

# Controlul fluxului de programare - Partea 2

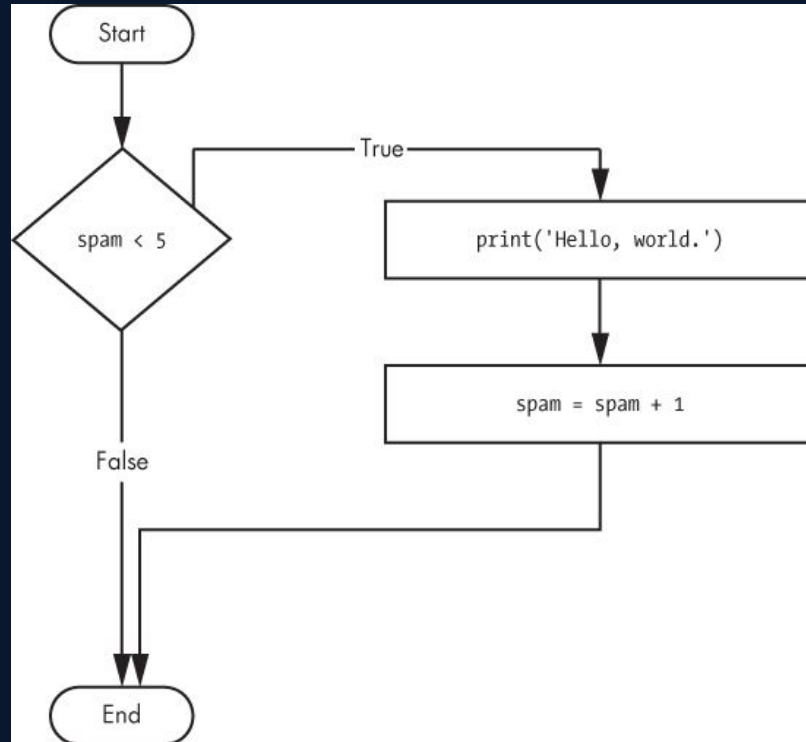


# Instrucțiunea *while* [1]

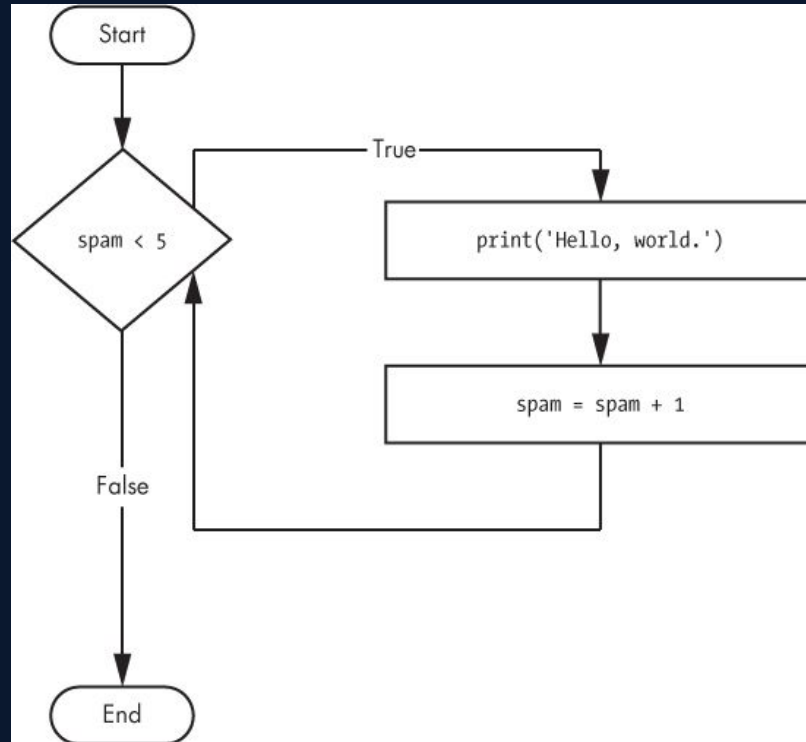
Puteți face ca un bloc de cod să se execute din nou și din nou folosind o instrucțiune *while*. Clauza instrucțiunii *while* va fi executată atâta timp cât condiția instrucțiunii *while* este Adevărată. În cod, o instrucțiune *while* constă întotdeauna din următoarele:

