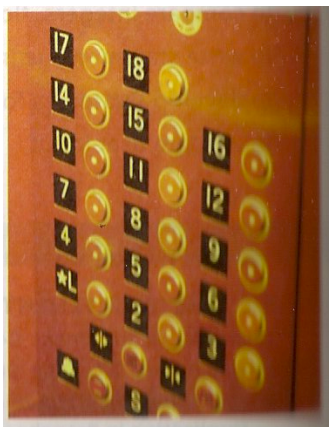
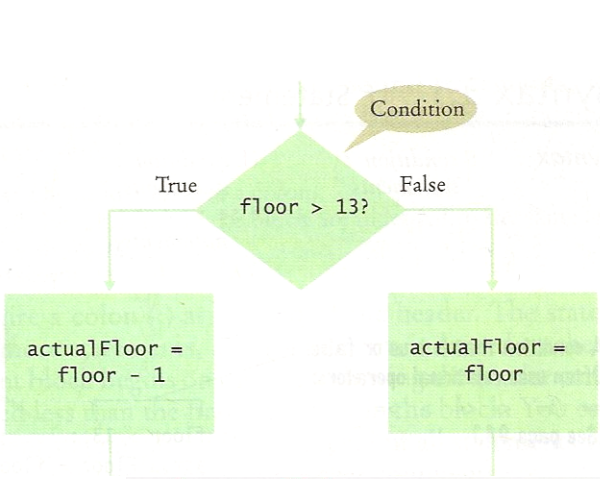
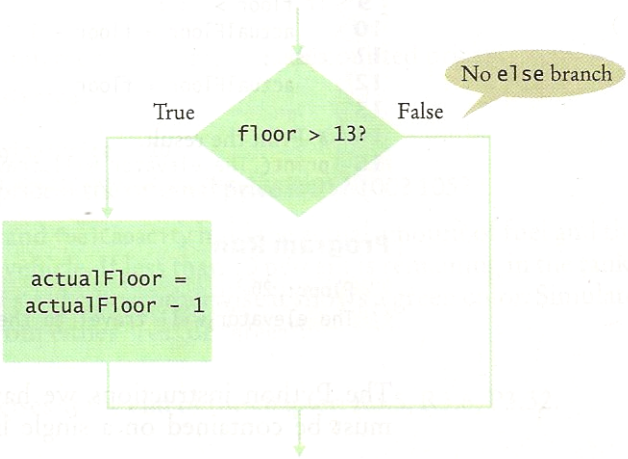
Cours 3 – Décision IF / ELIF / ELSE

Dans certains pays, par supersitition, l'étage numéro 13 des immeubles n'est pas indiqué, celui ci existe bien sûr physiquement, il est simplement numéroté 14 ! L'ordinateur qui contrôle l'ascenseur doit s'adapter. Une instruction conditionnelle est nécessaire.







Syntax 3.1 if Statement

Syntax

```
if condition :
   statements
```

```
if condition :
   statements1
else :
   statements2
```

A condition that is true or false. Often uses relational operators: == != < <= > >= (See Table 1.)

The colon indicates a compound statement.

```
if floor > 13 :
   actualFloor = floor - 1
else :
   actualFloor = floor
```

If the condition is true, the statement(s) in this branch are executed in sequence; if the condition is false, they are skipped.

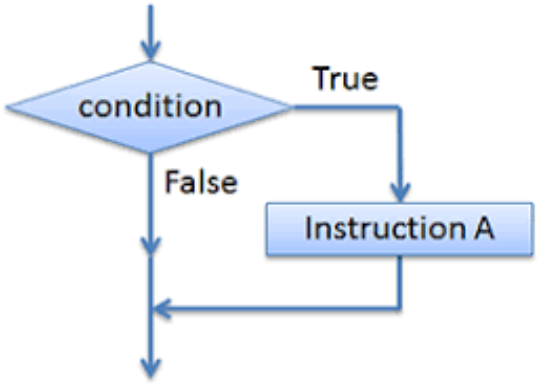
Omit the else branch if there is nothing to do.

If the condition is false, the statement(s) in this branch are executed in sequence; if the condition is true, they are skipped.

The if and else clauses must be aligned.

A / Tests simples

Si nous voulons pouvoir écrire des applications véritablement utiles, il nous faut des techniques permettant d'aiguiller le déroulement du programme dans différentes directions, en fonction des circonstances rencontrées. Pour ce faire, nous devons disposer d'instructions capables de **tester une certaine condition** et de modifier le comportement du programme en conséquence. La plus simple de ces instructions conditionnelles est l'instruction `if`.



