans le deuxième onglet du fichier tableur, on copie la fonction découverte dans l'étape 1 dans les 4 cellules B2, C2, D2 et E2 pour simuler une famille de 4 enfants.

	A	B	C	D	E
1		<b>Naissance 1</b>	<b>Naissance 2</b>	<b>Naissance 3</b>	<b>Naissance 4</b>
2	<b>Famille 1</b>	<b>Fille</b>	Fille	Fille	Fille

### Avec Python :

On construit la fonction Python qui permet de simuler une famille de 4 enfants.

On introduit à cette occasion la méthode `append (...)` qui permet d'ajouter un élément à une liste.

```
def famille(nbNaissance):
    listeEnfant=[]
    for i in range(nbNaissance):
        genre=naissance()
        listeEnfant.append(genre)
    return listeEnfant
```

# La fonction famille prend en paramètre le nombre de naissance  
# Création d'une variable de type liste nommée listeEnfant  
# Pour chaque naissance :  
# on simule une naissance et on l'affecte à la variable genre  
# on ajoute l'élément genre dans liste listeEnfant  
# on retourne la liste des enfants.

## Etape 3 : Calcul du nombre de filles dans une famille de 4 enfants

### Avec le tableur :

Dans la cellule F2 on compte le nombre de filles dans la famille simulée dans la plage B2:E2.

Pour cela on utilise la fonction `=nb.si (...; ...)` en saisissant `=nb.si (B2:E2; ''Fille'')`

	A	B	C	D	E	F
1		<b>Naissance 1</b>	<b>Naissance 2</b>	<b>Naissance 3</b>	<b>Naissance 4</b>	<b>Nb de filles</b>
2	<b>Famille 1</b>	<b>Fille</b>	Fille	Fille	Fille	<b>4</b>

### Avec Python :

On introduit à cette occasion la méthode `count (...)` qui donne le nombre d'occurrence de la liste qui lui est passée en paramètre.

```
def nbFille(liste):
    nb=0
    for i in range(4):
        nb=liste.count("Fille")
    return nb
```

#Fonction qui compte le nombre de fois où la valeur fille apparaît dans la liste  
# Pour i prenant les valeurs 0, 1, 2, 3  
# Compte le nombre d'occurrence de « Fille » dans liste  
# retourne la valeur de nb

## Etape 4 : Création d'un échantillon de plusieurs familles

### Avec le tableur :

On fait un simple copier-coller. Dans notre cas, on a simulé 1000 familles.

	A	B	C	D	E	F
1		<b>Naissance 1</b>	<b>Naissance 2</b>	<b>Naissance 3</b>	<b>Naissance 4</b>	<b>Nb de filles</b>
2	<b>Famille 1</b>	Fille	Fille	Fille	Fille	<b>4</b>
3	<b>Famille 2</b>	Fille	Fille	Garçon	Fille	3
4	<b>Famille 3</b>	Fille	Garçon	Garçon	Fille	2
5	<b>Famille 4</b>	Fille	Garçon	Fille	Garçon	2
6	<b>Famille 5</b>	Fille	Fille	Garçon	Garçon	2
7	<b>Famille 6</b>	Fille	Garçon	Garçon	Garçon	1
8	<b>Famille 7</b>	Fille	Garçon	Fille	Garçon	2
9	<b>Famille 8</b>	Fille	Garçon	Fille	Fille	3
10	<b>Famille 9</b>	Fille	Garçon	Garçon	Garçon	1
11	<b>Famille 10</b>	Garçon	Garçon	Garçon	Garçon	0

### Avec Python :

On souhaite construire la fonction qui permet de construire la colonne F du tableur, c'est à dire la liste du nombre de fille dénombrées dans les familles simulées.

On utilise, à nouveau, à cette occasion la méthode `count (...)` qui donne le nombre d'occurrence de l'élément passé en paramètre dans une liste.

```
def echantillon(tailleEchantillon): # Simulation d'un échantillon de taille : tailleEchantillon
    n=tailleEchantillon # on déclare une variable n à laquelle on affecte tailleEchantillon
    listeNbFille=[] # on déclare une liste listeNbFille
    for i in range(n): # On répète n fois :
        f=famille(4) # on simule une famille
        nb=nbFille(f) # on compte le nombre de fille dans cette famille
        listeNbFille.append(nb) # on stocke le nombre obtenu dans la liste listeNbFille
    return listeNbFille # la fonction retourne la liste listeNbFille
```

## Etape 5 : Tableau des effectifs et des fréquences

### Avec le tableur :

Dans la plage H2:N4 on souhaite remplir le tableau des effectifs et des fréquences de notre simulation de taille 1000 comme proposé en page 1.

