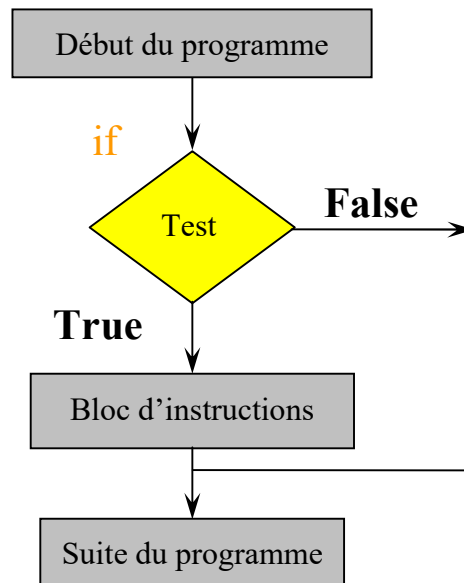


CHAPITRE 2 : LES CONDITIONS

I. L'instruction if.



Syntaxe :

```

if test:           # ne pas oublier le signe de ponctuation ':'
    bloc d'instructions  # attention à l'indentation
# suite du programme
    
```

Si le test est vraie (True) alors le bloc d'instructions est exécuté.

Si le test est fausse (False) on passe directement à la suite du programme.

Premier script

Nous allons commencer par créer le script Condition1.py :

Ouvrir IDLE :

Démarrer → Programmes → Python 3.X → IDLE (Python GUI)

File → New Window

Saisir le code source ci-dessous.

File → Save As

Répertoire : **sauvegarder obligatoirement dans votre répertoire personnel sur le réseau.**

Nom du fichier : Condition1.py

```
# script Condition1.py
```

```

ch = input("Note sur 20 : ")
note = float(ch)
if note >= 10.0:
    # ce bloc est exécuté si le test (note >= 10.0) est vrai
    print('J'ai la moyenne')
print('Fin du programme')
    
```

Pour exécuter le script :

Run → Run Module (ou touche F5)

```
>>>
```

```
Note sur 20 : 16
```

```
J'ai la moyenne
```

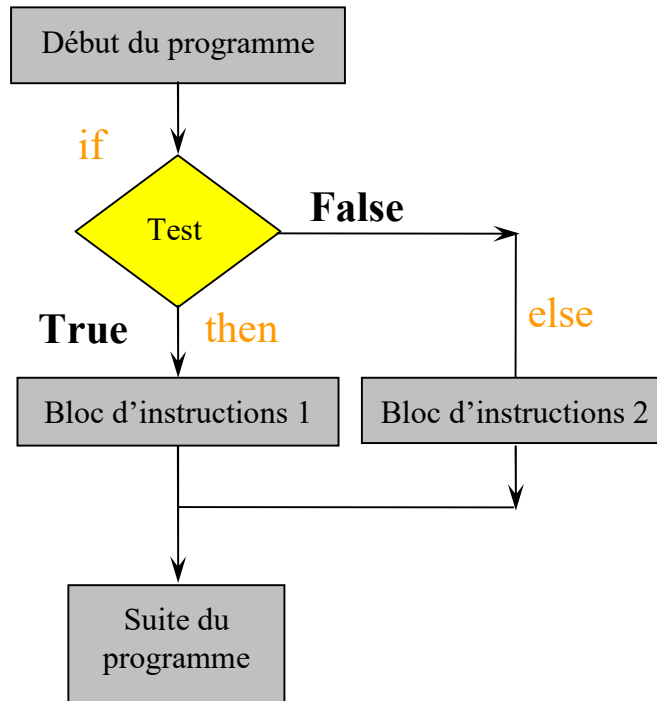
```
Fin du programme
```

>>>

Note sur 20 : 5

Fin du programme

II. L'instruction else.



Une instruction else est toujours associée à une instruction if.

Syntaxe :

```
if test:
    bloc d'instructions 1           # attention à l'indentation
else:
    bloc d'instructions 2           # else est au même niveau que if
                                    # attention à l'indentation
# suite du programme
```

Si le test est vraie (True) alors le bloc d'instructions 1 est exécuté.

Si le test est fausse (False) alors c'est le bloc d'instructions 2 qui est exécuté.

script Condition2.py

```
ch = input("Note sur 20 : ")
note = float(ch)           #conversion en réel
if note >= 10.0:
    # ce bloc est exécuté si le test (note >= 10.0) est vrai
    print('J'ai la moyenne')
else:
    # ce bloc est exécuté si le test (note >= 10.0) est fausse
    print('C'est en dessous de la moyenne')
print('Fin du programme')
```

>>>

Note sur 20 : 15

J'ai la moyenne

Fin du programme

>>>

```
Note sur 20 : 8.5
C'est en dessous de la moyenne
Fin du programme
>>>
Note sur 20 : 56
J'ai la moyenne
Fin du programme
```

