

## Contexte

On va utiliser l'environnement **Python** pour réaliser la simulation de 100 lancers d'un dé à six faces.

On va s'intéresser aux nombres de 6 obtenus à la fin de nos lancers.

## 1. Importer des bibliothèques

On crée un script **SIMUL** vierge et on l'exécute immédiatement en appuyant sur

**Exéc**

On appuie sur les touches **2nde** + **x<sup>2</sup>**. Cela fait apparaître l'instruction **sqrt()** que l'on complète avec le paramètre **49**, pour demander à obtenir  $\sqrt{49}$ . On obtient un message d'erreur.

La fonction n'est pas une fonction incluse par défaut en **Python**. Elle fait partie d'une bibliothèque, c'est-à-dire une sorte de module comportant un certain nombre de fonctions supplémentaires. **Sqrt()** fait partie de la bibliothèque **math**.

On retourne dans notre script (**Éditer**).

Puis **Fns...** et enfin onglet **Modul**. On commence par se rendre dans l'item **1:math...** et on sélectionne **1:from math import \***

On exécute de nouveau notre script et on répète la démarche précédente.

Cette fois-ci, on obtient comme réponse **7.0** à l'instruction **sqrt(49)**.

On retourne dans notre script et on importe, de la même manière, la bibliothèque **random** à partir, cette fois-ci, du module **2:random...**

## 2. Créer une liste

Une liste est un ensemble fini de valeurs que l'on souhaite stocker « au même endroit ». On utilise le crochet ouvrant **[** pour ouvrir notre liste et le crochet fermant **]** pour la refermer. Chaque élément est séparé par une virgule **,**. Le premier élément de la liste porte l'indice 0, le second 1 etc.

Exécutons et définissons une liste **l** à l'aide de l'instruction : **l = [7,8,9]**

**[** ou **]** : touche **2nde** + **x** ou **-** et = : touche **sto→**

Puis saisissons l'instruction **l[0]**. La console retourne 7, la première valeur (indice 0) contenue dans la liste **l** que nous avons créée.

Saisissons l'instruction **l[2]**. La console retourne 9, la troisième valeur (indice 2) contenue dans la liste **l**.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de SIMUL
>>> from SIMUL import *
>>> sqrt(49)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'sqrt' is not defined
>>> |
Fns... a A # Outils Éditer Script
  
```

```

ÉDITEUR : SIMUL
Fonc Ctl Ops List Type E/S Modul
1:math...
2:random...
3:time...
  
```

```

ÉDITEUR : SIMUL
math module
Math Const Trig
1:from math import *
2:fabs()
  
```

```

ÉDITEUR : SIMUL
LIGNE DU SCRIPT 0002
from math import *
  
```

```

ÉDITEUR : SIMUL
LIGNE DU SCRIPT 0003
from math import *
from random import *
  
```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de SIMUL
>>> from SIMUL import *
>>> l=[7,8,9]
>>> l[0]
7
>>> l[2]
9
>>> |
Fns... a A # Outils Éditer Script
  
```

### 3. Utiliser les méthodes pour les listes

On commence par saisir **l** le nom de notre liste puis retournons dans  et on se rend dans l'onglet **List**

De nombreuses méthodes sont disponibles pour manipuler les listes

Par exemple sélectionnons l'item **6**: **.append(x)**

On complète l'instruction pour obtenir **l.append(5)**. On valide. L'élément 5 a été ajouté à la liste **l**.

On reprend la manipulation pour utiliser l'item **A**: **.sort()**. On valide. La liste **l** a été triée dans l'ordre croissant.

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de SIMUL
>>> from SIMUL import *
>>> l=[7,8,9]
>>> l.append(5)
>>> l
[7, 8, 9, 5]
>>> l.sort()
>>> l
[5, 7, 8, 9]

```

```

PYTHON SHELL
Fonc Ctl Ops List Type E/S Modul
1:[ ]
2:list(séquence)
3:len()
4:max()
5:min()
6:.append(x)
7:

```

```

PYTHON SHELL
Fonc Ctl Ops List Type E/S Modul
3↑len()
4:max()
5:min()
6:.append(x)
7:.remove(x)
8:.insert(indice,x)
9:sum()
0:sorted()
A.sort()
B:.count()
Échap

```

### 4. Réaliser une simulation de lancer de dés

On complète le script **SIMUL** avec la fonction **simulation** qui prend le nombre de lancers **n** en paramètre et renvoie la liste **l** contenant chacun des résultats de lancer du dé.

On utilise la fonction **randint** de la bibliothèque **random**. Elle renvoie un nombre entier aléatoire compris entre les deux nombres entiers passés en paramètre.

Avec l'instruction **randint(1,6)**, on simule le lancer d'un dé à six faces.

A l'aide de la structure **for**, indentée correctement et l'utilisation notamment de l'instruction **range(n)**, on répète les lancers **n** fois.

Chaque résultat renvoyé par la fonction **randint(1,6)** est ajouté dans la liste **l** à l'aide de l'instruction **l.append(...)**

Il n'y a plus qu'à tester notre fonction dans la console.

Pour cela on exécute

On saisit l'instruction **l=simulation(100)** pour sauvegarder le résultat dans une liste **l**. La touche  permet de sélectionner le nom de la fonction sans avoir à le saisir. C'est un gain de temps. On valide.

On saisit **l** et on valide pour obtenir l'affichage du contenu de la liste **l**.

On compte le nombre de 6 obtenus à l'aide de l'instruction **l.count(6)** (item **B**: **.count()** dans l'onglet **List** via )

Dans notre simulation, nous avons obtenu 18 fois la face 6.

```

ÉDITEUR : SIMUL
LIGNE DU SCRIPT 0001
from math import *
from random import *

def simulation(n):
    l=[]
    for i in range(n):
        l.append(randint(1,6))
    return l

```

```

PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de SIMUL
>>> from SIMUL import *
>>> l=simulation(100)

```

```

PYTHON SHELL
4, 6, 1, 6, 1, 2, 6, 1, 2, 5, 2,
4, 1, 5, 1, 1, 1, 3, 1, 2, 6, 1,
, 4, 2, 1, 2, 4, 1, 3, 3, 5, 4,
5, 2, 4, 6, 6, 1, 3, 5, 6, 1, 2,
5, 6, 5, 1, 5, 6, 3, 1, 6, 2, 6
, 3, 2, 4, 6, 2, 2, 4, 2, 4, 4,
1, 1, 2, 5, 4, 2, 1, 6, 3, 1, 1,
1, 6, 1, 1]
>>> l.count(6)
18

```