

Énoncé

Livre Barbazo Math Première Générale

TP

1

Une première marche aléatoire



Objectif
Simuler
une marche aléatoire.

Un kangourou se trouve en un point d'une route déserte d'Australie occidentale. À chaque seconde, il saute d'un mètre vers l'avant ou d'un mètre vers l'arrière avec la même probabilité.

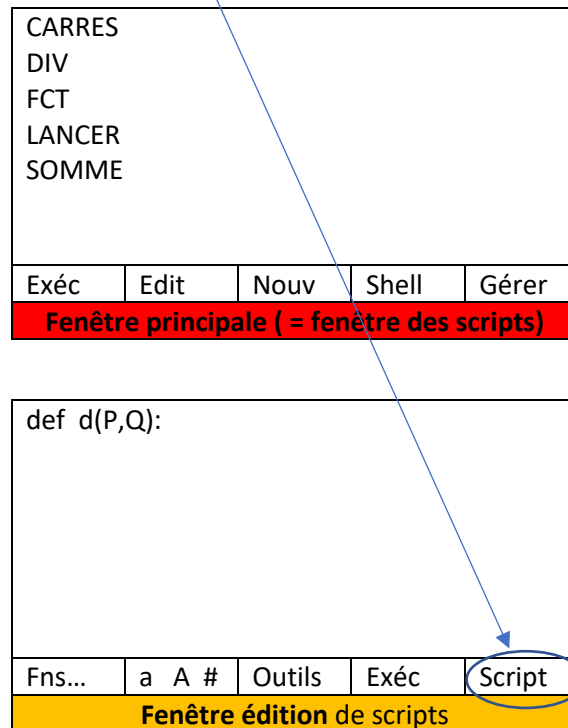
- 1 Recopier et compléter la fonction `mille_sauts` ci-dessous qui simule le déplacement du kangourou et qui renvoie la position du kangourou après avoir effectué mille sauts.

```
1 from random import random
2 def mille_sauts():
3     position=0
4     for i in ... :
5         if ... :
6             position=position+1
7         else:
8             position=position-1
9     return position
```

- 2 Exécuter la fonction `mille_sauts` plusieurs fois.
Quelle est la plus grande valeur renvoyée ?
- 3 En utilisant la fonction précédente, écrire le script d'une fonction `cent_metres` qui simule n expériences de mille sauts et qui renvoie le nombre de fois où le kangourou a avancé de plus de 100 mètres à la fin des mille sauts lors de ces n expériences.
- 4 Exécuter la fonction `cent_metres` pour $n = 10\,000$. Commenter.

TI83CE – Python – TP1 p286 « Une première marche aléatoire ».

La **fenêtre principale** est la **fenêtre des scripts** (c'est celle sur laquelle s'ouvre Python). Pour l'atteindre à tout moment, appuyer sur la touche F5 quand l'**onglet Script** est présent en bas de l'écran.



TP1 p286 :

Question 1

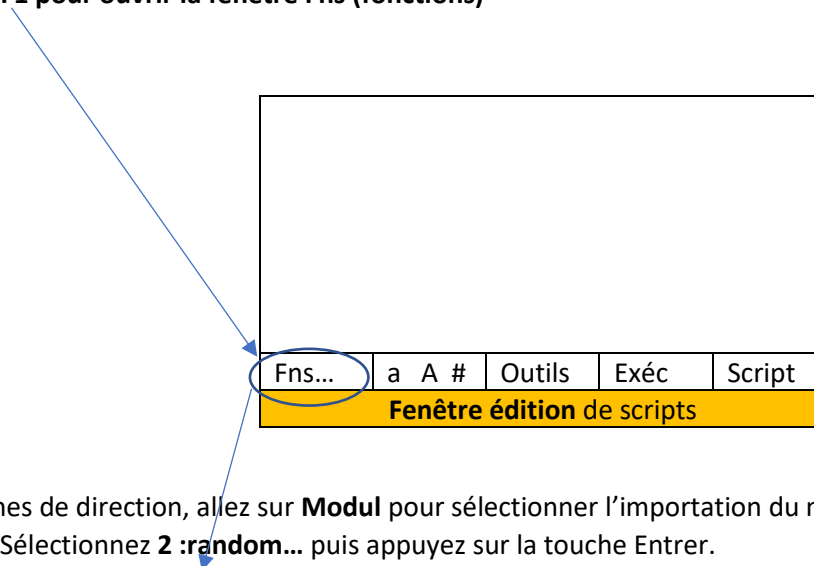
Depuis la fenêtre principale des scripts, appuyez sur la touche F3 "Nouv"

