

ლექცია 3 – ციკლები

კასკადური მინიჭება

```
a = b = c = 0  
print(a, b, c )
```

```
0 0 0
```

მრავალმხრივი მინიჭება

```
x, y = 1, 2  
print(x, y)
```

```
x = 1  
y = 2  
print(x, y)
```

```
1 2  
1 2
```

ინკრემენტაცია

```
x = 0  
x = x + 15  
print(x)
```

```
x = x + 2  
print(x)
```

```
x += 1  
print(x)
```

```
x -= 5  
print(x)
```

```
x *= 2  
print(x)
```

```
x /= 2  
print(x)
```

```
15  
17  
18  
13  
26  
13.0
```

ციკლები

for ციკლი

```
arr = [1, 2, 3, 4, 5, 6, 7]
```

```
for i in arr:  
    print(i)
```

```
1  
2  
3  
4  
5  
6  
7
```

განვიხილოთ როგორ მუშაობს for ციკლი შიგნიდან
for ციკლი ავტომატურად აკეთებს იტერაციას

```
arr1 = [1, 2, 3, 4, 5, 6, 7]  
arr2 = iter(arr1)
```

```
print(next(arr2))  
print(next(arr2))  
print(next(arr2))  
print(next(arr2))  
print(next(arr2))  
print(next(arr2))  
print(next(arr2))
```

```
print(next(arr2))
```

```
1  
2  
3  
4  
5  
6  
7
```

Traceback (most recent call last):

File "d:\PyApp\2024-2025\BPWSO-16-PM\ლექცია 3 - ციკლები\main.py", line 63, in <module>

```
    print(next(arr2))
```

```
          ^^^^^^^^^
```

StopIteration

იტერაციამ კვლავ რომ იმუშაოს, საჭიროა `arr2 = iter(arr1)` ჩავწერო

```
arr2 = iter(arr1)
print(next(arr2))
print(next(arr2))
print(next(arr2))
```

```
arr2 = iter(arr1)
print(next(arr2))
print(next(arr2))
print(next(arr2))
print(next(arr2))
print(next(arr2))
```

```
1
2
3
1
2
3
4
5
```

ფუნქცია `range()`

სინტაქსი:

`range(start, stop, step)`

1-დან 10-მდე 1-ის ბიჯით $\Leftrightarrow \text{stop} = 10 - 1 = 9$

<code>range(stop)</code>	<code>range(start, stop)</code>	<code>range(start, stop, step)</code>
<code>[0, stop-1]</code>	<code>[start, stop-1]</code>	<code>[start, stop-1, step]</code>
<code>[0, 10-1]</code>	<code>[5, 13-1]</code>	<code>[23, 31-1, 1]</code>
<code>[10-1]</code>		<code>[1, 10, 2]</code>
		<code>[15, 3, -1]</code>

```
for i in range(10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

```
for i in range(10):  
    print(i, end=' ')
```

```
0 1 2 3 4 5 6 7 8 9
```

```
for i in range(1, 11):  
    print(i, end=' ')
```

```
print()
```

```
1 2 3 4 5 6 7 8 9 10
```

```
for i in range(4, 15, 3):  
    print(i, end=' ')
```

```
print()
```

```
4 7 10 13
```

დავბეჭდოთ 0-დან 10-მდე ლუწი რიცხვები:

```
for i in range(0, 10, 2):  
    print(i, end=' ')
```

```
print()
```

```
0 2 4 6 8
```

დავბეჭდოთ 1-დან 10-მდე კენტი რიცხვები:

```
for i in range(1, 10, 2):  
    print(i, end=' ')
```

```
print()
```

```
1 3 5 7 9
```

დავბეჭდოთ 1-დან 15-მდე ყველა რიცხვი 3-ის ბოჯით:

```
for i in range(1, 15, 3):  
    print(i, end=' ')
```

```
print()
```

```
1 4 7 10 13
```

მოცემულია სია. დავბეჭდოთ სიის ელემენტების 2-ზე ნამრავლი

```
arr = [15, 21, 7, 26, 32, 8, 11]
```

```
for i in arr:  
    print(i * 2)
```

```
print(arr)
```

```
30  
42  
14  
52  
64  
16  
22  
[15, 21, 7, 26, 32, 8, 11]
```

როგორ შევცვალოთ arr სიის ელემენტები?

```
arr = [15, 21, 7, 26, 32, 8, 11]
```

```
for i in range(len(arr)):    # for i in range(7):  
    arr[i] *= 2
```

```
print(arr)
```

```
[30, 42, 14, 52, 64, 16, 22]
```

ოპერატორები break და continue

break ოპერატორი

```
for num in range(1, 11):  
    print(num)  
  
    if num == 5:  
        break
```

```
1  
2  
3  
4  
5
```

რა განსხვავებაა ზედა და ქვედა კოდს შორის?

```
for num in range(1, 11):  
    if num == 5:  
        break  
  
    print(num)
```

```
1 2 3 4
```

რას დაბეჭდავს და რა შემთხვევაში გაჩერდება ციკლი?

