



Distance Learning System



Rad sa petljama

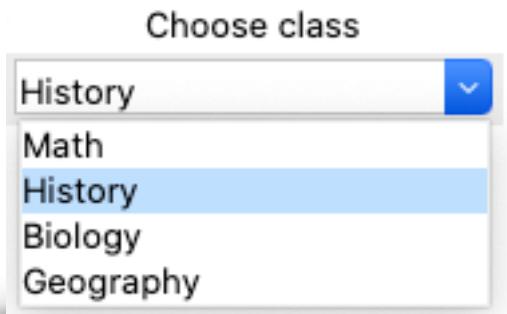
Python and programming fundamentals

Petlje u programiranju

- U programiranju, petlja nam omogućava da određeni kod izvršimo više puta
- Tokom izvršavanja, postoji mogućnost modifikacije vrednosti promenljivih
- Trajanje petlje se kontroliše nekom kontrolnom promenljivom ili iteratorom
- Jedno izvršavanje bloka petlje, naziva se **iteracija**

Gde srećemo petlje

- U **padajućim menijima**, svaka stavka se iscrtava istom tehnikom, ali se vrednost menja



A screenshot of a sign-up form titled "Sign Up" with the sub-instruction "It's quick and easy." The form includes fields for "First name", "Last name", "Email or phone number", and "Password". Below these is a date input field showing "11 1994" with a calendar icon. At the bottom are gender selection buttons: "Female", "Male", and "Custom". A question mark icon is also present.

Gde srećemo petlje

- U web prodavnici, gde god postoji listanje proizvoda ono je realizovano pomoću petlje

Showing 1 - 7 of 7 results

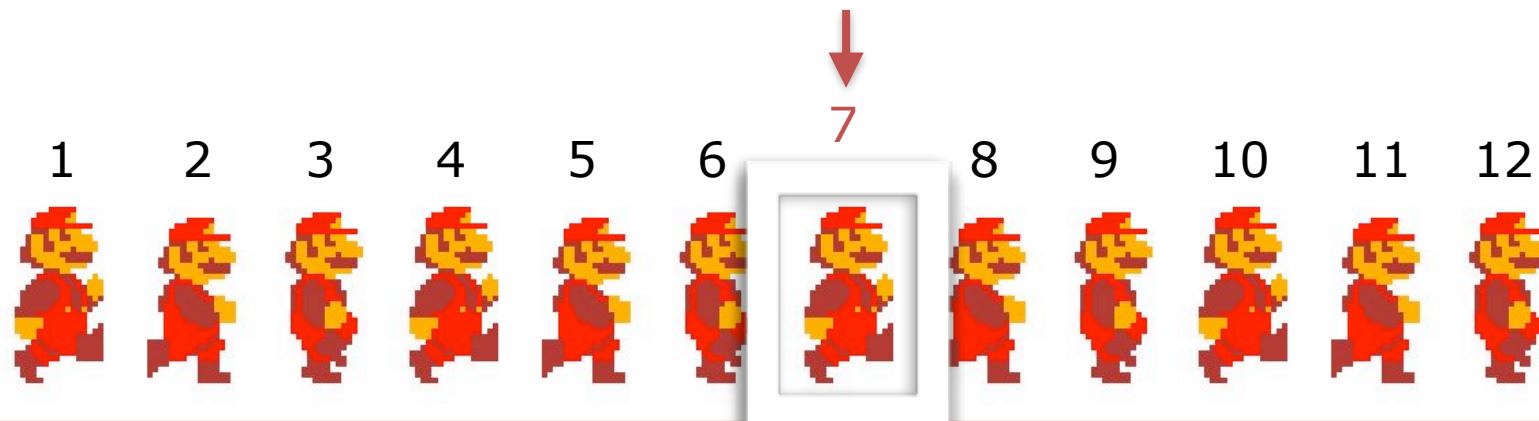
Sort by
Featured

Reset All
PRODUCT TYPE <input type="checkbox"/> Sets [7]
AGE <input type="checkbox"/> 3 - 5 [3] <input type="checkbox"/> 6 - 8 [7] <input type="checkbox"/> 9 - 11 [7] <input type="checkbox"/> 12+ [7]
PRICE <input type="checkbox"/> \$0 - \$25 [2] <input type="checkbox"/> \$25 - \$50 [3] <input type="checkbox"/> \$75 - \$100 [2]
FEATURED
THEME
PIECE COUNT
AVAILABILITY
RATING

