Informatique - TP 2 **Fonctions et boucles** 25 - 29 septembre 2023

Corrigé

Code de partage avec Capytale : c1c5-1915054

L'ajout de bibliothèques

On importera très souvent la bibliothèque *numpy* pour « numbers Python » qui permet de développer l'aspect calculatrice de Python et, entre autres, d'utiliser de nombreuses fonctions usuelles.
Pour la représentation graphique, on importera la bibliothèque *matplotlib.pyplot*
Par commodité, on renomme généralement ces bibliothèques avec un « alias » : par exemple *np* pour *numpy*

On pourra donc entrer (dans l'éditeur ou la console) :

import numpy as np
import matplotlib.pyplot as plt

Echauffement

Tester les commandes suivantes (à taper dans la console) :

```
a=np.exp(10)
b=np.log(a)
b
c=np.e
d=np.log(e) # donne un message d'erreur car e "tout seul" n'est pas défini
d=np.log(np.e)
np.sqrt(16)
np.sqrt(4)
np.sqrt(-3)
np.abs(1)
np.abs(-3)
np.abs(-2.5)
np.abs(7)
np.floor(2,1)
np.floor(-3.4)
np.floor(7.8)
np.pi
```

1 Exercices

1.1 Fonctions

Syntaxe pour l'écriture d'une fonction

```
def carre(x): # on nomme la fonction et entre parenthèse on précise la (ou les)
    variable(s) d'entrée
  return x**2 # on donne le résultat qui sort de la fonction
```

puis dans la console, on peut utiliser la fonction, par exemple avec `> carre(23)`

Exercice 1 - polynôme

Définir la fonction du second degré $f(x) = x^2 - 2x - 99$ puis, en tatônnant, essayer d'en trouver les racines à l'aide de Python.

De manière analogue, on définit la fonction (cf. ci-dessous), puis en testant des valeurs « manuellement », on remarque que -9 et 11 sont les racines de ce trinôme (en tapant dans la console `f(1)` ...).

```
def f(x):
  return x**2-2*x-99
```

Exercice 2 - moyenne

Définir une fonction qui renvoie la moyenne de deux nombres réels.

Cette fois, la fonction prend en entrée deux objets (deux nombres en l'occurence) et renvoie simplement la somme de ces deux nombres divisés par deux. On peut tester la aussi dans la console : `moyenne(50, 60)` par exemple, en respectant bien la spécificité de cette fonction (deux nombres en entrée).

```
def moyenne(a,b):
  return (a+b)/2
```

Exercice 3 - condition

Ecrire un programme qui définit la fonction valeur absolue.

Il faut cette fois insérer la condition propre à cette fonction mathématique dans la définition de cette fonction dans Python. Le `else` regroupe tous les cas non mentionnés précédemment (on n'a donc pas besoin de donner de condition à cet endroit). On peut ensuite tester dans la console : `vabs(15)` ou encore `vabs(-57)`...

```
def vabs(x):
  if x>= 0:
    return x
  else :
    return -x
```

1.2 Boucles

Les boucles

Les boucles sont des outils fondamentaux en algorithimque et nous utiliserons deux types de boucles : la boucle *for* (ci-dessous) et la boucle *while* (plus tard).

La boucle *for* permet d'exécuter une instruction un nombre de fois prédéfini. Cela peut être une simple répétition (question 1 ci-dessous), ou l'instruction peut dépendre de la variable (question 2 ci-dessous).

Exercice 4 - deux boucles for

1. Faire afficher 100 fois « il fait chaud ! »

   On utilise ici la boucle `for` dans une optique de répétition. Attention, le deuxième nombre du `range` est exclu, donc pour avoir 10 occurences, il faut $(0, 10)$ dans la gamme de valeurs ou $(1, 11)$ ou ...

   ```
   for i in range(0,100) :
       print('il fait chaud')
   ```

   On note les : et l'indentation qui marquent l'énoncé de l'instruction à exécuter à chaque passage dans la boucle.

2. Calculer les cubes des 100 premiers entiers naturels.

   Même remarque pour la gamme de valeurs, cette fois de 1 à 101 (ou 0 à 99 si on commence par 0 qui est en fait le premier entier naturel).

   ```
   for i in range(1,101):
     print(i**3)
   ```

   ou pour s'amuser

   ```
   for i in range(1,101):
       print('le cube de', i, 'est', i**3)
   ```