2 Mémo Python

Les modules (math et random)

Les modules - Importation

Toutes les fonctions et commandes ne sont pas présentes dans la version de base de python. Il faut donc faire appel à des modules spécifiques. Il en existe un très grand nombre et en voici quelques uns.

Modules	Description
math	*Fonctions, méthodes et constantes mathématiques pour les nombres non complexes.*
random	*Générateurs de nombres pseudo-aléatoires.*
mathplotlib.pyplot	*Permet de générer un environnement graphique avec repère gradué.*
turtle	*Permet de dessiner des formes diverses à l'aide d'une tortue qui se déplace.*
pylab	*Regroupe les bibliothèques **numPy** et **matplotlib***
scipy.stats	*Probabilités et fonctions statistiques*
tkinter	*Création d'interfaces graphiques d'interactions avec l'utilisateur.*

Pour faire appel à un module, il existe 2 méthodes :

<table>
<tr><th colspan="2">Méthode 1</th><th colspan="2">Méthode 2</th></tr>
<tr><td colspan="2">import nom_du_module
Charge le module Python dans son propre espace de noms, de sorte que vous devez ajouter le nom du module suivi d'un point devant les fonctions de tous les modules importés.</td><td colspan="2">from nom_du_module import*
Charge le module Python dans l'espace de noms actuels, de sorte que vous puissiez y faire référence sans avoir à mentionner à nouveau le nom du module. Inconvénient : Lors de nombreux imports, il peut y avoir confusion ou conflit entre différentes fonctions.</td></tr>
<tr><th colspan="2">Exemples</th><th colspan="2">Exemples</th></tr>
<tr><td>import math
b = math.sin(a)</td><td>import random
b = random.randint(1, 6)</td><td>from math import*
b = sin(a)</td><td>from random import*
b = randint(1, 6)</td></tr>
<tr><td colspan="2">Remarque : Si le nom du module est long, on peut le remplacer par un nom plus court.

Exemple :
import random as rd
b = rd.randint(1, 6)</td><td colspan="2">Remarque : * permet de charger l'ensemble du module mais on peut ne charger qu'une fonction.

Exemple :
from random import randint
b = randint(1, 6)</td></tr>
</table>

Le module math

Le module **math** permet d'accéder aux fonctions et constantes mathématiques de bases, pour les nombres entiers, décimaux, rationnels et réels. Ci-dessous quelques fonctions.

En tête de programme : **from** math **import*** (Si méthode **import math**, il faut saisir **math.*fonction***)

Fonctions	Description	Fonctions	Description
ceil()	*Retourne l'entier le plus proche par excès du nombre décimal ou réel indiqué.*	**log()**	*Retourne le logarithme naturel d'un nombre (base e).*
cos()	*Retourne le cosinus d'un angle exprimé en radians.*	**log10()**	*Retourne le logarithme d'un nombre en base 10.*
degrees()	*Convertit des radians en degrés.*	**modf()**	*Retourne la partie décimale et la partie entière d'un nombre.*
exp()	*Retourne l'exponentielle d'un nombre.*	**pow()**	*Retourne un nombre élevé à une certaine puissance.*
fabs()	*Retourne la valeur absolue d'un nombre.*	**radians()**	*Convertit des degrés en radians.*

factorial()	*Retourne la factorielle d'un nombre entier supérieur à 0.*	**sin()**	*Retourne le sinus d'un angle exprimé en radians.*
floor()	*Retourne l'entier le plus proche par défaut du nombre décimal ou réel indiqué.*	**sqrt()**	*Retourne la racine carrée d'un nombre.*
fmod()	*Retourne le modulo de deux nombres.*	**tan()**	*Retourne la tangente d'un angle exprimé en radians.*

Constantes	Valeurs
pi	*3.141592654 ...*
e	*2.718281828 ...*

Exemple : Calculs du sinus, cosinus et tangente d'un angle donné en degrés

```
from math import*

ad = float(input("Saisir un angle en degrés : "))
ar = radians(ad)  # Conversions degrés en radians
print("Le sinus de l'angle ", ad, " degrés est : ", round(sin(ar), 3))
print("Le cosinus de l'angle ", ad, " degrés est : ", round(cos(ar), 3))
print("La tangente de l'angle ", ad, " degrés est : ", round(tan(ar), 3))
```

Le module random

Le module **random** permet d'accéder à des fonctions permettant de générer des nombres aléatoires, des listes aléatoires, ... Ci-dessous quelques fonctions.

En tête de programme **from** random **import*** (Si méthode **import random**, il faut saisir **random.*fonction***)

Fonctions	Description
random()	*Retourne un nombre décimal aléatoire compris entre 0 et 1*
randint(*a, b*)	*Retourne un nombre entier aléatoire dans la plage précisée.*
uniform (*a, b*)	*Retourne un nombre décimal aléatoire dans la plage précisée.*
randrange(a, b, c)	*Retourne un nombre entier entre a et b avec un pas de c*
choice([*liste_valeurs*])	*Retourne une valeur aléatoirement parmi une collection de valeurs fournie.*
choices([*liste_valeurs*], weights=[*liste_coeff*], k=*valeur*)	*Retourne une liste de valeurs aléatoirement avec remise parmi la liste avec un poids (**weights**) défini pour chaque élément. **k** : nombre de valeurs à retourner.*
sample([*liste_valeurs*], weights=[*liste_coeff*], k=*valeur*)	*Même fonctionnement que **choices** mais sans remise. (k doit être inférieur au nombre d'éléments de la liste)*
shuffle([*liste_valeurs*])	*Mélange aléatoirement les éléments d'une liste*

Exemple : n lancers d'un dé à 6 faces et calcul de la fréquence d'obtention de la face "6"

```
from random import*
c = 0
n = int(input("Combien de lancers ? "))
for i in range(n):
    de = randint(1, 6)
    print(de, end=" ")
    if de == 6:
        c = c+1
print()
print("Fréquence d'ontention de la face 6 : ", round(c/n, 3))
```