

Lisibilité des programmes

Faites une copie avec la commande ci dessus "File>Make a Copy...", et renommez la en cliquant sur le titre (à côté de jupyter TD3...), par exemple en rajoutant votre nom.

Les bibliothèques

Une bibliothèque est un ensemble de fonctions préprogrammées, que l'on incorpore en début de programme.

On utilisera principalement les bibliothèques mathématiques (math) et aléatoire (random). Avant d'utiliser une bibliothèque, autre que les deux précédentes, posez-vous les questions suivantes:

- en ai-je besoin ? (9 fois sur 10 la réponse est non);
- cela va-t-il me simplifier le travail ? (la réponse est souvent non, ça sera bien plus compliqué, mais parfois oui, ça sera beaucoup plus simple...);
- la bibliothèque est-elle dépréciée ? (c'est à dire "périmée mais quand même vaguement comestible à vos risques et périls, et surtout n'attendez aucune aide du prof, ni de personne d'ailleurs").

Jusqu'à ce que vous ayez entamé vos projets, vous perdrez du temps à chercher des solutions toutes faites sur le web, et vous apprenez moins bien : je vous déconseille fortement d'utiliser le web comme autre chose qu'une source de documentation. Il est vrai que ça rassure et que ça semble plus simple de chercher sur internet, mais au début c'est une perte de temps pour ce qui est de la programmation proprement dite.

```
In [ ]: from math import *
        print(pi)
        print(cos(pi))
        print(sqrt(3))           # fonction racine carré (square root = sqrt)
        t)
```

```
In [ ]: from random import *
        for i in range(10):
            print(random())
```

```
In [ ]: for i in range(10):
        print(randint(3,8))
```

```
In [ ]: for i in range(10):
        print(randrange(3,8))
```

```
In [ ]: for i in range(10):
        print(randrange(3,8,2))
```

Commentaires à compléter :

- lorsqu'une bibliothèque a été importée, on n'a pas besoin de la réimporter (deux dernières boucles):
- différence entre randint et randrange :

Les fonctions

Un programme devient très vite illisible, c'est pourquoi on le découpe en fonctions. Lors de la réflexion préalable et indispensable avant de programmer, on écrit un algorithme sous forme de fonctions. On peut également, lors de cette réflexion, utiliser la syntaxe "fonction magique" : c'est un problème que l'on laisse en suspens pour le moment.

Exemple:

Début

Fonction $\text{epd}(a,b)$

"résolution d'une équation du type $ax + b = 0$ "

*paramètres a et b deux nombres réels quelconques, a différent de 0
renvoie la solution de l'équation $ax + b = 0$
renvoyer $(-b/a)$*

Fonction $\text{eln}(a,b)$

fonction magique "résolution d'une équation du type $a \cdot \ln(x) = b$ "

*paramètres et retour : on verra ça plus tard
renvoie (1) valeur choisie au hasard*

Programme principal

$sol \leftarrow \text{epd}(3,2)$

Afficher "la solution de l'équation $3x + 2 = 0$ " vaut sol

$sol \leftarrow \text{eln}(3,2)$

Afficher "la solution de l'équation $3\ln(x) = 2$ " vaut sol

Fin

La traduction en Python :

```
In [ ]: #####
#
# Bibliothèques
#
#####
from math import *           # on en aura probablement besoin pour
le logarithme neperien

#####
#
# Fonctions
#
#####

def epd(a,b):
    """
    Resolution de l'equation  $ax + b = 0$ 
    @param : a reel non nul
    @param : b reel
    @return : c réel
    """
    c = -b/a
    return (c)

def eln(a,b):
    """
    Resolution de l'equation  $a.\ln(x) = b$  a faire ulterieuremnt
    @param : a ?
    @param : b ?
    @return : c ?
    """
    return (1)

#####
#
# Programme principal
#
#####
sol = epd(3,2)
print("la solution de l'equation  $3x + 2 = 0$  vaut",sol)
#ci-dessous avec appel de la fonction dans le print
print("la solution de l'equation  $3.\ln(x) = 2$  vaut",eln(3,2))
```

Remarquez la précision des commentaires sur la nature des arguments : un utilisateur qui essaierait de résoudre $0 \cdot x + 2 = 0$ ne pourrait pas se plaindre de dysfonctionnement, puisqu'il est précisé qu'il ne faut pas le faire ($a \neq 0$).

En théorie, on fait des fonctions distinctes pour les entrées, le calcul et les sorties. Le programme principal se contente d'organiser les différentes fonctions.

Exercice :

Structurer en fonctions le programme de résolution de l'équation du second degré ci-dessous. Aucun calcul ne doit être fait dans le programme principal, seulement les entrées-sorties, et l'organisation entre les différentes fonctions

```
In [ ]: #####
#
# Bibliothèques
#
#####
from math import *

#####
#
# Fonctions
#
#####

#####
#
# Programme principal
#
#####
print("Résolution de l'équation du second degré ax^2 + bx + c = 0")
a = 0
while a == 0:
    a = int(input("donnez le coefficient a différent de 0:"))
b = int(input("donnez le coefficient b:"))
c = int(input("donnez le coefficient c:"))
delta = b**2 - 4*a*c
print("delta = ",delta)
if delta < 0:
    print("l'équation n'a pas de solution réelle")
elif delta == 0:
    print("l'équation admet une solution double : ",-b/(2*a))
else :
    print("l'équation admet deux solutions réelles : ",(-b + sqrt(delta))/(2*a)," et ",(-b - sqrt(delta))/(2*a))
```

[![Licence CC BY NC SA](https://licensebuttons.net/l/by-nc-sa/3.0/88x31.png "licence Creative Commons CC BY SA")](http://creativecommons.org/licenses/by-nc-sa/3.0/fr/)

Frederic Mandon (mailto:frederic.mandon@ac-montpellier.fr), Lycée Jean Jaurès - Saint Clément de Rivière - France (2015 - 2017)