 New Arendelle Castle Village 41167 \$79.99 Add to Bag Add to Wishlist	 New Elsa's Magical Ice Palace 43172 ★★★★★ \$79.99 Backorder Add to Wishlist	 New Enchanted Treehouse 41164 \$49.99 Add to Bag Add to Wishlist
 New Elsa's Jewelry Box Creation 41168 \$39.99 Add to Bag Add to Wishlist	 New Elsa's Wagon Adventure 41166 \$29.99 Add to Bag Add to Wishlist	 New Anna's Canoe Expedition 41165 \$19.99 Add to Bag Add to Wishlist

Gde srećemo petlje

- Čak ni najjednostavnija igra ne može funkcionisati bez petlje



Gde srećemo petlje

- Kompleksnije igre, sa animacijom, sadrže mnoštvo petlji



Vrste petlji u Python-u

- Python poznaje dve vrste petlji

for

- Primjenjuje se nad kolekcijama
- Zna se broj iteracija
- Nije potrebno kontrolisati je ručno
- Manja je opasnost od zaključavanja (mrtve petlje)

while

- Ne zahteva kolekciju
- Ne zna se broj iteracija
- Mora se kontrolisati "ručno"
- Veća opasnost od zaključavanja (mrtve petlje)

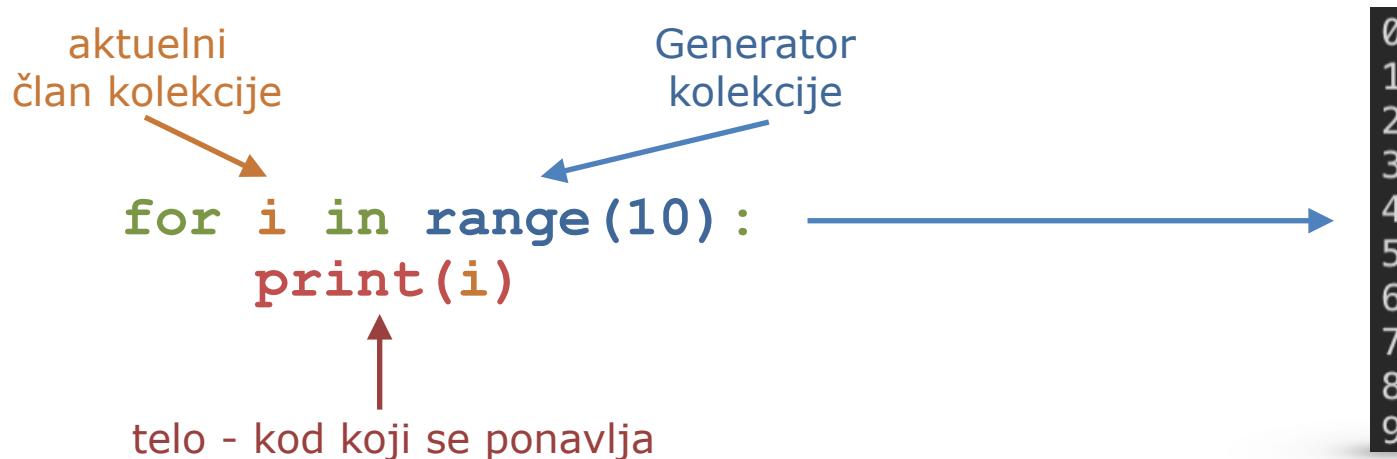
for petlja

```
for i in [1,2,3]:  
    print(i)
```

- Primjenjuje se nad **kolekcijama**
- Zna se broj iteracija
- Nije potrebno kontrolisati je ručno
- Manja je opasnost od zaključavanja (mrtve petlje)

for petlja

- For petlja zahteva kolekciju podataka da bi mogla da se izvrši
- Ova kolekcija može već postojati u programu, ali je takođe možemo generisati samo u svrhu izvođenja petlje



Vežba 1 (ppf-ex06 years.py)