- Cuvântul cheie *while*;
- O condiție (adică o expresie care se evaluează la adevărat sau fals);
- Două puncte;
- Începând cu următoarea linie, un bloc de cod indentat (numit *clauza while*);

# Instrucțiunea *while* [1]



# Instrucțiunea *while* [1]

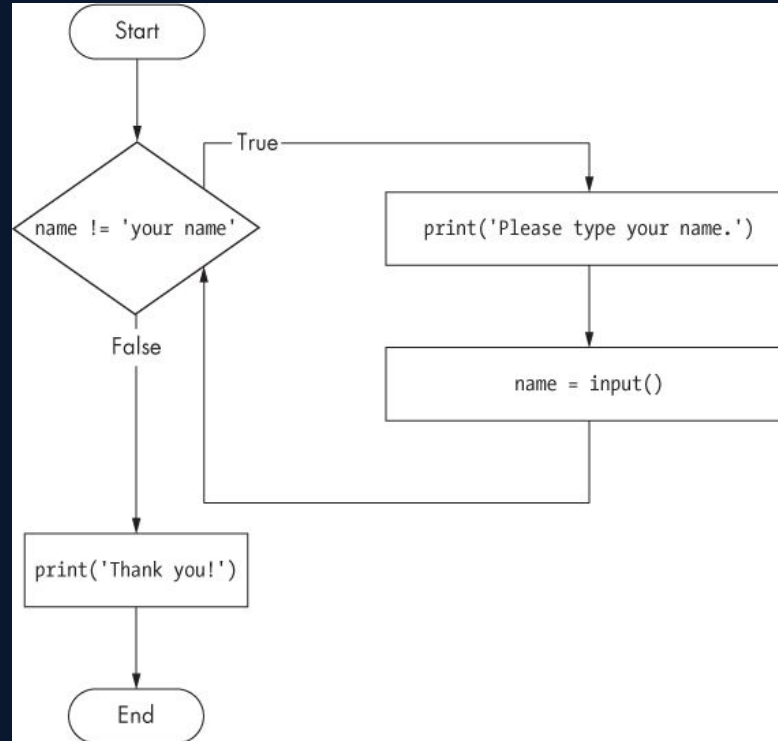




# Un exemplu de buclă while

```
❶ name = ''  
  
❷ while name != 'your name':  
    print('Please type your name.')  
    ❸ name = input()  
  
❹ print('Thank you!')
```

# Un exemplu de buclă while





## Un exemplu de buclă while

Dacă nu introduceți niciodată „your name”, atunci starea buclei while nu va fi niciodată False, iar programul va continua să ceară utilizatorului să tasteze un nume pentru totdeauna. Aici, apelul *input()* permite utilizatorului să introducă șirul corect pentru a face programul să meargă mai departe. În alte programe, starea s-ar putea să nu se schimbe niciodată, iar aceasta poate fi o problemă. Să vedem cum puteți ieși forțat dintr-o buclă *while*.

