

# Chapitre 2

## Les structures de contrôle

STRUCTURES DE CONTRÔLES?? QUEL GROS MOT!! DISONS QUE NOUS ALLONS VOIR DANS CE CHAPITRE, TOUT CE QUI CONCERNE LES TESTS... ON AVANCE!!

### 2.1 Les instructions conditionnelles

#### 2.1.1 Définition

L'instruction conditionnelle détermine si le bloc d'instructions suivant est exécuté ou non. La condition est une expression booléenne dont la valeur détermine le bloc d'instructions exécuté.

#### Remarque 1 :

1. La syntaxe de cette instruction est :

```
Si (condition) alors
  Bloc d'instruction N°1;
  #ce bloc est exécuté si (condition) est vrai
sinon
  Bloc d'instructions N°2;
  #ce bloc est exécuté si (condition) est faux
```

2. L'un des deux blocs est obligatoirement exécuté, l'autre ne l'est pas
3. En Python, nous utiliserons **if...else...** en utilisant les 2 points : et l'**indentation**

```
If (condition):
  Bloc d'instruction N°1;
  #ce bloc est exécuté si (condition) est vrai
else:
  Bloc d'instructions N°2;
  #ce bloc est exécuté si (condition) est faux
```

#### Exemple 1 :

##### 1. Exemple simple

```
>>> a=-2
>>> if a>0:
    print("a est positif")
else:
    print("a est négatif")

a est négatif
```

2. Ecrivons l'algorithme qui lit deux entiers et affiche le plus grand des deux

```
#algo MaxDeDeuxEntiers

n1= float(input("Donner le premier entier "))
n2= float(input("Donner le second entier "));
if n1 >n2:
    max=n1
else :
    max:=n2;
print("Le plus grand des deux nombres est " ,max)
```

**Exercice 1 :**

Reprenre l’algorithme MaxDeDeuxEntiers en affichant les résultats dans l’ordre décroissant (*algorithme* MaxDeDeuxEntiers\_1)

**Remarque 2 :**

Une version plus simple peut être utilisée si l’alternative n’a pas lieu.

**Exemple 2 :**

Donner la valeur absolue d’un réel.

```
# algo ValeurAbsolue

x=float(input("Donner le réel: "))
if x<0:
    x=-1*x
print("La valeur absolue est ",x)
```

Attention à l’indentation!!

**2.1.2 Instructions conditionnelles imbriquées**

Il est possible d’imbriquer des blocs d’instructions conditionnelles les uns dans les autres.

**Remarque 3 :**

C’est à dire que nous pouvons faire un **if** dans un autre **if**

**Exemple 3 :**

1.
 

```
a = 0
if a > 0:
    print('positif')
else:
    if a < 0:
        print('négatif') # deux tabulations !
    else:
        print('zéro')
```
2. Ecrivons, par exemple, un algorithme qui associe, à la note, un commentaire.
 

```
#algo NotesCommentaires

note=float(input("Donner la note: "))
if note <= 8:
    print("la note ",note,"est très insuffisante")
else:
    if note <=10:
```

```

        print("Lanote",note,"est très moyenne")
    else:
        if note <=12:
            print("La note",note," est moyenne")
        else:
            if note <= 16:
                print("La note", note,"est bien")
            else:
                print("la note",note,"est très bien")

```

### 2.1.3 Important en Python

En python, « **else + if** » peut se simplifier en « **if + elif +else** »

#### Remarque 4 :

##### Remarque importante

1. La condition **if.....elif .... else** est une structure conditionnelle encore plus complète que la condition **if ..... else** qui va nous permettre cette fois-ci d'effectuer autant de tests que l'on souhaite et ainsi de prendre en compte le nombre de cas souhaité.

En effet, nous allons pouvoir ajouter autant de **elif** que l'on souhaite entre le **if** de départ et le **else** de fin et chaque **elif** va pouvoir posséder son propre test ce qui va nous permettre d'apporter des réponses très précises à différentes situations.

2. Il faut cependant faire attention à un point en particulier lorsqu'on utilise une structure Python **if.....elif .... else** : le cas où plusieurs **elif** possèdent un test évalué à **True** par Python. Dans ce cas là, vous devez savoir que seul le code du premier **elif** (ou du **if** si celui-ci est évalué à **True**) va être exécuté. En effet, Python sort de la structure conditionnelle dans son ensemble sans même lire ni tester la fin de celle-ci dès qu'un cas de réussite à été rencontré et que son code a été exécuté.
3. **elif..** pourrait être interprété en langage courant comme « sinon si.. »

#### Exemple 4 :

Nous allons simplifier les exemples précédents

```

1. # -*- coding: Latin-1 -*-
   # algo If_Elif_Simple_01

   a=float(input("Donner une valeur à a "))
   if a > 0:
       print('positif')
   elif a < 0:
       print('négatif') # deux tabulations !
   else:
       print('zéro')

   # -*- coding: Latin-1 -*-
   # algo NotesCommentaires

   note=float(input("Donner la note: "))
   if note <= 8:
       print("la note ",note,"est très insuffisante")
   elif note <=10:
       print("Lanote",note,"est très moyenne")
   elif note <=12:
       print("La note",note," est moyenne")
   elif note <= 16:

```

```
print("La note", note,"est bien")
else:
    print("la note",note,"est très bien")
```

**Exercice 2 :**

1. Ecrire un algorithme qui trie trois nombres réels  $n_1, n_2, n_3$  et qui donne ce tri par ordre décroissant. (*algorithme TriDe3Nombres*)
2. Ecrire un algorithme qui résout une équation du second degré à coefficients dans  $\mathbb{R}$  (*algorithme SecondDegre*)