

Цикли:

Циклите в програмирането ни позволяват да повтаряме едни и същи действия определен брой пъти.

Оператор *for*

for i in range(начална стойност, крайна стойност):

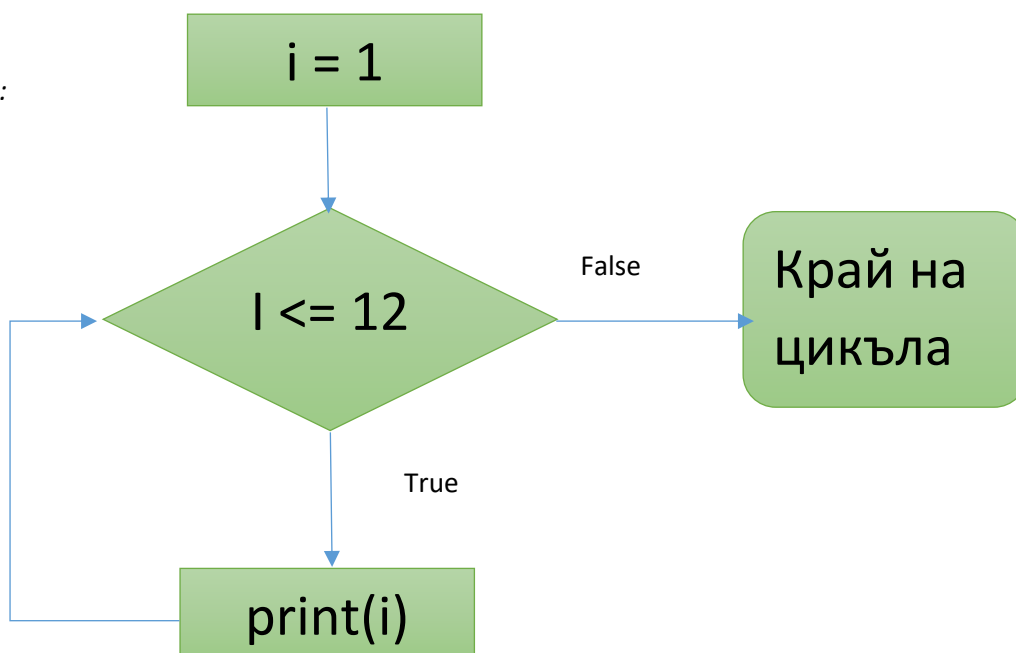
блок код

където "i" е служебна променлива, с която изразяваме текущата итерация на цикъла.

Ако имаме

for i in range(1, 13):

print(i)



Опитайте и вие, променете началната и крайната стойност, за да видите резултата. Можем да използваме **стъпка** ако искаме да променяме стойността на i със стойност различна от 1.

for i in range(2, 11, 2)

print(i)

Ще се отпечата числата от 2 до 10
със стъпка 2 (фиг.1)

```

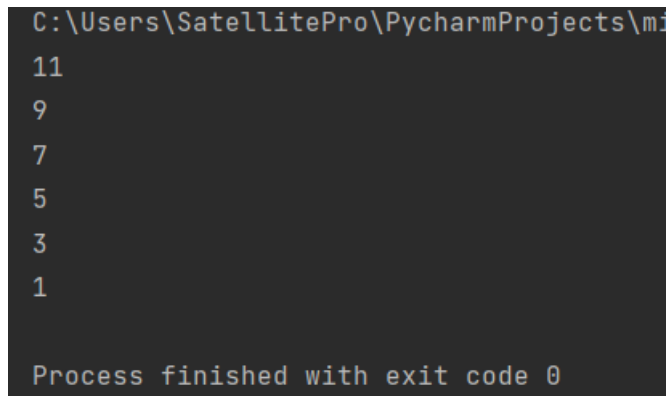
C:\Users\SatellitePro\PycharmProjects\m
2
4
6
8
10

Process finished with exit code 0
  
```

фиг.1

```
for i in range(11, 0, -2):
    print(i)
```

Ще се отпечата числата т 11 до 0
със стъпка -2 (фиг. 2)



```
C:\Users\SatellitePro\PycharmProjects\mi
11
9
7
5
3
1
Process finished with exit code 0
```

фиг. 2

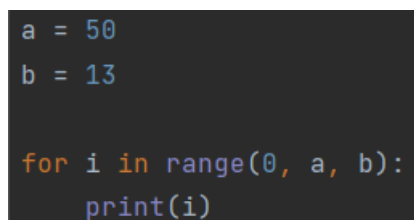
Напишете програма, която да изписва числата от 1 до N през 3.

1. Прочита цяло число n
2. Отпечатва числата от 1 до **n + 1** със стъпка 3

Примерен вход и изход:

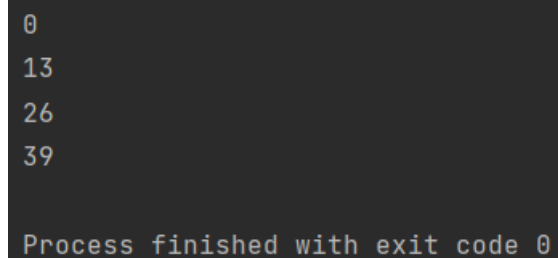
10 -> 1, 4, 7, 10

Тази задача демонстрира, че променлива може да бъде използвана за поставяне на начало, край и стъпка на for цикълът.



```
a = 50
b = 13

for i in range(0, a, b):
    print(i)
```



```
0
13
26
39
Process finished with exit code 0
```

Сега опитайте да добавите и потребителски вход към задачата.