On entre un nom de script **TP1P286** (En majuscules seulement, ou chiffres, le premier caractère étant une lettre. Maximum 8 caractères) puis OK.

Nom= TP1P286	
Echap	Ok

La calculatrice passe automatiquement en fenêtre **édition de scripts**

Appuyer sur **F1** pour ouvrir la fenêtre **Fns (fonctions)**



Avec les flèches de direction, allez sur **Modul** pour sélectionner l'importation du module random nécessaire au programme. Sélectionnez **2 :random...** puis appuyez sur la touche Entrer.

Fonc	Ctl	Ops	List	Type	E/S	Modul
1: math...						
2: random...						
Echap						

On obtient :

Random
1:from random import *
2:random()
3:uniform(min,max)
4:randint(min,max)
5:choice(séquence)
6:randrange(début,fin,pas)
7:seed()
Modul

Choisissez **1:from random import *** puis appuyez sur la touche Entrer.

Cela importe toutes les fonctions du module random, dont celle que nous allons utiliser qui est randint()

Appuyez sur la touche Entrer.

Vous devriez obtenir l'écran :

from random import *				
Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarque : A ce niveau, il est possible de n'importer que la fonction que nous allons utiliser.

Ce qui donnerait **from random import randint** au lieu de from random import *

Appuyez sur **Fns**

La fenêtre Fns (fonctions) s'ouvre sur l'onglet **Fonc** déjà activé et sur l'option **1:def fonction():** déjà activée.

Fonc	Ctl	Ops	List	Type	E/S	Modul
1:def fonction():						
2: return						
Echap						

Appuyez sur la touche Entrer pour valider ce choix. On obtient :

from random import *				
def ():				
Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Le curseur clignote sur la première parenthèse de def () :

Donc tout ce qui est saisi va se placer juste *avant* la première parenthèse.

Mettez le clavier en mode alphabétique en appuyant sur 2nd alpha.

On voit que le clavier est en mode alphabétique minuscules par l'apparition de la lettre **a** en haut à droite de l'écran.

Saisissez le nom de la première fonction **millesauts** tout attaché – ne pas mettre le trait de soulignement **_** du 8 entre mille et sauts. Cela provoquerait une erreur sur les TI83 car il serait interprété comme un espace vide entre les deux mots.

Placez le curseur avec les flèches de direction à *la fin de la ligne* def millesauts() : et appuyez sur la touche Entrer.

On a alors l'écran :

```
from random import *
def millesauts():
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

On remarque que l'indentation Python (qui est de deux espaces sur les TI83 Python) se fait automatiquement.

Saisissez le code suivant. S'il se produit une erreur, la **touche suppr** permet de supprimer le caractère *avant* le curseur.

Vous devez obtenir le code Python suivant :

```
from random import *
def millesauts():
    position=0
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarques :

- Pour le signe = , le plus simple est de faire **2nd test**.
- N'oubliez pas de faire **2nd alpha** pour sortir du mode alphabétique et pouvoir saisir le chiffre 0.

Continuez jusqu'à obtenir :

```
from random import *
def millesauts():
    position=0
    for i in range(1000):
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarque : Pour for i in range, allez dans **Fns** puis **Ctl** (situé en haut de la fenêtre) puis choisissez for i in range(taille): puis appuyez sur la touche Entrer.

Continuez jusqu'à obtenir :

```
from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:

    else:
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarques :

Pour if .. else .., allez dans **Fns** puis **Ctrl** (les instructions de contrôle) puis choisissez if .. else .. puis appuyez sur la touche Entrer.

