LES FACULTÉS
DE L'UNIVERSITÉ
CATHOLIQUE DE LILLE

TP N°2 :

Programmation en Python

Les structures de contrôle

SOMMAIRE

I. RAPPEL RESSOURCES

1. De l'enseignant

- Mémento Python
- Rappel de cours niveau 1 Python

2. Livre en accès libre

- Apprendre à programmer avec Python 3 (disponible gratuitement https://inforef.be/swi/download/apprendre_python3_5.pdf)

3. Editeur Python

- En ligne : https://replit.com/languages/python3
- Méthode privilégiée : un éditeur Python très simple d'utilisation mu-editor (mac et PC) :https://codewith.mu/en/download

II. OBJECTIF DU TP

Le but de ce troisième TD est une révision des boucles en langage Python.

Le degré de difficulté est indiqué par des étoiles à la fin du titre de l'exercice.

III. RAPPEL : LES STRUCTURES DE CONTRÔLE

1. La boucle for pour répéter un nombre de fois

Cette boucle est utilisée lorsqu'on connait le nombre d'itérations avant même de l'appeler c'est-à-dire :
La boucle pour est utilisée lorsque l'on connait le nombre de fois que la boucle va être parcourue

Par exemple au lieu d'écrire :

```
print('bonjour') # ceci est un commentaire 1
print('bonjour') # 2
print('bonjour') # 3
print('bonjour') # 4
print('bonjour') # 5
print('bonjour') # 6
print('bonjour') # 7
print('bonjour') # 8
print('bonjour') # 9
print('bonjour') # 10
```

Il est possible d'utiliser la boucle *for* :

Python
```
for  i in range(0,10):
    print('bonjour') #tabulation au début
```

Comme pour le if, python utilise **l'indentation** qui est soit une tabulation, soit quatre espaces.

La fonction range()

La fonction *range* permet de créer une liste de nombres compris entre un nombre de départ (inclus) et un nombre de fin (exclus).

Range() a trois paramètres : début, fin et pas, exemple :

```
for i in range(10): # de 0 à 9 (10 non inclus)
for i in range(1,10): # de 1 à 9
for i in range(0,100,5): # de 0 à 95 par pas de 5
for i in range(0,-10,-3): # de 0 à -9 par pas de -3
```

Par exemple pour afficher 10 fois bonjour :

```
for  i in range(0,10):
    print('bonjour') # indentation au début de ligne
```

Exemple simple

```
for i in range(1,11):
    print("i=",i)
```

Boucles imbriquées :

```
for i in range(101):
  for j in range (10):
    print(j)
    if (j%2 == 0) :
      print('Pair') # dans le if
    print(i*j) # dans le 2 for
  print(i) # dans le 1er
print('Fin du programme') # à la fin
```

2. While pour répéter en respectant une condition

Cette boucle est utilisée quand on ne connait pas le nombre de fois que la boucle doit être itérée.

Il ne faut pas oublier les deux points « : » à la fin de la ligne et l'indentation (tabulation ou espace).

```
x=45
y=55
while x<50 and y <70:
    x=x+1
    y=y+1
    print(x,y)
```

Exemples

Tant que la somme <100

```
somme = 0
while somme<100:
    somme = 2*somme + 1
print('somme=',somme)
```

A la place d'une boucle for

Python

```
for i in range(1,11):
    print("i =",i)
```

```
i = 1
while ( i <10 ):
  print("i =",i)
  i = i + 1
```

IV. RAPPEL : COMMENTAIRES ET MODULES

1. Les commentaires

Un code bien écrit doit être facilement compréhensif : en le lisant on doit comprendre ce que vous avez fait. Pour cela, il faut au moins utiliser des noms de variables qui indiquent leurs fonctions.Si cela n'est pas suffisant, les commentaires servent à préciser votre code.

L'explication sur un ligne débute par « # » :

```
# Sur une ligne
```

Et sur plusieurs lignes, encadré par « """ »

```
""" Sur plusieurs
lignes   """
```

2. Les modules

Par exemple pour importer la fonction sqrt (square root = racine carrée) :

```
from math import sqrt
print(sqrt(16))
```

Il est possible d'en importer plusieurs à la fois :

```
from math import sqrt, pi, cos
print(cos(pi/2))
```

Et même toutes les fonctions du module math :

```
import math
print(math.sqrt(16)) # Affiche 4
```

Un autre exemple où en plus d'importer, on a défini un synonyme (PI en majuscule au lieu de pi)

```
from math import pi as PI
print(PI)
```

Et même renommer un module :

```
import numpy as np
import matplotlib.pyplot as plt
```

V. EXERCICES

1. Les boucles

Voir cours : V.Les structures de contrôle

Exercice N°1. Afficher 1000 fois "Bonjour"1st *

Afficher 1000 fois "Bonjour"

```
for i in range(1000):
        print('Bonjour')
```

Exercice N°2. Tant que >10 *

Demander à l'utilisateur un nombre tant que ce dernier est >10 avec la boucle tant que

```
nb = 11
while nb >10:
        nb = int(input('nb ?'))
```

Exercice N°3. Moyenne des températures de la semaine *

Écrire un programme qui demande les températures de 7 jours, calcule puis affiche la température moyenne de la semaine.

somme = 0

```
for i in range(7):
      temperature = float(input('température N°'))
      somme = somme + temperature
print ('la moyenne est', somme/7)
```

Exercice N°4. Somme des 100 premiers entiers *

Écrire un programme qui fait la somme des 100 premiers entiers.
Faites deux versions l'une avec la boucle *pour* et l'autre avec *tant que*

```
somme = 0
for i in range(1001):
        somme = somme + i
print ('la somme est', somme)

somme = 0
i = 0
while i <=1000:
        somme = somme + i
        i +=1
print ('la somme est', somme)
```

Exercice N°5. Factorielle **

Écrire un algorithme qui demande un nombre de départ et qui calcule sa factorielle (exemple 5 !=1*2*3*4*5)

```
n = int (input('n?'))
produit = 1
for i in range(1,n):
        produit = produit * i
print(n,'!',produit)
```

Exercice N°6. Nombre mystérieux **

Soit un nombre mystérieux que vous initialiserez avec une valeur que vous choisissez, écrire un algorithme qui demande un nombre compris entre 1 et 20, jusqu'à ce que l'utilisateur trouve le nombre mystérieux.
En cas de réponse supérieure, on fera apparaitre un message : Plus petit ! , et inversement, Plus grand ! si le nombre est inférieure.
Pour finir, si le nombre est trouvé, « Gagné » s'affichera et le programme s'arrêtera.

```
from random import randint
mystere = randint(1,20)
saisie = mystere + 1
```

```
while saisie != mystere:
        saisie = int(input('un nombre ? '))
        if saisie > mystere:
                print('trop grand')
        elif saisie < mystere:
                print('trop petit')
        else: print('gagné')
```

Exercice N°7. Imbriquer deux boucles heures : minutes de la journée ***

Écrire un programme qui affiche l'heure : minutes de la journée. On débute à minuit 00 :00, 00 :01,...,10 :32,…15 :59 jusqu'à 23 :59
Il faut utiliser deux boucles imbriquées : une première pour les heures qui englobera celle des minutes

```
for heure in range(0,60):
        for minute in range(0,60):
                if heure<10:
                        heureStr = '0'+ str(heure)
                else:
                        heureStr = str(heure)
                if minute<10 :
                        minuteStr = '0' + str(minute)
                else:
                        minuteStr = str(minute)

                print(heureStr+':'+minuteStr)
```