

Petlje

Kako da računar ponovi isti posao, a mi napišemo samo nekoliko redova

Danas pravimo: igru „Pogodi broj“ i figure od znakova



Tri pitanja za zagrevanje

1. prošla tema

```
x = 10
if x > 5:
    print("A")
elif x > 8:
    print("B")
```

Ispis: A

2. tipovi podataka

```
print(7 // 2)
```

Ispis: 3

3. mešano

```
tekst = ""
if tekst:
    print("pun")
else:
    print("prazan")
```

Ispis: prazan

Pet novih reči za danas

petlja

deo programa koji se ponavlja

telo petlje

uvučeni redovi koji se ponavljaju

prolaz (iteracija)

jedno izvršavanje tela petlje

uslov petlje

pitanje „da li nastavljamo?“ koje se proverava pre svakog prolaza

brojač

promenljiva koja broji prolaze, najčešće se zove i

Petlja zamenjuje ponavljanje istog koda

Bez petlje: 5 redova

```
print("Zdravo!")  
print("Zdravo!")  
print("Zdravo!")  
print("Zdravo!")  
print("Zdravo!")
```

Sa petljom: 2 reda

```
for i in range(5):  
    print("Zdravo!")
```

ISTI ISPIS U OBA SLUČAJA

```
Zdravo!  
Zdravo!  
Zdravo!  
Zdravo!  
Zdravo!
```

Python ima dve petlje: while i for

while

ponavlja dok je uslov tačan

Ne znamo unapred koliko puta.

Primer: pitaj igrača za broj dok ne pogodi.

for

ponavlja za svaki element
kolekcije

Uz range ponavlja zadati broj puta.

Primer: za svaku boju u paleti nacrtaj krug.

Petlja while

Ponavljaj dok je uslov tačan

while ponavlja telo dok je uslov tačan

```
i = 2
while i < 6:
    print(i)
    i += 1
print("Kraj")
```

ISPIS

```
2
3
4
5
Kraj
```

- Uslov se proverava pre svakog prolaza.
- Ako uslov nije tačan već na početku, telo se ne izvrši nijednom.
- `i += 1` je kraći zapis za `i = i + 1`.

Šta će se desiti ako zaboravimo `i += 1`?

```
i = 2
while i < 6:
    print(i)
    print("Kraj")
```

Prvo razmislite sami, pa glasajte.

A

2
3
4
5
Kraj

B

2
Kraj

C

tačno

2
2
2
... bez kraja

D

NameError

Čudovište: beskrajna petlja



Kako izgleda

Program ne staje: ispis teče bez kraja ili prozor ne reaguje.

Zašto se pojavljuje

Uslov petlje nikad ne postaje netačan. Najčešće je zaboravljeno menjanje brojača.

Kako se pobeđuje

Prekid sa Ctrl+C (u Thonnyju dugme Stop). Zatim proveriti da li se brojač menja i da li ide ka kraju.

break prekida petlju odmah

```
i = 1
while i < 10:
    print(i)
    if i == 5:
        break
    i += 1
print("Kraj")
```

ISPIS

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

Uslov $i < 10$ je i dalje tačan

ali break izlazi iz petlje i program nastavlja od prvog reda posle nje.

continue preskače ostatak jednog prolaza

```
i = 1
while i < 10:
    i += 1
    if i == 5:
        continue
    print(i)
print("Kraj")
```

ISPIS

```
2
3
4
6
7
8
9
10
Kraj
```

Broj 5 se ne ispisuje

a petlja ide dalje sve do 10.

Šta ispisuje ovaj program?

```
i = 0
while i < 5:
    i += 1
    if i == 3:
        continue
    print(i, end=" ")
```

Pažljivo: gde se nalazi `i += 1`?

A

1 2 3 4 5

B

tačno

1 2 4 5

C

1 2

D

1 2 4

else uz petlju radi samo kad nije bilo break

Petlja završi sama

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("Kraj")
```

ISPIS

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

Petlju prekine break

```
i = 1
while i < 6:
    print(i)
    i += 1
    if i == 5:
        break
else:
    print("Kraj")
print("Posle kraja")
```

ISPIS

```
1
2
3
4
Posle kraja
```

while True: petlja kojoj sami biramo izlaz

```
zbir = 0
while True:
    broj = int(input("Unesite broj: "))
    if broj == 0:
        break
    if broj % 5 != 0:
        continue
    zbir += broj
print("Zbir je", zbir)
```

ISPIS

