

TP Python 2 : conditions

L.-C. LEFÈVRE

I Les comparaisons

On peut utiliser les opérations suivantes pour **comparer** deux nombres :

- plus petit <, plus grand >
- plus petit ou égal <=, plus grand ou égal >=
- égal ==, différent !=

La valeur d'une telle comparaison est, pour le langage Python, soit **True** (vrai), soit **False** (faux). Par exemple, essayer chacune des lignes suivantes :

```
print(3 < 4)
print(3 > 4)
print(5 + 15 == 30)
print(25 + 1 >= 26)
print(1 != 2)
```

Question 1. Donner la valeur de 2^{100} et de 3^{63} . Quel est le plus grand des deux ? Utiliser la comparaison `2 ** 100 < 3 ** 63`

II Les conditions

On veut écrire un morceau de programme qui s'exécute **si** une condition est vraie. Dans le langage Python cela se traduit par le mot-clé **if** suivi de la condition. La partie du programme qui est exécutée sous la condition est ensuite décalée vers la droite (on dit **indentée**) et forme un **bloc d'instructions**.

Par exemple observer le programme ci-dessous et avec plusieurs valeurs de **x** :

```
x = 5
if x < 10:
    print("plus petit que 10")
print("c'est fini")
```

À la suite du bloc d'instructions **if** on peut rajouter un bloc d'instructions **else** qui est exécuté **dans le cas contraire** (*sinon*).

```
x = 5
if x < 10:
    print("plus petit que 10")
else:
    print("plus grand que 10")
print("c'est fini")
```

Souvent ce sera intéressant si on demande à l'utilisateur la valeur à tester.

Question 2. Écrire un programme qui demande son âge à l'utilisateur, et lui affiche s'il est majeur ou mineur :

```
age = int(input("Votre âge :"))
if age ... :
    print("Vous êtes majeur")
else:
    ...
```

Question 3. Écrire un programme qui demande à l'utilisateur de faire un calcul lui-même (ne pas hésiter à mettre vos propres nombres ici) :

```
x = int(input("à combien est égal 12 + 13 ?"))
if ... :
    print("c'est correct")
else:
    print("c'est FAUX")
```

III Sinon, si...

Après avoir réalisé un premier test, on peut parfois avoir envie d'en réaliser un deuxième avant de passer au `else`. En français c'est ce que l'on appelle **sinon, si** et en Python cela se traduit par le mot-clé `elif`, contraction de l'anglais *else if*, suivi de son bloc d'instructions. Observer et tester le programme suivant :

```
x = int(input("Entrez un nombre :"))
if x > 0:
    print("positif")
elif x < 0:
    print("négatif")
else:
    print("zéro")
```

Question 4. Écrire un programme qui demande à l'utilisateur sa note. Si elle est au dessus de 16 il répond « très bien ». Sinon il faut vérifier si elle est en dessous de 6 et alors il répond... que ce n'est pas très bien (à vous d'imaginer le message!). Sinon, il répond que c'est correct.

```
note = int(input("Votre note :"))
if note ... 16:
    print("très bien")
elif ... :
    print(...)
else:
    print("ça va")
```

IV Les chaînes de caractères

Du texte écrit entre guillemets s'appelle en informatique une **chaîne de caractères**. On peut aussi la stocker dans une variable, et on peut aussi tester l'égalité entre chaînes de caractères! Par exemple essayer le programme suivant :

```
m = input("Entrez le mot de passe top secret :")
if m == "poquelin":
    print("C'est bon !")
else:
    print("C'est faux !")
```

Question 5. Choisir de vous-même une question et sa réponse et réaliser un petit quizz et appeler l'examineur pour le faire jouer.

Question 6. Créer un quizz de trois questions, qui ne répond pas nécessairement vrai ou faux à chaque fois, mais avec une variable **n** qui compte le nombre de bonnes réponses obtenues. Au départ **n** vaut 0 et à chaque bonne réponse on doit l'augmenter de 1.

```
n = 0
r = input("Première question : ...")
if r == ... :
    n = n + 1
r = input("Deuxième question : ...")
...
...
print("Vous avez", n, "bonnes réponses")
```