Pour cela on doit compter le nombre de fois où les valeurs 0 puis 1, puis 2, puis 3, puis 4 sont contenues dans la plage F2:F1001.

On utilise pour cela la fonction =nb.si(...;...)

Dans la cellule I3, on écrit =nb.si(\$F2:\$F1001;I2)

Dans la cellule I4, on écrit =I3/\$N3

Pour obtenir les sommes des lignes on utilise la fonction =somme(...:...) .

H	I	J	K	L	M	N
<b>Résultats possibles</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>Total</b>
<b>Effectifs</b>	59	252	362	260	67	1000
<b>Fréquences</b>	0,059	0,252	0,362	0,260	0,067	1

### Avec Python :

On construit une fonction qui a comme paramètre une liste de nombres contenus dans l'ensemble {0 ; 1 ; 2 ; 3 ; 4}. Cette fonction compte le nombre de fois où les valeurs 0, puis 1, puis 2, puis 3, puis 4 apparaissent dans une liste passée en argument. C'est l'occasion d'utiliser à nouveau count (...).

#Fonction Python qui crée l'affichage des effectifs des événements élémentaires

```
def effectif(liste):
    nb=0
    for i in range(5):
        nb=liste.count(i)
        print('Nombre de
familles à ',i, 'filles : ',nb)
```

# la fonction prend une liste en paramètre  
# on initialise un compteur à 0  
# pour i prenant les valeurs 0, 1, 2, 3, 4  
# on compte le nombre de i dans la liste liste  
# on affiche le nombre de famille a i filles

### Remarque importante :

La structure for i in range(5): permet de répéter 5 fois les instruction qu'elle contient mais attention, en fait, la variable i définie par cette instruction prend les valeurs 0, 1, 2, 3 et 4 car range(5) est la liste {0;1;2;3;4}.

#Fonction Python qui crée l'affichage des fréquences des événements élémentaires

```
def frequence(liste):
    long=len(liste)
    for i in range(5):
        nb=liste.count(i)
        freq=nb/long
        print('Fréquence de
familles à ',i, 'filles :
',freq)
```

# la fonction prend une liste en paramètre  
# long prend la valeur de la longueur de la liste  
# pour i prenant les valeurs 0, 1, 2, 3, 4  
# on compte le nombre de i dans la liste liste  
# on calcule la fréquence de l'occurrence de i  
# on affiche la fréquence de i dans la liste

## Etape 6 : Moyenne

Dernière étape, on souhaite calculer la moyenne de la série statistique obtenue dans l'étape 5. C'est à dire le nombre moyen de filles.

### Avec le tableur :

Quatre méthodes sont proposées :

Méthode 1 : on calcule la moyenne de la liste contenue dans la colonne F

$$=MOYENNE (F2 : F1001)$$

Méthode 2 : on calcule la moyenne de la liste contenue dans la colonne F

$$=SOMME (F2 : F1001) /N3$$

Méthode 3 : on utilise la formule de la moyenne

$$= (I3*I2+J3*J2+K3*K2+L3*L2+M3*M2) /N3$$

Méthode 4 : après avoir démontré le résultat, on utilise la formule utilisant les fréquences

$$=I4*I2+J4*J2+K4*K2+L4*L2+M4*M2$$

Cette dernière méthode permet d'introduire la formule de l'espérance mathématiques lorsqu'on travaille avec le tableau de la loi de probabilité de l'expérience aléatoire.

<b>Moyenne (Version 1) :</b>	<b>2,02</b>	A l'aide de la fonction moyenne
<b>Moyenne (Version 2) :</b>	<b>2,02</b>	A l'aide de la fonction somme
<b>Moyenne (Version 3) :</b>	<b>2,02</b>	A l'aide des effectifs
<b>Moyenne (Version 4) :</b>	<b>2,02</b>	A l'aide des fréquences

Dans les trois cas on obtient une valeur proche de 2.