```
for i in range(1, 20):  
    print(i)  
    break
```

```
1
```

```
for i in range(1, 20):  
    break  
    print(i)
```

ლუწი რიცხვების დაბეჭდვა range() ფუნქციის გამოყენების გარეშე:

```
for i in range (1, 11):  
    if i % 2 == 0:  
        print(i)
```

```
2  
4  
6  
8  
10
```

continue ოპერატორი

დავბეჭდოთ ლუწი რიცხვები continue-ს გამოყენებით:

```
for i in range(20):  
    if i % 2 != 0:    # if i % 2 == 1  
        continue  
  
    print(i, end=' ')  
  
print()
```

```
0 2 4 6 8 10 12 14 16 18
```

რა დაიბეჭდება?

```
for i in range(10):  
    continue  
  
    print(i)  
    print(i ** 2)
```

ამ შემთხვევაში?

```
for i in range(10):  
    print(i)  
  
    continue  
  
    print(i ** 2)
```

```
0  
1  
2
```

```
3
4
5
6
7
8
9
```

დავბეჭდოთ 1-დან 20-ის ჩათვლით რიცხვების ჯამი 10-სა და 11-ის გამოტოვებით:

```
s = 0
```

```
for i in range(1, 21):
    if i == 10 or i == 11:
        continue
```

```
s += i
```

```
print(s)
```

```
189
```

continue-ს გამოყენებით დავბეჭდოთ რიცხვები 1-დან 20-ის ჩათვლით, გარდა რიცხვებისა 10-დან 15-ის ჩათვლით

```
for i in range(1, 21):
    if 10 <= i <= 15:    # if i >= 10 and i <= 15:
        continue
```

```
print(i, end=' ')
```

```
print()
```

```
1 2 3 4 5 6 7 8 9 16 17 18 19 20
```

იგივე ამოცანა continue-ს გარეშე

```
for i in range(1, 21):
    if i < 10 or i > 15:
        print(i, end=' ')
```

```
print()
```

```
1 2 3 4 5 6 7 8 9 16 17 18 19 20
```



```

for i in range(1, 21):
    if not (i < 10 or i > 15):
        print(i, end=' ')

print()

```

```
10 11 12 13 14 15
```

შეიძლება range() ფუნქციის გამოყენებაც:

```

for i in range(1, 21):
    if i in range(10, 16):
        print(i, end=' ')

print()

```

```
10 11 12 13 14 15
```

```

for i in range(1, 21):
    if i not in range(10, 16):
        print(i, end=' ')

print()

```

```
1 2 3 4 5 6 7 8 9 16 17 18 19 20
```

**continue-ს და break-ის ერთდროულად გამოყენება
რა დაიბეჭდება?**

```

for i in range(1, 21):
    if 10 <= i <= 15:
        continue

    if i == 18:          # if i == 8:
        break

    print(i, end= ' ')

print("END")

```

```
1 2 3 4 5 6 7 8 9 16 17 END
```

დაბეჭდოთ 1-დან 100-ის ჩათვლით კენტი რიცხვები

```
for i in range(1, 101):  
    if i % 2 == 1:  
        print(i, end=' ')
```

```
print()
```

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63  
65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

დავამატოთ ერთი პირობა: არ დაბეჭდოს რიცხვები [50..70] დიაპაზონში

```
for i in range(1, 101):  
    if i % 2 == 1 and not (50 <= i < 70):  
        # if i % 2 == 1 and (i < 50 or i > 70):  
        print(i, end=' ')
```

```
print()
```

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 71 73 75 77 79 81 83  
85 87 89 91 93 95 97 99
```

გავაგრძელოთ წინა პირობა და 87-ის მიღწევისას ციკლი გაჩერდეს

```
for i in range(1, 101):  
    if i == 87:  
        break  
    elif i % 2 == 1 and not (50 <= i <= 70):  
        print(i, end=' ')
```

```
print()
```

```
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 71 73 75 77 79 81 83  
85
```

while ციკლი

სინტაქსი:

```
while <პირობა>:
```

```
    ოპერატორი 1
```

```
    ოპერატორი 2
```

```
    ...
```

```
    ოპერატორი N
```

უსასრულო ციკლი

```
while True:    # while 2 > 1:
```

```
    print("*")
```

დაგეგმილი ფიფქები (მართვითა სამკუთხედი)

```
i = 10
```

```
while True:
```

```
    if i < 1:
```

```
        break
```

```
    print('*' * i)
```

```
    i -= 1    # i = i - 1
```

```
*****
*****
*****
*****
*****
*****
****
***
**
*
```

გავაკეთოთ ფიფქების ანიმაციური გამოტანა
ყოველი სტრიქონის დაყოვნებით გამოტანა

```
import time
```

```
i = 10
```

```
while True:
    if i < 1:
        break

    print('*' * i)

    time.sleep(0.25)

    i -= 1    # i = i - 1
```

დაიბეჭდება ფიფქების სტრიქონი 0.25 წამიანი დაყოვნებით

უსასრულო ციკლი. **break** და **continue**

```
import time
```

```
i = 10
```

```
while True:
    if i < 1:
        break
    elif i < 4:
        continue

    print(i)
    print("sleep\n")

    time.sleep(0.25)

    i -= 1
```

```
10
sleep
9
sleep
8
sleep
7
sleep
```

```
6
sleep
```

```
5
sleep
```

```
4
sleep
```