Pour traiter le cas des notes invalides (<0 ou >20), on peut imbriquer des instructions conditionnelles :

```
# script Condition3.py

ch = input("Note sur 20 : ")
note = float(ch)
if note > 20.0 or note < 0.0:
    # ce bloc est exécuté si le test (note>20.0 or note<0.0) est vrai
    print('Note invalide !')
else:
    # ce bloc est exécuté si le test (note>20.0 or note<0.0) est fausse
    if note >= 10.0:
        # ce bloc est exécuté si le test (note>=10.0) est vrai
        print('J'ai la moyenne')
    else:
        # ce bloc est exécuté si le test (note>=10.0) est fausse
        print('C'est en dessous de la moyenne')
print('Fin du programme')
>>>
Note sur 20 : 56
Note invalide !
Fin du programme
>>>
Note sur 20 : 14.6
J'ai la moyenne
Fin du programme
```

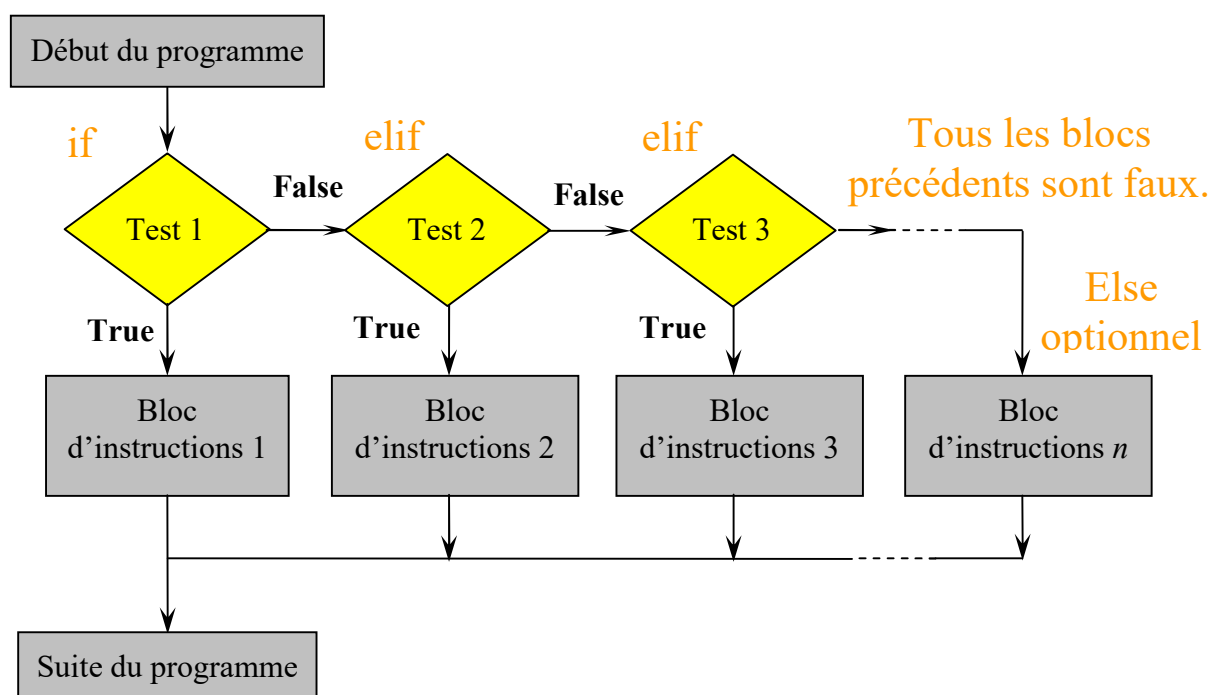
On ajoute encore un niveau d'imbrication pour traiter les cas particuliers 0 et 20 :

```
# script Condition4.py

ch = input("Note sur 20 : ")
note = float(ch)
if note > 20.0 or note < 0.0:
    print('Note invalide !')
else:
    if note >= 10.0:
        print('J'ai la moyenne')
        if note == 20.0:
            # ce bloc est exécuté si le test (note==20.0) est vrai
            print('C'est même excellent !')
    else:
        print('C'est en dessous de la moyenne')
        if note == 0.0:
            # ce bloc est exécuté si le test (note==0.0) est vrai
```

```
print(... lamentable !)  
print('Fin du programme')  
>>>  
Note sur 20 : 20  
J'ai la moyenne  
C'est même excellent !  
Fin du programme  
>>>  
Note sur 20 : 3  
C'est en dessous de la moyenne  
Fin du programme
```

III. L'instruction elif.



Une instruction elif (contraction de else if) est toujours associée à une instruction if.

Syntaxe :

```
if test 1:  
    bloc d'instructions 1  
elif test 2:  
    bloc d'instructions 2  
elif test 3:  
    bloc d'instructions 3    # ici deux instructions elif, mais il n'y a pas de limitation  
else:  
    bloc d'instructions n    # bloc optionnel par défaut  
# suite du programme
```

Si le test 1 est vrai alors le bloc d'instructions 1 est exécuté, et on passe à la suite du programme.