Pour randin(), allez dans **Fns Modul 2 :random...** et choisissez randint(min,max) et appuyez sur la touche Entrer.

Pour == qui teste l'égalité, allez dans 2nd test

La fonction randint(0,1) renvoie de façon équiprobable 1 (le kangourou avance d'un pas), et 0 (le kangourou recule d'un pas).

Continuez jusqu'à obtenir

```
from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
    else:
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Rappel : le signe = se trouve dans 2nd test

le retour au clavier en mode numérique se fait avec 2nd alpha. Il faut que la lettre **a** en haut à droite de l'écran ait disparu pour pouvoir saisir +1

On va utiliser un copier-coller pour obtenir :

```

from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
        else:
            position=position-1

```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Pour cela depuis l'écran :

```

from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
        else:

```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Placez le curseur au début de la ligne position=position+1 (la première lettre de position doit clignoter).

Allez dans l'onglet Outils (touche F3) sélectionnez 6 :Copier Ligne

Placez le curseur juste *après* les deux points de else :

Allez dans l'onglet Outils (touche F3) sélectionnez 7 :Coller Ligne

Vous devez obtenir

```

from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
        else:
            position=position+1

```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Il suffit de changer le + en - pour avoir :

```

from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
        else:
            position=position-1

```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarque : Avec « Outils », on peut supprimer une ligne directement.

Il faut placer le curseur sur la ligne à supprimer, aller dans **Outils**, choisir **5:Couper Ligne**

Terminez le code :

```

from random import *
def millesauts():
    position=0
    for i in range(1000):
        if randint(0,1)==1:
            position=position+1
        else:
            position=position-1
    return position

```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Remarque : pour avoir return, allez dans **Fns** et choisissez **2 :return**

Pour exécuter ce script, allez dans **Exéc** (touche F4)

La calculatrice passe en mode shell (ou mode console) ce qui est visible parce que les lignes commencent par 3

chevrons >>>

Le message

```

>>> # L'exécution de TP1P286
>>> from TP1P286 import *

```

indique que toutes les fonctions (en l'occurrence il n'y en a qu'une, c'est **millesauts()**) ont été importées du script TP1P286 dans la mémoire de travail de la calculatrice.

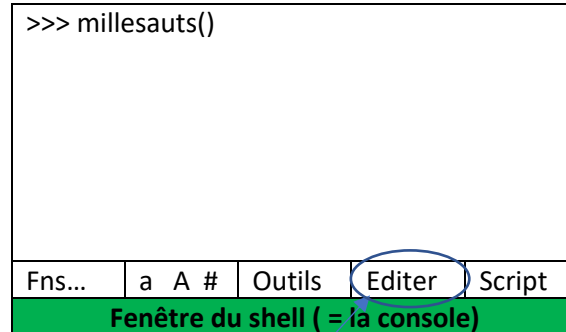
Remarque : si vous voulez nettoyer la console, allez dans **Outils** et choisissez **5:Effacer l'écran**

Appuyer sur la touche **var**

choisissez la fonction **millesauts()**

puis OK (touche F5)

L'écran suivant apparait :



Appuyez sur la touche Entrer pour exécuter la fonction millesauts()

Noter le nombre qui apparait : C'est l'abscisse où se trouve le kangourou après mille sauts d'un mètre en avant ou en arrière.

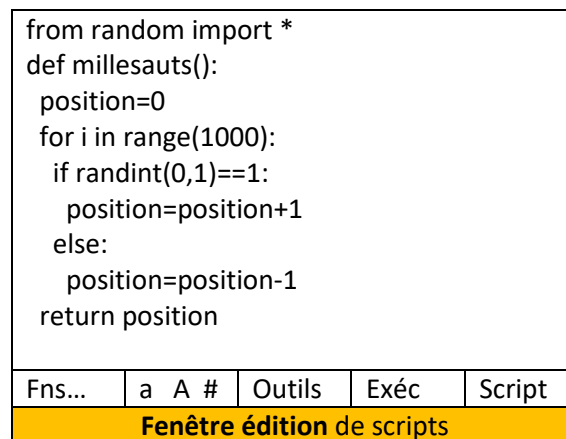
Pour exécuter d'autres fois la fonction millesauts(), appuyez sur la flèche de direction (vers le haut) pour rappeler la dernière commande entrée c'est-à-dire millesauts().

Question 3 :

Création d'une deuxième fonction **centmetres()** toujours dans le script TP1P286.

Pour cela appuyez sur Editer :

On est toujours dans le script TP1P286 :



Il suffit de placer à la suite de la première fonction millesauts la deuxième fonction :

Passez une ligne pour aérer le code et le rendre plus lisible.

Allez dans **Fns, Modul, math...**

Choisissez **1:from math import *** puis appuyez sur la touche Entrer.

Cela importe toutes les fonctions du module math, dont celle que nous allons utiliser qui est fabs()

Puis on définit la fonction centmetres(n) :

Cette fois, il y a un paramètre à passer qui est n le nombre de séries de 1000 sauts.

