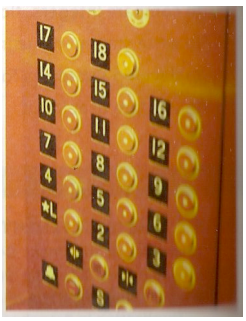
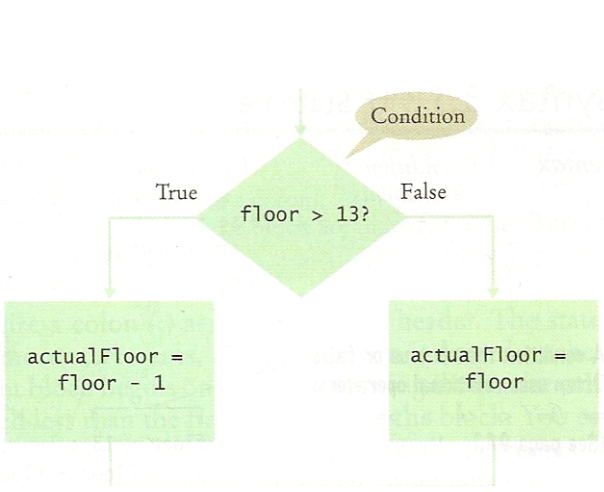
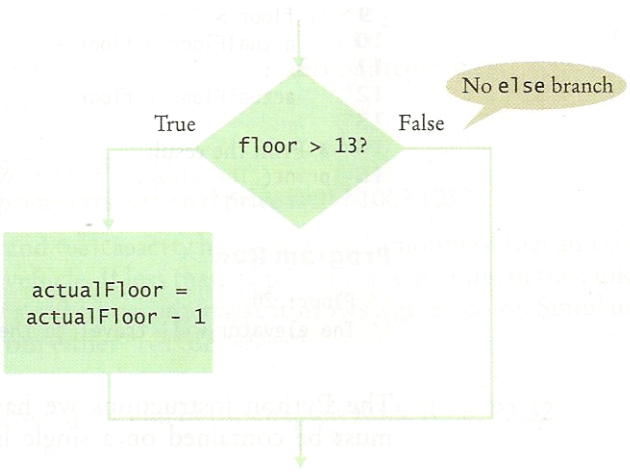
Instructions conditionnelles – les tests –

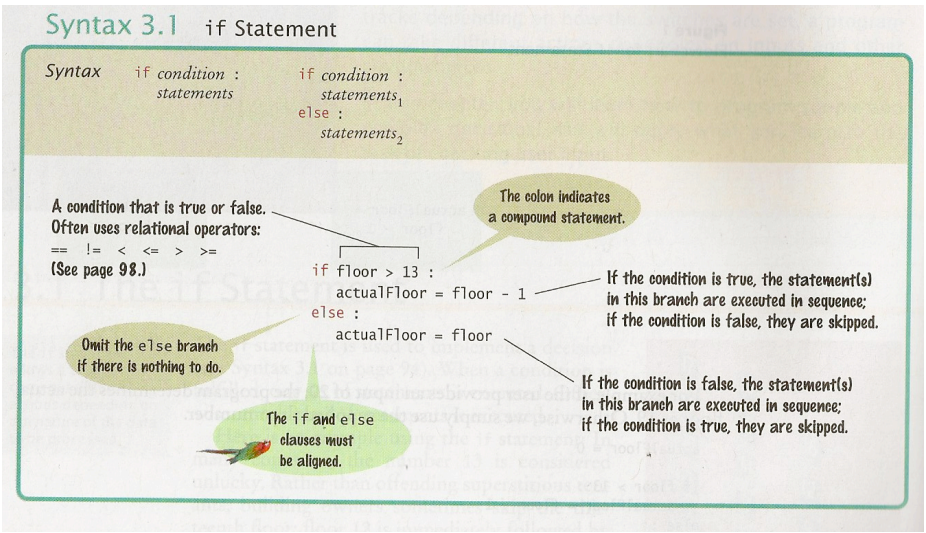
IF / ELIF / ELSE

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



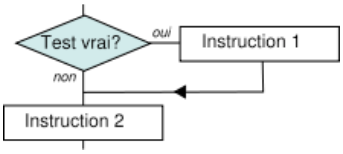






Instructions conditionnelles – les tests

1. Introduction



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`.