```
Unesite broj: 10
Unesite broj: 7
Unesite broj: 25
Unesite broj: 0
Zbir je 35
```

Telo se izvrši bar jednom.

Izlaz iz petlje je tamo gde stoji break.
Sabiraju se samo 10 i 25.

Pravimo igru „Pogodi broj“

Cilj: za 15 minuta imamo igru u kojoj računar zamisli broj, a mi ga pogađamo.

- 1 Računar zamisli broj: `random.randint(1, 20)`
- 2 `while True`: pitamo igrača za broj
- 3 Grananje: „Više!“, „Manje!“ ili pogodak
- 4 Na pogodak izlazimo iz petlje sa `break`
- 5 Dodatak: brojimo pokušaje

KAKO IZGLEDA IGRA

Tvoj broj: 10

Manje!

Tvoj broj: 5

Više!

Tvoj broj: 8

Pogodak iz 3 pokušaja!

Ceo program „Pogodi broj“

```
import random

tajni = random.randint(1, 20)
pokusaji = 0

while True:
    broj = int(input("Tvoj broj: "))
    pokusaji += 1
    if broj < tajni:
        print("Više!")
    elif broj > tajni:
        print("Manje!")
    else:
        print("Pogodak iz", pokusaji, "pokušaja!")
        break
```

while True

igra traje dok ne pogodimo

break

kraj igre posle pogotka

pokusaji += 1

brojač pokušaja

pass čuva mesto za kod koji ćemo tek napisati

```
n = int(input("Unesite broj: "))  
  
if n > 1:  
    pass # ovo dopisujemo kasnije  
else:  
    print("Niste uneli ispravan broj")
```

- pass je naredba koja ne radi ništa.
- Koristimo je kad blok mora da postoji, a još ne znamo šta će u njemu biti.
- Bez nje program ne može da se pokrene: IndentationError.

Petlja for

Ponovi za svaki element

for ponavlja blok za svaki element liste

```
for boja in ["crvena", "zelena", "plava"]:  
    print("Slikam", boja)
```

1. prolaz

boja = "crvena"



Slikam crvena

2. prolaz

boja = "zelena"



Slikam zelena

3. prolaz

boja = "plava"



Slikam plava

range daje niz brojeva bez gornje granice

range(kraj)

od 0 do kraj – 1

```
for x in range(5):  
    print(x, end=" ")
```

0 1 2 3 4

range(početak, kraj)

od početka do kraj – 1

```
for x in range(2, 6):  
    print(x, end=" ")
```

2 3 4 5

range(početak, kraj, korak)

svaki korak-ti broj

```
for x in range(1, 30, 5):  
    print(x, end=" ")
```

1 6 11 16 21 26

Šta ispisuje ovaj program?

```
boje = ["crvena", "plava", "žuta"]  
for i in range(1, len(boje)):  
    print(i, boje[i])
```

Pomoć: len(boje) je broj elemenata liste.

A

0 crvena
1 plava
2 žuta

B

tačno

1 plava
2 žuta

C

1 crvena
2 plava

D

IndexError

A ovaj, sa negativnim korakom?

```
for i in range(10, 0, -3):  
    print(i, end=" ")
```

Korak može da bude i negativan: tada se broji unazad.

A tačno
10 7 4 1

B
10 7 4 1 -2

C
10 7 4

D
ništa se ne
ispisuje

Petlja for sa grananjem: zbir brojeva deljivih sa 5

```
n = int(input("Unesite n: "))
zbir = 0
for i in range(1, n + 1):
    if i % 5 == 0:
        zbir += i
print(zbir)
```

ISPIS

Unesite n: 20
50

5 + 10 + 15 + 20 = 50

Petlja prolazi kroz sve brojeve od 1 do n, a grananje bira one koji su deljivi sa 5.

Korak u range može da zameni grananje

```
n = int(input("Unesite n: "))  
for i in range(2, n + 1, 2):  
    print(i, end=" ")
```

ISPIS

```
Unesite n: 10  
2 4 6 8 10
```

Umesto provere $i \% 2 == 0$

počinjemo od 2 i idemo korakom 2. Neparni brojevi se i ne pojavljuju.

Petlja u petlji

Redovi i kolone, kao pikseli na slici

Petlja u petlji crta redove i kolone

```
for red in range(3):  
    for kolona in range(5):  
        print("*", end=" ")  
    print()
```

ISPIS

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

- Spoljašnja petlja: 3 reda.
- Unutrašnja petlja: 5 zvezdica u svakom redu, ukupno $3 \times 5 = 15$.
- `print()` bez argumenata samo prelazi u novi red.

Množenje stringa brojem pravi red istih znakova

Trougao

```
n = 5
for red in range(1, n + 1):
    print("*" * red)
```

ISPIS

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

Tekstualni gradijent

```
for znak in ".*#@":
    print(znak * 24)
```

ISPIS

```
.....
:::::::::
*****
#####
@@@@@@@@@@@@@@@@@@@@
```

Pronađi grešku

```
for i in 5:  
    print(i)
```

Kako glasi ispravan prvi red?

PORUKA O GREŠCI