Une instruction conditionnelle est une instruction pouvant s'exécuter seulement si une condition est vérifiée par l'état courant. Pour cela on utilise l'instruction `if`, qui a en Python la syntaxe suivante :

```
if condition:
    bloc d'instructions
```
→ Exécute le bloc d'instructions uniquement si la condition est vérifiée.

```
if x % 2 == 1:
    x = x + 1
```
→ Ajoute 1 à la variable x seulement si dans l'état courant elle a une valeur impaire. Si x a une valeur paire, l'instruction d'ajout n'est pas exécutée

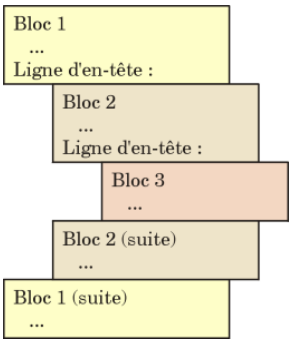
Note : `%` retourne le reste de la division.

Table 1 Relational Operators

Python	Math Notation	Description
>	>	Greater than
>=	≥	Greater than or equal
<	<	Less than
<=	≤	Less than or equal
==	=	Equal
!=	≠	Not equal

B / L'indentation

On a déjà vu dans le paragraphe sur les séquences d'instructions qu'un bloc d'instructions est une suite d'instructions qui s'exécutent les unes après les autres. Pour savoir quand s'arrête le bloc d'instructions à l'intérieur d'une instruction `if`, il est nécessaire D'INDENTER LES INSTRUCTIONS du bloc.



Au sein d'un bloc le niveau d'indentation doit être le même. Le code suivant est valide, et la ligne `y=y//x` ne s'exécute que si `x` est non nul :

```
if x!=0:
    y=3
    y=y//x
```

Note : // retourne la partie entière de la division.

Ce code n'est pas valide, pourquoi ?

```
if x!=0:
    y=3
        y=y//x
```

La première instruction qui suit une instruction conditionnelle et qui est placée au même niveau d'indentation que l'instruction if marque la fin du bloc. En effet, seules les instructions indentées font partie du bloc. Ici la dernière instruction s'exécute toujours après le `if` :

```
if x!=0:
    y=3
y=y//x
```

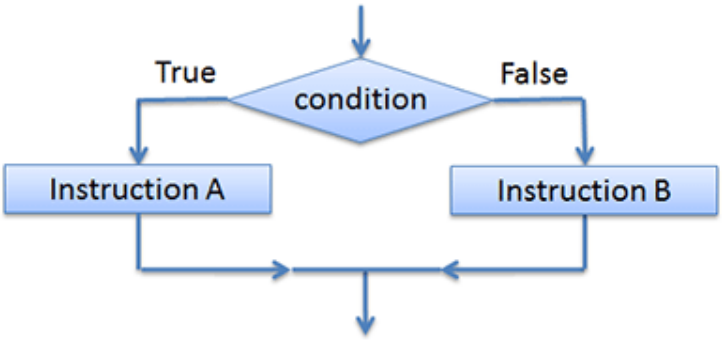
En cas de `if` au sein d'un `if`, le second bloc doit avoir un niveau d'indentation encore supérieur. Par exemple :

```
if x!=0:
    y=x*x
    if y%2 == 0:
       y=y+1
    x=x+y
```

Ci-dessus, l'instruction `y=y+1` n'est exécutée que si la seconde condition est vérifiée.

C / Tests avec alternatives

C'est l'instruction `else` (« sinon », en anglais) qui permet de programmer une exécution alternative, dans laquelle le programme doit choisir entre deux possibilités. On peut faire mieux encore en utilisant aussi l'instruction `elif` (contraction de « `else if` », « sinon si »).




```
prix=250

if prix > 100:
    remise = prix*25/100
else:
    remise = prix*10/100

print(remise)
```

```
62.5
```

```
lettre = "A"

if lettre == "B":
   print("la lettre est B")

elif lettre == "C":
   print("la lettre est C")

elif lettre == "A":
   print("la lettre est A")

else:
   print("la lettre n'est pas A, B ou C")
```

```
la lettre est A
```

L'instruction `if` autorise plusieurs conditions comme sur l'exemple suivant qui n'autorise l'accès qu'aux enfants entre 8 et 12 ans.

```
age = 18

if ((age>= 8) and (age<= 12)):
    print("YOU ARE ALLOWED. WELCOME !")
else:
    print("SORRY ! YOU ARE NOT ALLOWED. BYE !")
```

```
SORRY ! YOU ARE NOT ALLOWED. BYE !
```

D / Tests imbriqués

Comme on a pu le voir dans le paragraphe sur l'indentation, il est possible de réaliser une instruction conditionnelle dans une instruction conditionnelle. On parle alors de tests imbriqués. Les tests imbriqués servent à séparer une situation avec plus de deux alternatives. Par exemple le programme suivant exécute trois instructions différentes suivant la valeur de `x%3`.

```
if x%3 == 0:
    x=x+1
else:
    if x%3 == 1:
        x=x-1
    else:
        x=2*x
```