Примерен вход и изход:

101, 25 -> 0, 25, 50, 75, 100

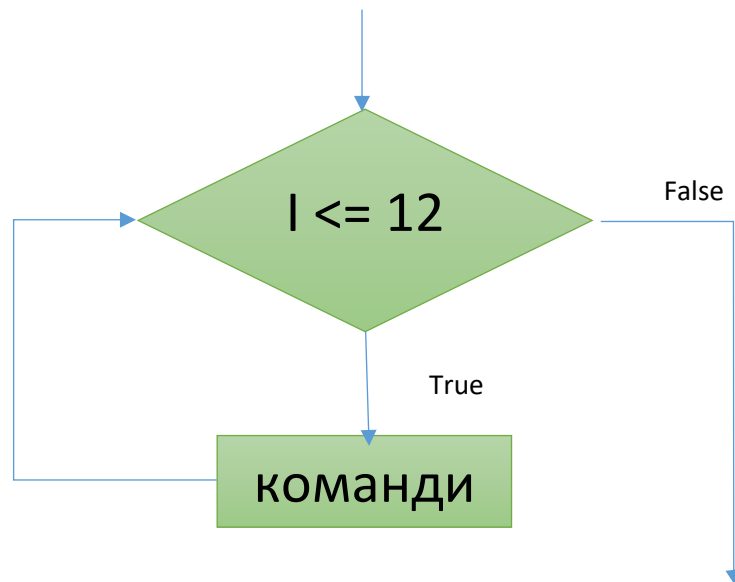
Оператор while

While цикъл – това е команда в програмирането, която казва на компютъра: „Повтаряй тези действия отново и отново, докато (while) едно определено условие е вярно“. Представете си го като аларма на телефон – тя ще продължи да звъни на всеки 5 минути, докато не натиснете бутона за спиране. В нашия проект използваме този цикъл, за да държим калкулатора „буден“, така че потребителят да може да пресмята колкото числа поиска, без програмата да се затваря сама.

Синтаксис

`while <условие>:`

`блок код`



Когато използваме оператор while има опасност да създадем безкраен цикъл, който ще ни създаде главоболия, за да избегнем това, на помощ идва операторът break, който прекъсва цикъла, той не може да съществува самостоятелно извън цикъл.

`while true:`

`print("loop")`

`if <условие>:`

`break`

```

a = 5
while True:
    if a > 10:
        break
    print("a = " + str(a))
    a += 1
  
```



```

a = 5
a = 6
a = 7
a = 8
a = 9
a = 10

Process finished with exit code 0
  
```

Към мини-проекта:

Ще вложим условната конструкция от урок 3 в while-цикъл, като за целта първо ще създадем променлива която ще контролира дали цикълът да продължи или да се прекрати. Нека да декларираме променлива *prodalji_cikal* = 'y', Нейната стойност ще може да бъде променена от потребителя в зависимост от това, дали иска да продължи с използването на програмата или не. След това създайте while цикъл, чието условие да бъде

prodalji_cikal == 'y' и изместете цялата условна конструкция по начина показан по-долу

The screenshot shows a code editor with a Python script. The script starts with a `while prodalji_cikal == 'y':` loop. Inside the loop, there's a comment in Bulgarian: `# Условните оператори, отговарящи за менюто`. Then, there's a `if izbor == 1:` block where `chislo` is input and printed. This is followed by `elif` blocks for `izbor == 2` and `izbor == 3`, and an `else:` block that prints an error message. A context menu is open over the code, showing various actions like Cut, Copy, Paste, and Run. The `Column Selection Mode` option is highlighted.

```
while prodalji_cikal == 'y':
    # Условните оператори, отговарящи за менюто
    if izbor == 1:
        chislo = input("Въведете двоично число: ")
        print(f"Избрахте преобращаване на {chislo}")
    elif izbor == 2:
        chislo = int(input("Въведете двоично число: "))
        print(f"Избрахте преобращаване на {chislo}")
    elif izbor == 3:
        chislo = input("Въведете двоично число: ")
        print(f"Избрахте преобращаване на {chislo}")
    elif izbor == 4:
        chislo = int(input("Въведете двоично число: "))
        print(f"Избрахте преобращаване на {chislo}")
    else:
        print("Грешка! Моля, изберете правилна опция.")

if izbor == 1
```

След това, натиснете **tab**, за да може while цикълът да я усвои. (фиг. 3)

```

5 while prodalji_cikal == 'y':
6     # Условните оператори, отговарящи за менюто
7     if izbor == 1:
8         chislo = input("Въведете двоично число: ")
9         print(f"Избрахте преобразуване от Двоична към Десетична за: {chislo}")
10
11     elif izbor == 2:
12         chislo = int(input("Въведете десетично число: "))
13         print(f"Избрахте преобразуване от Десетична към Двоична за: {chislo}")
14
15     elif izbor == 3:
16         chislo = input("Въведете шестнадесетично число: ")
17         print(f"Избрахте преобразуване от Шестнадесетична към Десетична за: {chislo}")
18
19     elif izbor == 4:
20         chislo = int(input("Въведете десетично число: "))
21         print(f"Избрахте преобразуване от Десетична към Шестнадесетична за: {chislo}")
22
23     else:
24         print("Грешка! Моля, изберете число между 1 и 4.")
25
26

```

Фиг. 3

Добавете и следния код на края на цикъла, за да може потребителя да смени y на n за да спре цикъла.

```

20 elif izbor == 4:
21     chislo = int(input("Въведете десетично число: "))
22     print(f"Избрахте преобразуване от Десетична към Шестнадесетична за: {chislo}")
23
24 else:
25     print("Грешка! Моля, изберете число между 1 и 4.")
26
27 # Тук даваме шанс на потребителя да смени 'y' на 'n' и да спре цикъла:
28 prodalji_cikal = input("Искате ли да продължите? (y/n): ")

```