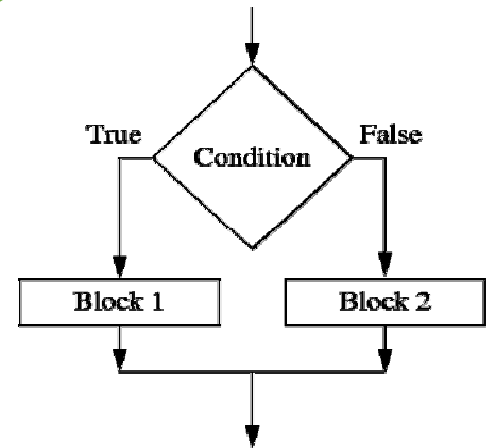


1 Structura IF

Dar dacă utilizatorul introduce de la tastatură litere în loc de cifre? Ce se întâmplă?
Ei bine, vom obține o eroare. Hai să ne protejăm împotriva ei!

```
vrs = input( "Si acum, pe bune, varsta ta este: ")
if vrs.isdigit():
    vrs = int(vrs)
    print("Ahaaa...deci tu ai "+str(vrs)+ " ani! ")
else:
    print("Te rog sa introduci o varsta valida!")
    vrs = input( " Deci, te intreb din nou, varsta ta este ")
    print(gata)
```

Wow! Am folosit prima structură condițională IF / ELSE
Folosim funcția `isdigit()` ca să verificăm dacă datele introduse de utilizator sunt numai cifre. Această funcție returnează True sau False.



În algoritmică secvența de mai sus s-ar desena așa:
Dealtfel toate programele mai complexe ar trebui începute cu desenarea unor scheme de acest fel.
Cum citim aceasta secvență?

Dacă utilizatorul a introdus date corecte (adică numai cifre) atunci putem transforma șirul de caractere introdus într-un număr, **dacă nu**, îi mai cerem încă o dată să introducă vârsta. Adică, în engleză **IF ... then ELSE**.

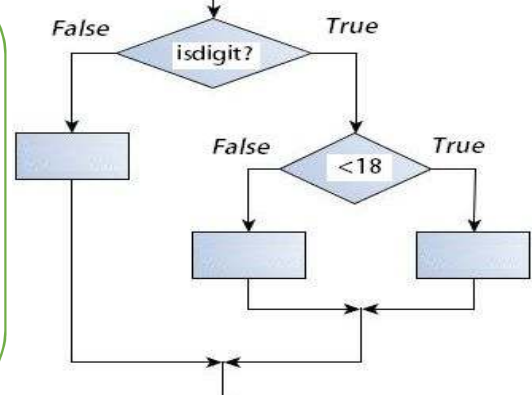
Condiția care se scrie după **IF** trebuie să fie întotdeauna o variabilă (expresie) de tip Boolean (adică True sau False)

Observă **indentările** (spațiile de la începutul rândurilor) din structura IF.
Sunt extrem de importante în Python! Acestea indică secvențele de cod care se execută în fiecare caz (dacă da, dacă nu). Observă că rândurile 3,4, 6 și 7 sunt perfect aliniate. Observă că ultimul print este aliniat iar de la începutul rândului, ceea ce înseamnă că această comandă se va executa la ieșirea din structura IF, indiferent de cazul tratat.

Și acum o structură IF mai complicată:

```
vrs = input( "Si acum, pe bune, varsta ta este: ")
if vrs.isdigit():
    vrs = int(vrs)
    if vrs < 18 ):
        print("Ahaa deci tu ai "+str(vrs)+ " ani! ")
    else:
        print(" Tu ai "+str(vrs)+ " ani. Esti major! ")
else:
    print("Te rog sa introduci o varsta valida!")
    vrs = input( " Deci, te intreb din nou, varsta ta este ")
```

Schema ar fi cam așa:



2 Structura repetitivă WHILE

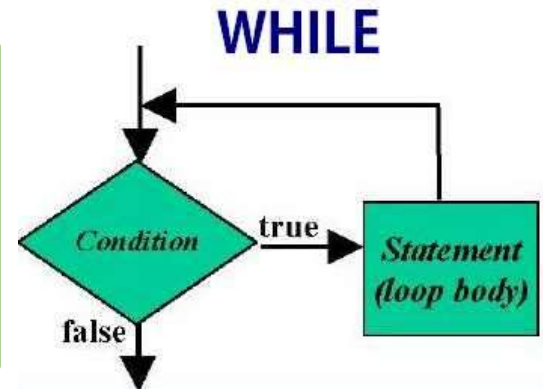
Ai testat codul de mai sus? Problema este că utilizatorul are doar o singură încercare de a introduce vârsta sa. Apoi programul se termină.

Noi am dori să i se ceară introducerea vârstei de mai multe ori, până când va introduce o vârstă corectă (numerică), adică **atâta timp cât** greșește să fie întrebat din nou.

Adică, în engleză .. **WHILE**.

```
vrs = input("Si acum, pe bune, varsta ta este: ")
while not vrs.isdigit():
    print("Te rog sa introduci o varsta valida!")
    vrs = input("Deci, te intreb din nou, varsta ta este ")

vrs = int(vrs)
if (vrs < 18):
    print("Ahaa deci tu ai "+str(vrs)+ " ani! ")
else:
    print("Tu ai "+str(vrs)+ " ani. Esti major! ")
```



Hai încercă acum:

```
nume = input("Numele tau e: ")
while True:
    print("Hello," +nume + " You are the best! ")
```

Ce se întâmplă ?

Ai intrat într-o buclă infinită!

Apasă **CTRL+ C** ca să întrerupi procesul!

În orice structură repetitivă trebuie să ai grija să se poată ieși din ea. Trebuie să se întâmple ceva în buclă pentru ca să iasă, fie o introducere de la tastatură, fie niște calcule.

Dacă condiția de sus (de lângă while) nu mai este îndeplinită nu se mai intră în buclă.

Noi am pus condiția True (adică e mereu îndeplinită), deci trebuie să ne asigurăm că în interiorul buclei se va produce un eveniment care să o spargă!

```
nume = input("Numele tau e: ")
while True:
    print("Hello," +nume + " You are the best! ")
    continuam = input("mai vrei D/N? ")
    if continuam == "N":
        break
    print("Gata! ")
```

Aici ieșirea din buclă este asigurată de **break**.

Observă linia 5, când testăm valoarea unei variabile, folosim **==** în loc de **=**

3 Structura repetitivă FOR. Este asemănătoare cu WHILE, dar aici trebuie să știm de la început de câte ori vrem să parcurgem bucla.

```
sum = 0
for i in range(10):
    sum = sum + i
    print("Suma este: ", sum)
```

Ține minte funcția **range(n)**. Ea generează un șir de n elemente, numerele de la 0 la n-1.

De exemplu **range(5,8)** generează numerele 5,6,7. Testează!

```
textul_meu = input("Introdu un text: ")
for lit in textul_meu:
    print(lit)
```

Putem parcurge cu **for** și elementele unei liste sau chiar ale unui șir de caractere (un string).

Ține minte acest exemplu, ne va fi de mare folos!