Traceback (most recent call last):

```
File "e:\PyApp\2024-2025\BPWSO-16-PM\ლექცია 3 - ციკლები\main.py", line 420, in <module>
    continue
```

KeyboardInterrupt

^C

აქ i ცვლადი გახდება 3-ის ტოლი

```
elif i < 4:
    continue
```

elif-ის პირობა მუდმივად ჭეშარიტია (i-ის მნიშვნელობა არ იცვლება, სულ 3-ის ტოლია). ამიტომ continue-ს წინ უნდა ჩავსვათ: i -= 1

უსასრულო ციკლი. break და continue

```
import time
```

```
i = 10
```

```
while True:
    if i < 1:
        break
    elif i < 4:
        i -= 1
        continue
```

```
print(i)
print("sleep\n")
```

```
time.sleep(0.25)
```

```
i -= 1
```

```
10
sleep
```

```
9
sleep
```

```
8
sleep

7
sleep

6
sleep

5
sleep

4
sleep
```

dry (Don't Repeat Yourself)-ს პრინციპი რომ არ დაირღვეს შეიძლება, ასე ჩავწეროთ:

```
import time
```

```
i = 11
```

```
while True:
```

```
    i -= 1
```

```
    if i < 1:
```

```
        break
```

```
    elif i < 4:
```

```
        continue
```

```
    print(i)
```

```
    print("sleep\n")
```

```
    time.sleep(0.25)
```

```
10
sleep

9
sleep

8
sleep

7
sleep

6
sleep

5
```

```
sleep
```

```
4
```

```
sleep
```

while ციკლის გაჩერება **break**-ის გარეშე

```
from time import sleep
```

```
i = 10
```

```
while i > 0:
```

```
    print(i)
```

```
    sleep(0.35)
```

```
    i -= 1
```

```
10
```

```
9
```

```
8
```

```
7
```

```
6
```

```
5
```

```
4
```

```
3
```

```
2
```

```
1
```

while .. else

```
from time import sleep
```

```
i = 10
```

```
while i > 0:
```

```
    print(i)
```

```
    sleep(0.35)
```

```
    i -= 1
```

```
else:
```

```
    print("While end")
```

```
print("Script end")
```

```
10
```

```
9
```

```
8
```

```
7
```

```
6
5
4
3
2
1
While end
Script end
```

ახლა ვნახოთ, როდესაც ციკლიდან გამოვალ **brake**-ით:

```
from time import sleep
```

```
i = 10
```

```
while i > 0:
```

```
    if i < 2:
```

```
        break
```

```
    print(i)
```

```
    sleep(0.35)
```

```
    i -= 1
```

```
else:
```

```
    print("While end")
```

```
print("Script end")
```

```
10
9
8
7
6
5
4
3
2
Script end
```


გამოვიცნოთ კომპიუტერის მიერ ჩაფიქრებული რიცხვი

```
from random import randint
```

```
num = randint(1, 100)
```

```
i = 1
```

```
print("Was guessed a number between 1 and 100. Guess it...\n")
```

```
while True:
```

```
    guess = int(eval(input(f"Step #{i}. Your guess: ")))
```

```
    if guess == num:
```

```
        print(f"You guessed a number! It was {num}.")
```

```
        break
```

```
    elif guess > num:
```

```
        print("Too great")
```

```
    else:
```

```
        print("Too less")
```

```
    print()
```

```
    i += 1
```

```
print("\nGame Over!")
```

Was guessed a number between 1 and 100. Guess it...

Step #1. Your guess: 50

Too less

Step #2. Your guess: 75

Too less

Step #3. Your guess: 87

Too less

Step #4. Your guess: 94

Too great

Step #5. Your guess: 90

Too great

Step #6. Your guess: 88

You guessed a number! It was 88.

Game Over!

ოგვე პროგრამა **break**-ის გარეშე და **else**-ის გამოყენებით:

```
from random import randint
```

```
num = randint(1, 100)
```

```
i = 1
```

```
print("Was guessed a number between 1 and 100. Guess it...\n")
```

```
guess = 0
```

```
while guess != num:
```

```
    guess = int(eval(input(f"Step #{i}. Your guess: ")))
```

```
    if guess > num:
```

```
        print("Too great")
```

```
    else:
```

```
        print("Too less")
```

```
    print()
```

```
    i += 1
```

```
else:
```

```
    print(f"You guessed a number! It was {num}.")
```

```
print("\nGame Over!")
```

```
Was guessed a number between 1 and 100. Guess it...
```

```
Step #1. Your guess: 50
```

```
Too less
```

```
Step #2. Your guess: 75
```

```
Too less
```

```
Step #3. Your guess: 87
```

```
Too great
```

```
Step #4. Your guess: 82
```

```
Too great
```

```
Step #5. Your guess: 79
```

```
Too great
```

```
Step #6. Your guess: 76
```

```
You guessed a number! It was 76.
```

```
Game Over!
```