### Avec Python :

On utilise la méthode 2. On crée une fonction qui prend une liste en paramètre (la liste des données dont on veut faire la moyenne) et qui renvoie la valeur de la moyenne

```
def moyenneFille(liste):
    somFille=0
    long=len(liste)
    for i in range(long):
        somFille=somFille+liste[i]
    moyenne=somFille/long
    return moyenne
```

# fonction qui prend une liste comme paramètre  
# on initialise une variable somFille à 0  
# on attribue à la variable long la longueur de la liste  
# on fait la somme des valeurs de la liste en utilisant la variable somFille  
# on calcule la moyenne  
# on retourne cette moyenne

## Bilan :

Avec le tableur, on peut travailler avec des échantillons de taille 1000 (c'est le cas dans notre exemple) ou de taille 5000 ou encore de taille 10000. Au delà, on constate que les copier-coller des formules sont de plus en plus pénibles et que les calculs se font moins rapidement.

Avec Python, on peut obtenir des simulations pour des échantillons de tailles  $10^6$  voir même de taille  $5 \times 10^6$ . Au delà les résultats mettent plus de temps à s'afficher mais travailler avec un échantillon de taille  $5 \times 10^6$  est déjà largement suffisant !

Avoir utilisé le tableur avant de programmer en Python a permis aux élèves de visualiser les objets qu'ils avaient à utiliser dans leur programme Python. Par exemple, dans l'étape 2, le tableur permet de voir le contenu de la variable listeEnfant de Python. Même chose dans l'étape 4, pour la liste listeNbFille de Python que l'on observe dans le tableur dans la colonne F. Dans l'étape 3, le tableur permet de comprendre ce que compte la fonction nbFille(...).

L'utilisation en parallèle du tableur et de Python permet aussi de mieux comprendre la structure répétitive bornée for comme le montre le tableau ci-dessous.

Dans l'étape 2 les lignes de code

```
for i in range(nbNaissance):  
    genre=naissance()  
    listeEnfant.append(genre)
```

se traduisent dans le tableur par un simple copier-coller

Dans l'étape 5 les lignes de code

```
for i in range(5):  
    nb=liste.count(i)  
    print('Nombre de familles à  
,i, 'filles : ',nb)
```

dans lesquelles la variable i est utilisée permettent de comprendre qu'il ne s'agit pas seulement de répéter 5 fois les instructions qu'elle contient mais que i prend les valeurs 0, 1, 2, 3 et 4.

Dans l'étape 6, les lignes de code

```
somFille=0  
for i in range(long):  
    somFille=somFille+liste[i]
```

peuvent être expliquée en demandant aux élèves de calculer de tête la somme des valeurs contenues dans la plage F2:F1001 ! Ils comprennent très vite le principe d'hérédité utilisé dans le code Python. Lorsqu'on en est à la n+1 ième ligne du tableau, on doit ajouter la valeur de la n+1 ième ligne à la somme des valeurs des n premières lignes.

Cette activité est l'occasion de faire constater aux élèves que :

- pour 2 échantillons de même taille la fréquence d'un événement, {« 1 Fille par famille »} par exemple, fluctue autour de 0,25
- plus la taille de l'échantillon augmente moins cette fluctuation est importante.
- si on fait tendre la taille de l'échantillon vers  $+\infty$ , alors la fréquence de l'évènement {« 1 Fille par famille »} tend vers 0,25 qui est la « fréquence théorique » ou la probabilité de l'évènement. Cette propriété est appelée la loi des grands nombres.

Après toutes ces simulations, on peut demander, dans un premier temps, aux élèves de conjecturer les probabilités de chaque événements élémentaires puis, dans un deuxième temps, de modéliser la situation à l'aide d'un arbre pondéré pour construire la loi de probabilité et ainsi vérifier leurs conjectures.

Le tableau de la loi de probabilité étant établi, on peut aussi donner du sens au nombre  $\sum_{i=1}^{i=5} i \times p_i$  (espérance mathématiques) en le rapprochant de la moyenne calculée à l'aide de la méthode 4 dans le tableur.