- U programu su definisane promenljive startDate i endDate koje predstavljaju početnu i krajnju godinu
- Potrebno je kreirati program koji će na osnovu ove dve vrednosti prikazati listu dozvoljenih godina
- Aplikacija treba da ima izlaz sličan sledećem

Napomena:

Komanda range() može se koristiti sa parametrima. Na primer:

```
range(1,10)
```

Generiše kolekciju od broja 1 do broja 9

```
***** Allowed years *****  
2010  
2011  
2012  
2013  
2014  
2015  
*****
```

Vežba 2 (ppf-ex06 multable.py)

- Korisnik unosi broj
- Potrebno je kreirati tablicu množenja do 3 za brojeve od 1 do vrednosti unetog broja
- Na primer, ako korisnik unese broj 5, tablica množenja će biti kao na slici
-

Napomena:

Da sprečimo štampanje novog reda u komandi print, koristimo parametar end:

```
print('Number is: ', 5, end='')
```

Za tab oznaku unutar štampanog teksta, koristimo kod \t

Enter number: 5		
1	2	3

1	2	3
2	4	6
3	6	9
4	8	12
5	10	15

Vežba 3 (ppf-ex06 rect.py)

- U programu se nalaze dve promenljive: **h** i **w**
- Potrebno je, pomoću jedne petlje,
iscrtati pravougaonik
širine **w** i visine **h**.

```
w = 10  
h = 5
```

10

	#####	#####
#		#
#		#
#		#
	#####	#####

5

Ugnježđena **for** petlja (petlja u petlji)

- Unutar jedne petlje se može naći neograničena količina koda, koja može takođe sadržati jednu ili više for petlji
- U sledećem primeru, spoljašnja petlja se izvršava 10 puta, a unutrašnja takođe 10 puta. Ovo znači da će se telo unutrašnje petlje izvršiti 100 puta

```
for i in range(10):  
    for j in range(10):  
        print("Hello")
```

Ugnježđena **for** petlja (petlja u petlji)

- Prilikom izvršavanja petlje u petlji, treba voditi računa o izloženim promenljivima

Ispравно

```
for i in range(10):  
    for j in range(10):  
        print("Hello")
```

Može biti problem

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

Vežba 4 (ppf-ex06 triangle.py)

- U aplikaciji postoji promenljiva H
- Potrebno je, pomoću petlje, kreirati trougao kao na slici desno

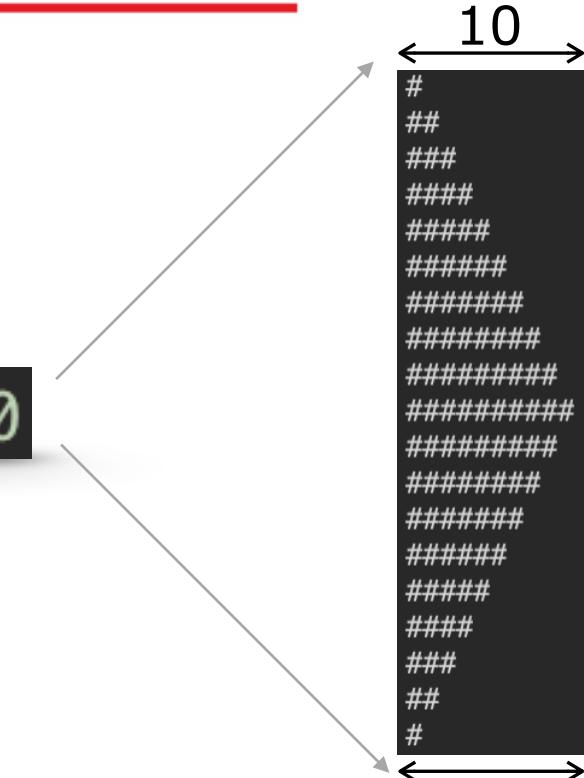
H = 10 → 10



Vežba 5 (ppf-ex06 triangle1.py)

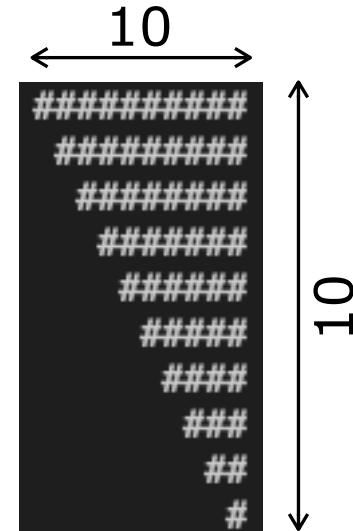
- U aplikaciji postoji promenljiva H
- Potrebno je, pomoću petlje, kreirati trougao kao na slici desno