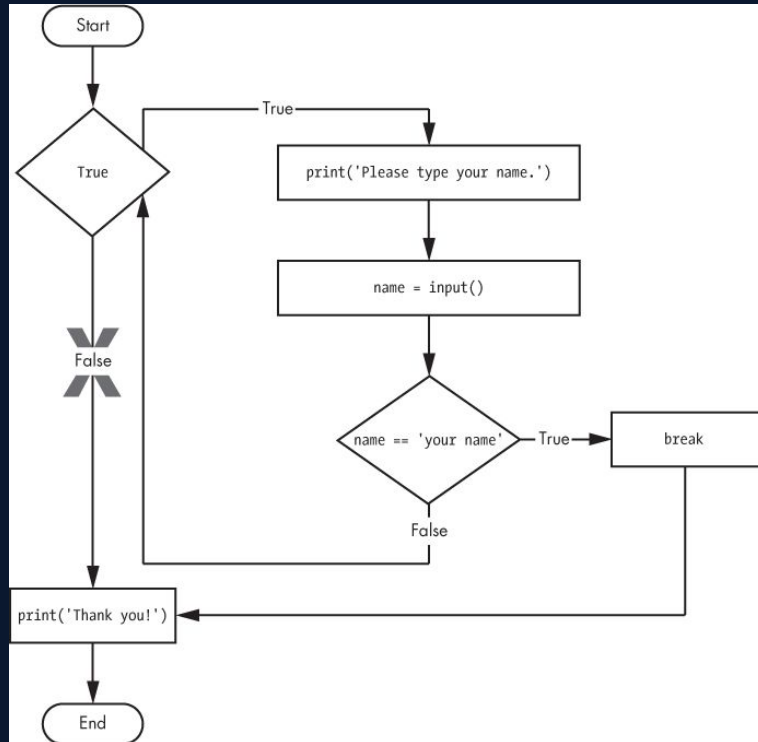


# Instrucțiunea break [1]

Există o comandă pentru a evada mai rapid clauza unei bucle *while*. Dacă execuția ajunge la o instrucțiune *break*, execuția iese imediat din clauza buclei *while*. În cod, o instrucțiune *break* conține pur și simplu cuvântul cheie *break*.



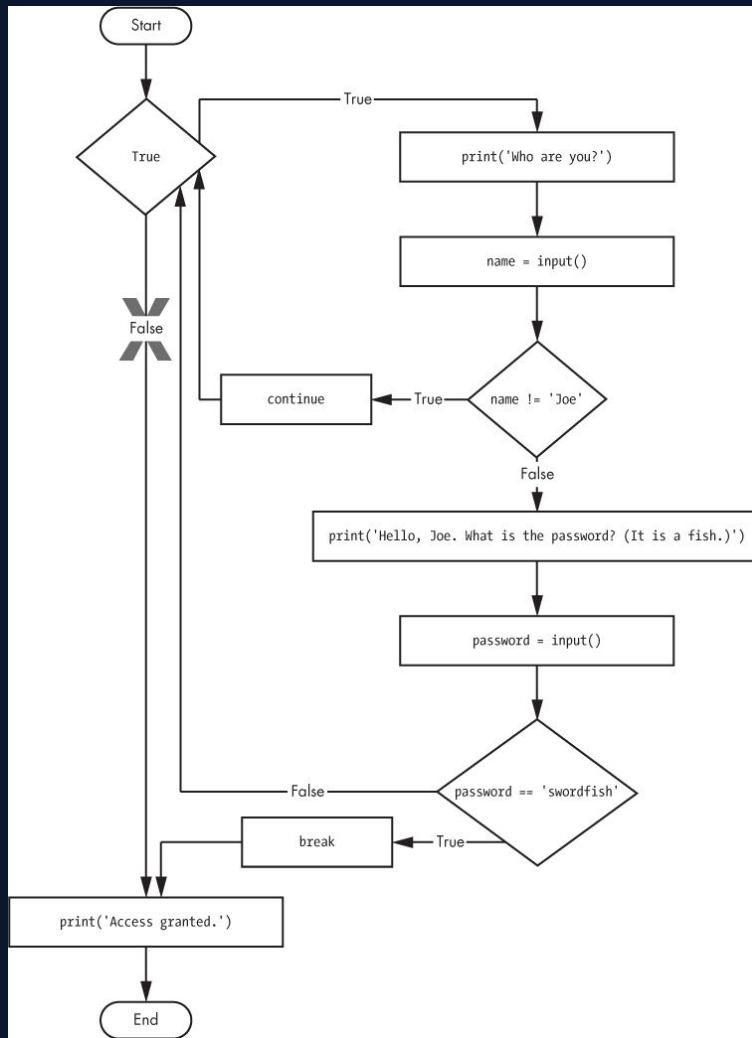
# Instrucțiunea break [1]





# Instrucțiunea *continue* [1]

La fel ca declarațiile *break*, declarațiile *continue* sunt utilizate în cadrul unor bucle. Când execuția programului ajunge la o instrucțiune *continue*, execuția programului sare imediat înapoi la începutul buclei și reevaluează starea buclei. (Aceasta este, de asemenea, ceea ce se întâmplă atunci când execuția ajunge la sfârșitul buclei.)





