

TP N°2: Les structures conditionnelles

Principe et syntaxe en python

Une instruction conditionnelle permet d'exécuter des instructions seulement sous une certaine condition. En python, on utilise les mots clefs `if`, `else` et `elif` qui consistent à tester si une affirmation est vraie ou non. L'instruction `if` permet d'exécuter un programme P1 ou un autre P2 selon qu'une condition C est vraie ou non. La syntaxe est la suivante :

```
1  if C:
2      P1
3  else:
4      P2
```

Attention à bien indenter les instructions des programmes P1 et P2 . Indenter signifie décaler vers la droite de *n* (généralement 4) espaces. Ce n'est pas décoratif, mais cela a un sens pour python, car l'indentation permet de regrouper des instructions par blocs. Un bloc est un groupe d'instructions consécutives ayant la même indentation.

Exemple: Le programme suivant met le maximum de *a* et *b* dans la variable *z* et l'imprime à l'écran.

```
1  a = int(input("a = "))
2  b = int(input("b = "))
3  if a < b:
4      z = b
5  else:
6      z = a
7  print(z)
```

Rappel: l'instruction `r = input(s)` affiche la chaîne *s*, attend que l'utilisateur tape une chaîne de caractères au clavier terminée par la touche Entrée, puis met cette chaîne de caractères dans la variable *r*.

Remarque: On peut imbriquer les `if` (et donc ajouter les indentations des blocs), par exemple pour calculer le maximum *m* de *a*, *b* et *c* :

```
1  a = int(input("a = "))
2  b = int(input("b = "))
3  c = int(input("c = "))
4  if a < b:
5      if b < c :
6          m = c
7      else:
8          m = b
9  else: # ici on est dans le cas a >= b
10     if a < c:
11         m = c
12     else:
13         m = a
14  print(m)
```

Remarque: En Python, la structures conditionnelle `if` possède une forme généralisée dont chaque `else: if condition:` est remplacée par `elif condition:` . Par exemple :

```
1  if a < b:
2      if b < c :
3          m = c
4      else:
5          m = b
6  elif a < c: # ici on est dans le cas b <= a < c
7      m = c
8  else: # ici on est dans le cas b <= c <= a
9      m = a
10  print(m)
```

Remarques

- Une condition est une expression logique de type booléen (dont son évaluation retourne soit True soit False). Tous les nombres différents de zéro et toutes les structures de données non vides sont évalués à True.
- Dans le cas où une condition est vérifiée et on ne souhaite rien faire, on utilisera le mot clé `pass`. Cette instruction est certes inutile mais sachez que vous êtes obligé de mettre au moins une instruction après une condition (après chaque utilisation de deux points :).
- Pour enchaîner les tests, on peut remplacer `if a < b and b < c :` par `if a < b < c :`
- Il est possible que dans votre code vous ayez besoin de faire une condition dans une condition. Dans ce cas, il faut faire très attention aux indentations (décalage) : si on les modifie le retour du programme peut être différent.

Exercices

Pour écrire et sauvegarder des programmes, la procédure générale est la suivante :

1. Pour créer un nouveau programme (script) Python, utiliser le menu file / New File.
2. Enregistrer-sous votre script sous le dossier D:/votre_groupe/nom_prénom/num_tp/num_exercice.py
Exemple : D:/M1/Ali BEN Ahmed/tp2/ex1.py
Remarque : lorsque vous saisissez le nom du fichier «ex1.py» , n'oubliez pas de spécifier l'extension du fichier «.py», cela permet à l'éditeur IDLE de mieux visualiser votre code source.
3. Pour lancer l'exécution de votre programme, utiliser le menu Run / Run Module (F5).

Attention Il n'est pas possible d'ouvrir un fichier dans IDLE en double-cliquant sur ce fichier depuis l'explorateur de fichier. Pour ouvrir un script, il faut toujours lancer IDLE et utiliser la commande « Ouvrir... » du menu Fichier.

Il faudra bien sûr enregistrer votre script très régulièrement pour ne pas tout perdre en cas de problème informatique.

Exercice 1

1. Créer un fichier intitulé `test_poids`. Définir deux variables de type flottants `taille` et `poids` affectées avec votre taille et votre poids.
2. L'indice de masse corporelle (abrégié IMC, égal à $\frac{\text{poids}}{\text{taille}^2}$) d'un individu donne une indication sur sa santé. Écrire un script qui calcule l'IMC que vous stockerez dans une variable `imc`, et indiquera par un message à l'écran si l'individu est en surpoids ($\text{IMC} > 25$) ou en sous-poids ($\text{IMC} < 18$).
3. Tester pour 3 individus : 1,80 m et 50 kg, puis 1,50 m et 100 kg, puis 1,75 m et 70 kg.

Exercice 2: Équation du second degré

On considère une équation du second degré de la forme $ax^2 + bx + c = 0$, dont les trois coefficients seront supposés être des entiers.

1. Écrire un script qui, en fonction des valeurs attribuées à a , b et c , indique automatiquement *le nombre de solutions* de l'équation ainsi que *leurs valeurs numériques*. On travaillera dans le corps des complexes, ce qui veut dire que si le discriminant est négatif on donnera des solutions complexes.
2. On initialisera les valeurs dans le script par l'affectation des trois coefficients. Par exemple :

```
1 a,b,c = 1,2,-3
2 [ suite du script ]
```
3. On testera le programme avec $(a,b,c) = (1,2,3)$ puis $(a,b,c) = (1,2,1)$ et enfin $(a,b,c) = (2,2,1)$.