H = 10



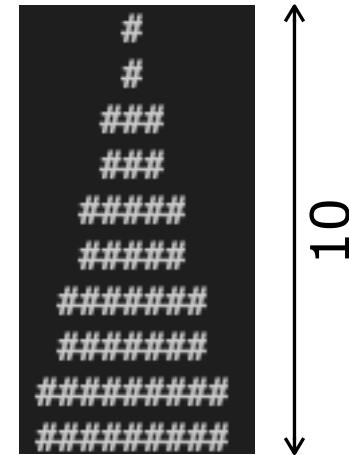
Vežba 6 (ppf-ex06 triangle2.py)

- U aplikaciji postoji promenljiva H
- Potrebno je, pomoću petlje, kreirati trougao kao na slici desno



Vežba 7 (ppf-ex06 triangle3.py)

- U aplikaciji postoji promenljiva H
- Potrebno je, pomoću petlje, kreirati trougao kao na slici desno



Vežba 8 (ppf-ex06 anim.py)

- U programu je definisana promenljiva **target**
- Potrebno je animirati oznaku # tako da se kreće s leva na desno, od početka konzole, do pozicije definisane promenljivom target



Napomena:

Moguće je vratiti se na početak reda u konzoli, oznakom \r

```
print("\r",end="")
```

Moguće je pauzirati izvršavanje programa dodavanjem sledećih linija

```
import time #na početku fajla
```

...

```
time.sleep(0.1) #tamo gde je potrebno pauzirati. 0.1 = 100 milisekundi
```

while petlja

- While petlja se izvršava neodređeno puta
- Ova petlja je najbliža implementacija standardne petlje u Python-u
- Mora se ručno kontrolisati izvršavanje
- Postoji opasnost od zaključavanja

```
while True:  
    print("Hello")
```

The diagram illustrates the structure of a `while` loop in Python. It features a green arrow pointing to the `while` keyword labeled "Ključna reč". A blue arrow points to the expression `True` labeled "Uslov". A red arrow points to the block of code `print("Hello")` labeled "Uslovljeni blok".

while petlja

- Uslov u while petlji je standardan izraz koji rezultuje boolean tipom
- While petlja se može posmatrati kao if uslov koji se ponavlja



Poruka Hello se ispisuje beskonačno

```
x = 1
while x == 1:
    print("Are we there yet")
```

```
while(True):
    print("Are we there yet")
```

Ugnježđena while petlja

- Petlja while se, kao i for petlja, može gnjezditi

```
x = 0
while x < 3:
    y = 0
    while y < 3:
        print("y : ", y)
        y+=1
    print("x : ", y)
    x+=1
```



y :	0
y :	1
y :	2
x :	3
y :	0
y :	1
y :	2
x :	3
y :	0
y :	1
y :	2
x :	3
y :	0
y :	1
y :	2
x :	3

Vežba 9 (ppf-ex06 calc.py)

- Program traži od korisnika unos broja
- Sve dok korisnik unosi numeričke vrednosti, program ih sabira
- Kada korisnik pritisne taster enter umesto numeričke vrednosti program ispisuje zbir i počinje ponovno sabiranje
- Ukoliko korisnik unese vrednost koja nije numerička ispisuje se poruka i preuzima se sledeći unos

```
Enter number: 2
Enter number: 3
Enter number:
Total result: 5
Enter number: f
Value is not numeric
Enter number: █
```

while petlja

- Uslov u while petlji je standardan izraz koji rezultuje boolean tipom
- While petlja se može posmatrati kao if uslov koji se ponavlja



Poruka Hello se ispisuje beskonačno

```
x = 1
while x == 1:
    print("Are we there yet")
```

```
while(True):
    print("Are we there yet")
```

Kontrola petlje

- For i while petlje imaju mogućnost kontrole na nivou iteracije
 - continue** - momentalno prekida trenutnu iteraciju i prelazi na sledeću
 - break** - momentalno prekida petlju

