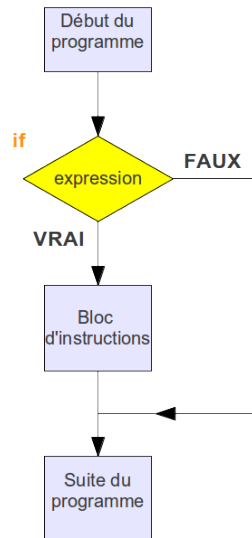


Chapitre 2 - Les conditions

L'instruction if



Syntaxe

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

Si l'expression est vraie (True) alors le bloc d'instructions est exécuté. Si l'expression 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 → IDLE (Python GUI)
File → New Window Copier puis coller le code source ci-dessous :

```
# script Condition1.py

chaine = input("Note sur 20 : ")
note = float(chaine)
if note >= 10.0:
    # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
    print("J'ai la moyenne")
print("Fin du programme")
```

File → Save As

Au Lycée des Flandres : Répertoire : Votre dossier de travail, NSI, Python, Cours Nom du fichier : Condition1.py Exemple : *H:\Travail\NSI\Python\Cours\Condition1.py* Pensez aussi à partager vos travaux sur Google Drive... 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
```

L'instruction else

Une instruction `else` est toujours associée à une instruction `if`

Syntaxe

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

Si l'expression est vraie (True) alors le bloc d'instructions 1 est exécuté. Si l'expression est fausse (False) alors c'est le bloc d'instructions 2 qui est exécuté.

script Condition2.py

```
chaine = input("Note sur 20 : ")
note = float(chaine)
if note >= 10.0:
    # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
    print("J'ai la moyenne")
else:
    # ce bloc est exécuté si l'expression (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
```

```
chaine = input("Note sur 20 : ")
note = float(chaine)
if note > 20.0 or note < 0.0:
    # ce bloc est exécuté si l'expression (note > 20.0 or note < 0.0) est vraie
    print("Note invalide !")
else:
    # ce bloc est exécuté si l'expression (note > 20.0 or note < 0.0) est fausse
    if note >= 10.0:
        # ce bloc est exécuté si l'expression (note >= 10.0) est vraie
        print("J'ai la moyenne")
    else:
        # ce bloc est exécuté si l'expression (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
```

```
chaine = input("Note sur 20 : ")
note = float(chaine)
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 l'expression (note == 20.0) est vraie
        print("C'est même excellent !")
    else:
        print("C'est en dessous de la moyenne")
        if note == 0.0:
            # ce bloc est exécuté si l'expression (note == 0.0) est vraie
            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

```

L'instruction elif

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

Syntaxe

```

if expression 1:
    bloc d'instructions 1
elif expression 2:
    bloc d'instructions 2
elif expression 3:
    bloc d'instructions 3    # ici deux instructions elif, mais il n'y a pas de limitation
else:
    bloc d'instructions 4
# suite du programme

```

Si l'expression 1 est vraie alors le bloc d'instructions 1 est exécuté, et on passe à la suite du programme. Si l'expression 1 est fausse alors on teste l'expression 2 :

- si l'expression 2 est vraie on exécute le bloc d'instructions 2, et on passe à la suite du programme.
- si l'expression 2 est fausse alors on teste l'expression 3, etc.

Le bloc d'instructions 4 est donc exécuté si toutes les expressions sont fausses (c'est le bloc "par défaut"). Parfois il n'y a rien à faire. Dans ce cas, on peut omettre l'instruction `else` :

```

if expression 1:

```

```

        bloc d'instructions 1
elif expression 2:
        bloc d'instructions 2
elif expression 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

```

Exercices

Exercice 2.1 * 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 : $97 - (\text{numéro de sécurité sociale modulo } 97)$

Ecrire un script qui contrôle la validité d'un numéro de sécurité sociale. On pourra utiliser la fonction `int()` pour convertir le type `str` en type `int`. 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 !
>>>
```

Exercice 2.2 * Nombre entier non signé et signé

Dans un octet, on peut stocker un nombre entier compris entre $0b00000000 = 0$ et $0b11111111 = 255$ (entier non signé, en numération binaire naturel). On peut aussi stocker un entier compris entre -128 et $+127$ (entier signé, représentation dite en complément à deux).

En complément à deux, les nombres négatifs sont codés de la manière suivante :

- -1 correspond à 255 en binaire naturel
- -2 correspond à 254 en binaire naturel
- ...
- -127 correspond à 129 en binaire naturel
- -128 correspond à 128 en binaire naturel

1. Ecrire un script qui donne la correspondance entre entier signé et entier non signé.

Par exemple :

```
>>>
Entrer un entier signé en complément à deux (-128 à +127): 25
La représentation en binaire naturel est : 25
>>>
Entrer un entier signé en complément à deux (-128 à +127): -15
La représentation en binaire naturel est : 241
```

2. Ecrire un script qui donne la correspondance entre entier non signé et entier signé.

Par exemple :

```
>>>
    Entrer un nombre entier (0 à 255): 250
    Cela représente l'entier signé : -6
```

Exercice 2.3 * Ecrire 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
>>>
```

Exercice 2.4 * Ecrire un script qui calcule l'indice de masse corporelle (IMC) d'un adulte et qui en donne l'interprétation (corpulence normale, surpoids...). Calculer mon IMC. Il se calcule simplement en divisant le poids (en kg) par le carré de la taille (m).

Indice de masse corporelle (IMC)	Interprétation (d'après l'OMS)
moins de 18,5	Insuffisance pondérale (maigreur)
18,5 à 25	Corpulence normale
25 à 30	Surpoids
30 à 35	Obésité modérée
35 à 40	Obésité sévère
plus de 40	Obésité morbide ou massive

Par exemple :

```
>>>
Votre taille en cm ? 170
Votre masse en kg ? 68.5
IMC = 23.70 kg/m²
Interprétation : corpulence normale
>>>
```

Exercice 2.5 ** - Niveau première en mathématiques -
Ecrire 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 solutions :
```

```
0.0672468158199
```

```
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 solution.
```

QCM

QCM sur les structures conditionnelles