Si le test 1 est faux alors on évalue le test 2 :

- si le test 2 est vrai on exécute le bloc d'instructions 2, et on passe à la suite du programme.
- si le test 2 est faux alors on évalue le test 3, etc...

Le bloc d'instructions 4 est donc exécuté si tous les tests sont faux (c'est le bloc "par défaut").

Parfois il n'y a rien à faire.

Dans ce cas, on peut omettre l'instruction else :

```
if test 1:
    bloc d'instructions 1
elif test 2:
    bloc d'instructions 2
elif test 3:
    bloc d'instructions 3
# suite du programme
```

L'instruction elif évite souvent l'utilisation de conditions imbriquées (et souvent compliquées).

Exemple :

```
# script Condition5.py
# ce script fait la même chose que Condition4.py

note = float(input("Note sur 20 : "))
if note == 0.0:
    print('C\'est en dessous de la moyenne')
    print('... lamentable !')
elif note == 20.0:
    print('J'ai la moyenne')
    print('C\'est même excellent !')
elif note < 10.0 and note > 0.0:      # ou bien : elif 0.0 < note < 10.0:
    print('C\'est en dessous de la moyenne')
elif note >= 10.0 and note < 20.0:    # ou bien : elif 10.0 <= note < 20.0:
    print('J'ai la moyenne')
else:
    print('Note invalide !')
print('Fin du programme')

>>>
Note sur 20 : 20
J'ai la moyenne
C'est même excellent !
Fin du programme
>>>
Note sur 20 : 3
C'est en dessous de la moyenne
Fin du programme
>>>
Note sur 20 : 77
Note invalide !
Fin du programme
```

LES CONDITIONS : TESTS

I. Écrire un programme qui permute le contenu de deux variables a et b.

II. Valeur absolue d'un nombre.

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre réel.
- En sortie : la valeur absolue de ce nombre.

III. Signe d'un nombre.

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre réel.
- En sortie : le signe de ce nombre (positif ; négatif ou nul).

IV. Parité d'un nombre.

Concevoir un programme avec les contraintes suivantes.

- En entrée : un nombre entier positif.
- En sortie : la parité de ce nombre.

V. a) Le numéro de sécurité sociale est constitué de 13 chiffres auquel s'ajoute la clé de contrôle (2 chiffres).

La clé de contrôle est calculée par la formule : $97 - (\text{numéro de sécurité sociale modulo } 97)$.

Écrire un programme qui contrôle la validité d'un numéro de sécurité sociale.

On pourra utiliser la fonction `int()` pour convertir le type `str` en type `int`.

Par exemple :

```
>>>
Entrer votre numéro de sécurité sociale (13 chiffres) : 1891126108268
Entrer votre clé de contrôle (2 chiffres) : 91
Votre numéro de sécurité sociale est valide.
>>>
Entrer votre numéro de sécurité sociale (13 chiffres) : 2891126108268
Entrer votre clé de contrôle (2 chiffres) : 91
Votre numéro de sécurité sociale est INVALIDE !
>>>
```

b) Reprendre le programme en saisissant le numéro de sécurité sociale à 15 chiffres.

VI. Écrire un script qui demande la note au bac et qui affiche la mention correspondante.

Par exemple :

```
>>>
Note au bac (sur 20) : 13.5
Bac avec mention Assez Bien
>>>
Note au bac (sur 20) : 10.9
Bac avec mention Passable
>>>
Note au bac (sur 20) : 4
Recalé
>>>
```

VII. Écrire un script qui calcule l'indice de masse corporelle $IMC = \text{masse}/\text{taille}^2$ avec la masse en kg et la taille en m d'un adulte et qui en donne l'interprétation.

- $IMC < 18.5$: maigre ;
- $18,5 \leq IMC < 25$: corpulence normale ;
- $25 \leq IMC < 30$: surpoids ;
- $IMC \geq 30$: obésité).

Par exemple :

```
>>>
Votre taille en cm ? 170
Votre masse en kg ? 68.5
IMC = 23.7024221453 kg/m²
Interprétation : corpulence normale
>>>
```

VIII. 8.1. Écrire un script qui résout l'équation du second degré : $ax^2 + bx + c = 0$

Par exemple :

```
>>>
Résolution de l'équation du second degré :  $ax^2 + bx + c = 0$ 
Coefficient a ? 1
Coefficient b ? -0.9
Coefficient c ? 0.056
Discriminant : 0.586
Deux racines distinctes :
x1=0.0672468158199
x2=0.83275318418
>>>
Résolution de l'équation du second degré :  $ax^2 + bx + c = 0$ 
Coefficient a ? 2
Coefficient b ? 1.5
Coefficient c ? 4
Discriminant : -29.75
Il n'y a pas de racine réelle.
```

8.1. Rajouter les racines complexes.