Beskonačno se ispisuje
Hello ali nikad World

```
while True:  
    print("Hello")  
    continue  
    print("World")
```

Samo jednom se
ispisuje Hello

```
while True:  
    print("Hello")  
    break
```

Slučaj kompletiranja petlje

- Često petlju koristimo kako bismo sekvenčijalno pretražili neki izvor podataka. U takvim situacijama petlja gubi smisao u trenutku kada je podatak nađen

```
search = 5
for i in range(10):
    print("Current number: " , i)
    if(i==search):
        print("Number found!")
        break
```

Nakon ovog dela
petlja nema smisao



- Nakon koda iz primera, znamo da je broj nađen, ali šta se dešava ako broj nije pronađen?

Slučaj kompletiranja petlje

- Petlje u Python-u imaju mogućnost obrade slučaja kompletiranja petlje komandom **else**

```
search = 15
for i in range(10):
    print("Current number: " , i)
    if(i==search):
        print("Number found!")
        break
else:
    print("Number not found")
```

```
Current number: 0
Current number: 1
Current number: 2
Current number: 3
Current number: 4
Current number: 5
Current number: 6
Current number: 7
Current number: 8
Current number: 9
Number not found
```

Vežba 10 (ppf-ex06 anim1.py)

- U programu je definisana promenljiva **target**
- Potrebno je animirati oznaku # tako da se kreće s leva na desno i nazad od početka konzole, do pozicije definisane promenljivom target



Napomena:

Moguće je vratiti se na početak reda u konzoli, oznakom \r

```
print("\r",end="")
```

Moguće je pauzirati izvršavanje programa dodavanjem sledećih linija

```
import time #na početku fajla
```

...

```
time.sleep(0.1) #tamo gde je potrebno pauzirati. 0.1 = 100 milisekundi
```

Vežba 11 (ppf-ex06 frame.py)

- U programu postoje promenljive w i h koje predstavljaju širinu i visinu matrice
- Potrebno je dopuniti program tako da se u konzoli prikaže matrica širine w i visine h
- Svaka pozicija u matrici treba da bude predstavljena znakom #
- Matrica treba da se briše i ponovo iscrtava u vremenskom intervalu

Napomena:

Za brisanje konzole dodati:

```
import os
```

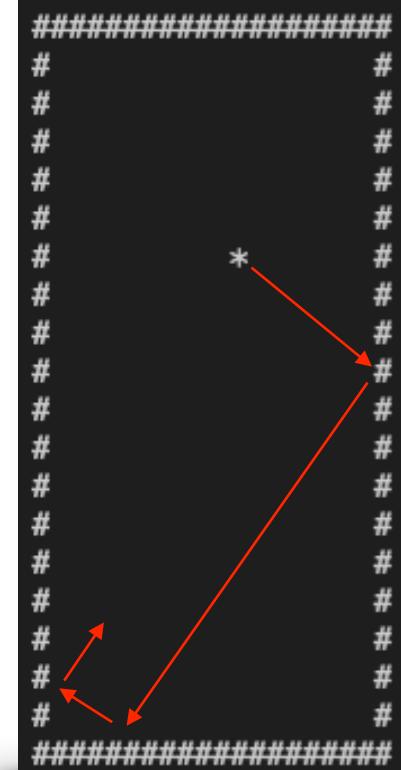
```
...
```

```
os.system("clear")
```

```
#####
#####
```

Vežba 12 (ppf-ex06 ball.py)

- U programu postoje promenljive w i h koje predstavljaju širinu i visinu "ekrana"
- Potrebno je dopuniti program tako da se prikaže okvir veličine ekrana
- Unutar okvira treba da se kreće karakter odbijajući se o ivice ekrana



Vežba 13 (ppf-ex06 rotation.py)

- Kreirati matricu određene veličine
- Uokviriti matricu oznakom # ili nekom drugom
- Unutar matrice, animirati oznaku # tako da se kreće kružno ili po nekom drugom šablonu

Napomena:

Sledeća forma izračunava rotaciju tačke:

```
import math
...
ptx1 = (math.cos(alpha) * ptx) + (-math.sin(alpha) * pty)
pty1 = ((math.sin(alpha)) * ptx) + (math.cos(alpha) * pty)
```