```
Traceback (most recent call last):  
  File "petlja.py", line 1, in <module>  
    for i in 5:  
            ^  
TypeError: 'int' object is not iterable
```

line 1

red u kom je greška

TypeError

vrsta greške: pogrešan tip vrednosti

not iterable

broj nije kolekcija i ne može se proći kroz njega

Izaberite nivo i počnite

LEVEL 1

Odbrojavanje

Ispišite brojeve od 10 do 1, svaki u svom redu, a zatim reč Start!

Pomoć: range može da broji unazad.

LEVEL 2

Tablica množenja

Ispišite tablicu množenja 5×5 u pet redova i pet kolona.

Pomoć: petlja u petlji i `print(..., end="\t")`

LEVEL 3

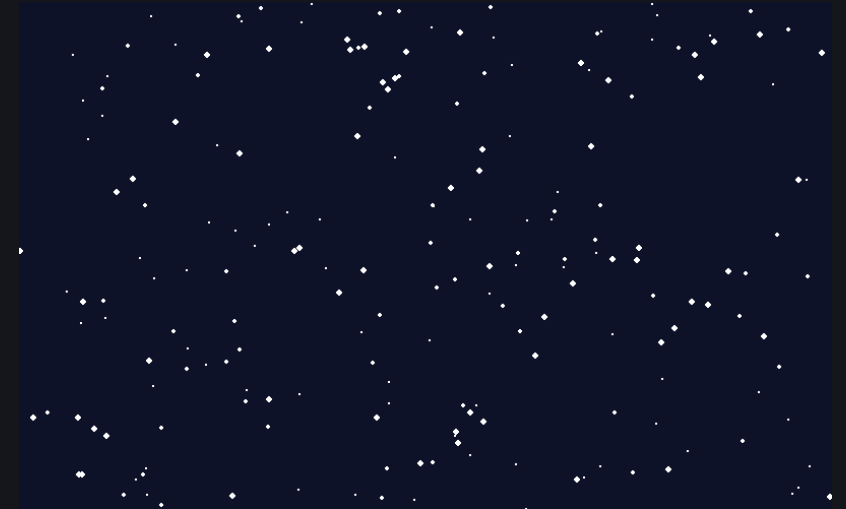
Romb od znaka +

Za neparan broj n nacrtajte romb kao na naslovnom slajdu.

Pomoć: koliko razmaka ide ispred znakova u svakom redu?

Ista petlja kasnije crta 200 zvezda

```
for i in range(200):  
    x = random.randint(0, SIRINA)  
    y = random.randint(0, VISINA)  
    r = random.randint(1, 3)  
    pg.draw.circle(ekran, "white", (x, y), r)
```



U temi o PyGame-u petlja for crta po prozoru umesto u konzoli. Svaki prolaz daje jednu zvezdu na slučajnom mestu.

Rezime

while

```
while i < 6:  
    i += 1
```

Ponavlja dok je uslov tačan. Ne zaboravite brojač.

for sa range

```
for i in range(1, 6):  
    print(i)
```

Gornja granica se ne uključuje.

break i continue

```
break      # izađi  
continue   # preskoči
```

break završava petlju, continue samo prolaz.

petlja u petlji

```
for red in range(3):  
    for k in range(5):
```

Unutrašnja petlja ide ceo put za svaki red.

Pitanje pre odlaska

```
for i in range(2, 8, 2):  
    print("zdravo")
```

Koliko puta se ispisuje reč zdravo?

Odgovor pošaljite pre nego što izađete. Jedan broj je dovoljan.



Literatura

- Python dokumentacija, Tutorial, poglavlje „More Control Flow Tools“: docs.python.org/3/tutorial
- Wentworth, P., Elkner, J., Downey, A. B., Meyers, C. How to Think Like a Computer Scientist: Learning with Python 3 (besplatna knjiga)
- Sweigart, A. Invent Your Own Computer Games with Python, 4. izdanje, No Starch Press (besplatno na inventwithpython.com)

Dodatna literatura i zanimljivi sajtovi

Petlja: Priručnik za Python

petlja.org · Priručnik za Python

Besplatan priručnik na srpskom, sa zadacima po nivoima težine.

Python Tutor

pythontutor.com

Crta korak po korak šta program radi i šta je u kojoj promenljivoj.

W3Schools: Python

w3schools.com/python

Kratke lekcije sa primerima koje menjate i pokrećete u pregledaču.

Exercism: Python

exercism.org/tracks/python

Besplatni zadaci za vežbu, poređani po temama.

Hvala!

Pitanja su uvek dobrodošla: na času, u četvrtu ili međuvremeno.

Slajdovi, primeri i kartica za podsećanje su na sajtu predmeta.

PREDAVANJA

dr Staša Vujičić Stanković

stasa.vujicic.stankovic@math.rs

www.matf.bg.ac.rs/~stasa.vujicic.stankovic

VEŽBE

Nevena Ćirić

poincare.matf.bg.ac.rs/~nevena.ciric/p1uu.html