2. Tests simples

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 :

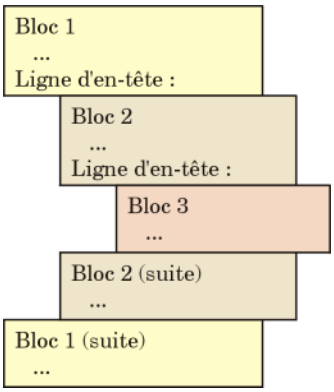
```
if condition:
    bloc d'instructions
```

Exécute le bloc d'instructions uniquement si la condition est vérifiée

```
if x % 2 == 1:
    x = x + 1
```

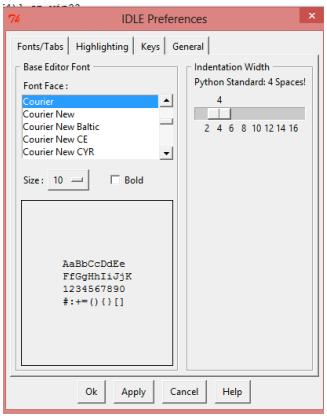
Ajoute 1 à la variable x seulement si dans l'état courant elle a une valeur impaire. Si x a une valeur paire, l'instruction d'ajout n'est pas exécutée

3. 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.

Dans Python, on peut définir la valeur d'indentation (4 espaces en général).



Remarques :

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
```

mais celui-ci ne l'est pas :

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

Instructions conditionnelles – les tests

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
```

Là l'instruction `y = y + 1` n'est exécutée que si la seconde condition est vérifiée.

Exercice : Quel est le résultat de l'exécution des instructions ?

```
if a == 0:
    b = 4
else:
    c = 5
b = 1
```

et

```
if a == 0:
    b = 4
else:
```

```
    c = 5
b = 1
```

dans l'état

a	b	c
0	2	1

?

4. 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 »).

Les opérateurs de comparaison sont les suivants :

```
x == y      # x est égal à y
x != y      # x est différent de y
x > y       # x est plus grand que y
x < y       # x est plus petit que y
x >= y      # x est plus grand que, ou égal à y
x <= y      # x est plus petit que, ou égal à y
```

Exercice : Ecrire un programme qui demande à l'utilisateur d'entrer un nombre décimal, puis selon cette valeur affiche 3 possibilités : le nombre est positif, le nombre est négatif ou le nombre est nul.

```
Variable a en entier
Début
        Ecrire « entrez un nombre décimal : »
        Lire a
        Si a > 0 alors
                Ecrire « le nombre est positif »
        Sinon si a < 0 alors
                Ecrire « le nombre est négatif »
        Sinon alors
                Ecrire « le nombre est nul »
Fin
```

File Edit Format Run Options Windows Help

```
#programme cours sur les tests

a=int(input("entrez un nombre décimal:"))
if a > 0:
    print("le nombre est positif")
elif a < 0:
    print("le nombre est négatif")
else:
    print("le nombre est nul")
```

Instructions conditionnelles – les tests

5. 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***. Lorsque les tests imbriqués ne servent qu'à séparer une situation en plus de deux cas de figure.
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
```

Exercice :

1°) Que fait ce programme ?

```
if a>b:
    if a>c:
        m = a
    else:
        m = c
else:
    if b>c:
        m = b
    else:
        m = c
```

Les contenus des variables `a`, `b` et `c` ne sont pas modifiés dans ce programme : il s'agit de données à partir desquelles on calcule une valeur, ici dans la variable `m`.
Dérouler ensuite le programme pour quelques valeurs de `a`, `b` et `c` permet rapidement de conjecturer qu'à la fin de l'exécution de ce programme `m` contient le maximum des trois valeurs. Il serait judicieux d'appeler plutôt cette variable `max` pour rendre son rôle évident et éviter par exemple la confusion avec un minimum.

2°) Le programme suivant permet de déterminer si un individu est en surpoids, par le biai du calcul de son indice de massecorporelle. Adapter ce programme pour qu'il détermine également si l'individu est en sous-poids, ce qui correspond à un indice de masse corporelle inférieur à 18.

```
masse = float(input("Quelle est votre masse en kg ?"))
taille = float(input("Quelle est votre taille en m ?"))
imc = masse / taille**2
if imc > 25 :
    print("Vous etes en surpoids.")
```

Il suffit donc d'ajouter à la fin du programme les deux lignes suivantes.

```
if imc < 18:
    print("Vous etes en sous-poids.")
```