```
position=0
for i in range(1000):
    if randint(0,1)==1:
        position=position+1
    else:
        position=position-1
return position

from math import *
def centmetres(n) :
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

La fonction **millesauts** fait partie du même script TP1P286 (elle est écrite juste au-dessus), donc elle peut être « appelée » (c'est-à-dire utilisée) dans la fonction **centmetres**.

On commence par initialiser à 0 une variable **fois** qui comptera le nombre de fois où le kangourou aura avancé de plus de 100 mètres.

```
position=0
for i in range(1000):
    if randint(0,1)==1:
        position=position+1
    else:
        position=position-1
return position

from math import *
def centmetres(n) :
    fois=0
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

Ensuite on répète n fois l'expérience de 1000 sauts

```
for i in range(1000):
    if randint(0,1)==1:
        position=position+1
    else:
        position=position-1
return position

from math import *
def centmetres(n) :
    fois=0
    for i in range(n):
```

Fns...	a A #	Outils	Exéc	Script
Fenêtre édition de scripts				

On teste si la valeur absolue (puisque c'est une distance) de millesauts dépasse 100. La fonction valeur absolue est fabs(). Elle appartient au module math.

Si l'expression `fabs(millesauts())>100` est vraie , alors on incrémente la variable **fois** de 1. Sinon on ne fait rien.

Pour terminer, la fonction doit retourner la variable **fois**.

Ce qui donne le code :

```
else:
    position=position-1
return position

from math import *
def centmetres(n) :
    fois=0
    for i in range(n):
        if fabs(millesauts())>100:
            fois=fois+1
    return fois
```

Fns...	a A #	Outils	Exéc	Script
--------	-------	--------	------	--------

Fenêtre édition de scripts

Remarque : comme il est rare que le kangourou termine à plus de 100 mètres de sa position de départ (cela voudrait dire que sur 1000 sauts, la différence entre le nombre de pas vers l'avant et vers l'arrière dépasse 100), on peut commencer par `if fabs(millesauts())>50:` au lieu de `if fabs(millesauts())>100:`

Et on peut répéter l'expérience qu'un petit nombre de fois. Par exemple 30 fois.

Appuyez sur Exéc (la touche F4).

Appuyez sur la touche **var**. Choisissez la fonction `centmetres`.

Et exécutez la 30 fois :

`>>> centmetres(30)` et appuyez sur la touche Entrer.

Essayez plusieurs fois (Rappel : pour rappeler la dernière commande, appuyez sur la flèche de direction « vers le haut »).

Si le kangourou termine par exemple 3 ou 4 fois à plus de 50 mètres de son point de départ, vous pouvez considérer que le programme est correct.

Vous pouvez alors modifier le code en remettant `if fabs(millesauts())>100:` et laisser la calculatrice (assez longtemps) chercher si le kangourou peut se retrouver à plus de 100 mètres de son point de départ au cours de 10000 expériences (question 4 du TP).

La page suivante est une feuille de résumé des trois types de fenêtres Python présentes sur la calculatrice (scripts, édition, shell) et donne les circuits pour passer d'une fenêtre à l'autre.