# Instrucțiunea *for* [1]

Bucula *while* continuă să se repete cât timp starea sa este Adevărată (care este și motivul numelui ei), dar ce se întâmplă dacă doriți să executați un bloc de cod doar de un anumit număr de ori? Puteți face acest lucru cu instrucțiunea *for* și instrucțiunea *range()*.

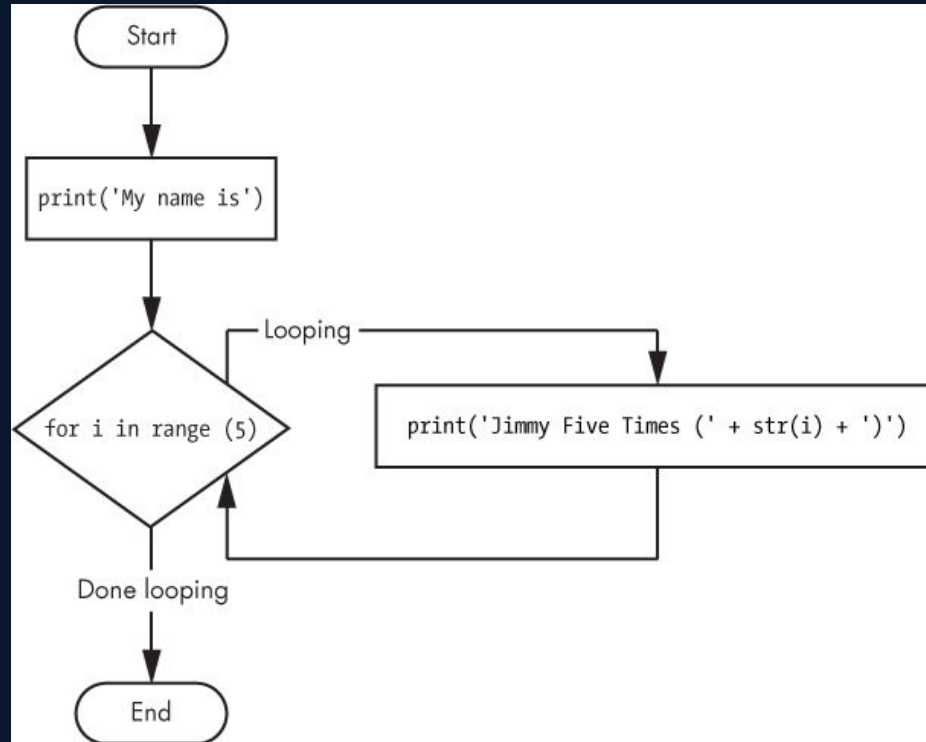


# Instrucțiunea *for* [1]

În cod, o declarație *for* arată ceva asemănător cu `for i in range(5):` și include următoarele:

- Cuvântul cheie *for*;
- Un nume de variabilă, în cazul acesta *i*;
- Cuvântul cheie *in*;
- Un apel la instrucțiunea *range()*;
- Două puncte;
- Începând cu următoarea linie, un bloc de cod indentat (numit *clauza for*)

# Instrucțiunea *for* [1]





## Argumentele de pornire, oprire și pas pentru *range()*

Unele instrucțiuni pot fi apelate cu mai multe argumente separate printr-o virgulă, iar *range()* este una dintre ele. Acest lucru vă permite să schimbați numărul întreg trecut în intervalul dintre paranteze pentru a urmări orice succesiune de numere întregi, inclusiv începând cu un număr diferit de zero.



## Argumentele de pornire, oprire și pas pentru *range()*

Instrucțiunea *range()* poate fi apelată și cu trei argumente. Primele două argumente vor fi valorile de pornire și oprire, iar al treilea va fi argumentul pasului. Pasul este cantitatea cu care variabila este mărită după fiecare iterație.





# Referințe

## 1 - Automate the Boring Stuff with Python, 2nd Edition

- by Al Sweigart
- Publisher: No Starch Press
- Release Date: November 2019
- ISBN: 9781593279929



*That's all Folks!*