

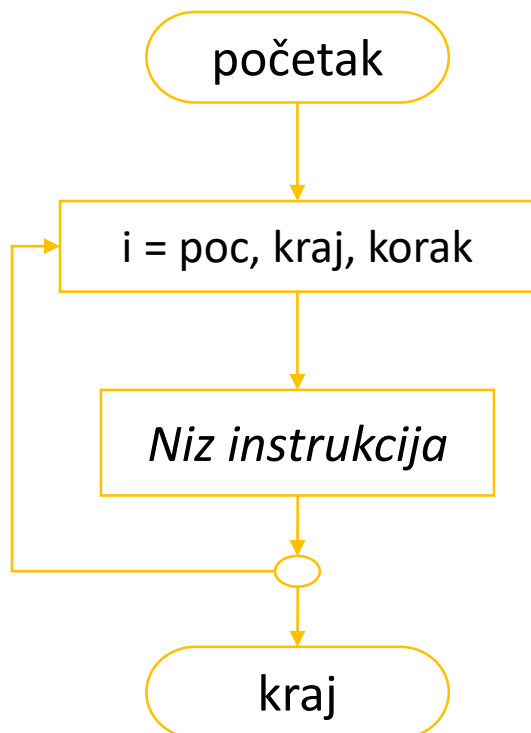
PROGRAMIRANJE 1

Ciklička struktura

- Niz algoritamskih koraka u kojem se jedan ili više algoritamskih koraka može izvršiti više od jedanput, pri jednom izvršavanju algoritma zadatka, tvori cikličku algoritamsku shemu.
- Ciklička struktura nastaje kada je potrebno neku instrukciju ili dio programa ponoviti više puta
- Možemo ih podijeliti na:
 - Petlje koje se izvršavaju određen broj puta
 - Petlje koje se izvršavaju dok se ne ispuni neki uvjet
 - petlje s ispitivanjem uvjeta prije izvođenja niza instrukcija – “dok je *uvjet* činiti”.
 - petlje s ispitivanjem uvjeta nakon izvođenja niza instrukcija – “ponavljati ... do *uvjet*”.

Petlja s poznatim brojem ponavljanja

```
za i = poc do kraj (korak k)  
  niz_instrukcija
```



FOR

for **<var>** **in** **<vrijednosti>**:
 <naredbe>

Indeks petlje

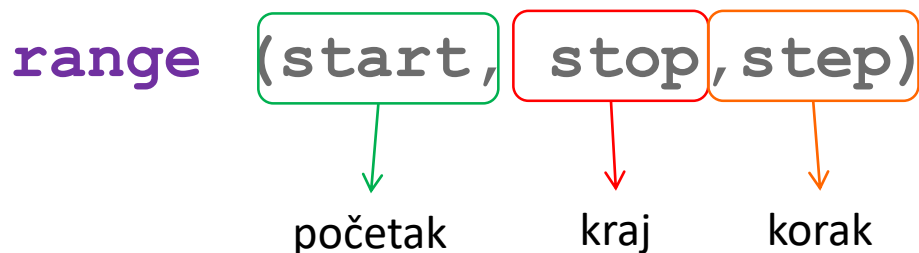
Blok naredbi koje se izvršavaju

```
for i in [0,1,2,3]:  
    print(i, "dobar dan")
```

- Funkcija **range()** je ugrađena funkcija Pythona koja generira niz brojeva
- Sintaksa:

range (start, stop, step)

početak kraj korak



- Ako se upišu sve tri vrijednosti, onda petlja ide od vrijednosti **start** do (**stop-1**) uz povećanje **step**

```
range(1, 10, 2) →  
    1, 3, 5, 7, 9  
range(3, 12, 3) →  
    3, 6, 9
```

- Ako se upišu samo dvije vrijednosti, onda se smatra da su to početna i krajnja vrijednost, te da je **korak 1**

`range(2,5) →`

2, 3, 4

- Ako se upiše samo jedna vrijednost, smatra se da je to krajnja vrijednost, te da je **korak 1**, a **početna vrijednost 0**

`range(5) →` 0, 1, 2, 3, 4

5 vrijednosti

Ako nam nije bitna konkretna vrijednost indeksa petlje, onda ćemo obično staviti samo koliko puta trebamo izvršiti petlju... odnosno, samo jedan broj!

- Ispiši “Pozdrav!” pet puta.

```
for x in range(5):  
    print ("Pozdrav!")
```

- Ispiši sve parne brojeve manje ili jednake 50.

```
for broj in range(2, 51, 2):  
    print (broj)
```

```
for x in range(10, 0, -1):  
    print (x)
```

```
for x in range (1,6,2):  
    print (x)
```

1
3
5

```
for n in range (6,0,-1):  
    print (n)
```

6
5
4
3
2
1

```
for x in range (6,0,-2):  
    print (x)
```

6
4
2

Ugnježdživanje

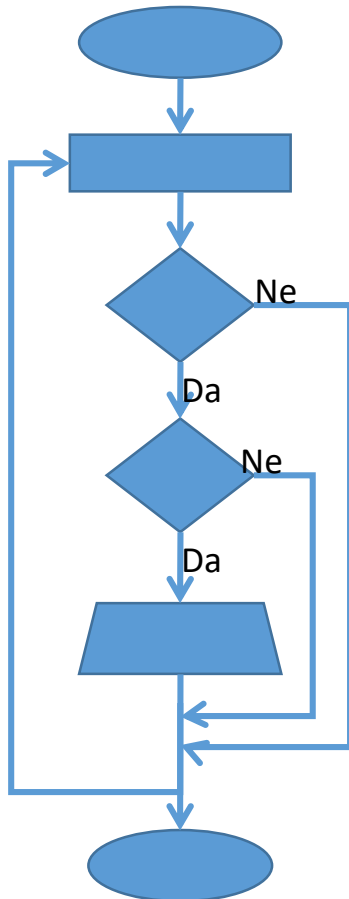
```
for i in range(1,4):  
    for j in range(1,4):  
        print("Ana")  
    print("*****")
```

```
Ana  
Ana  
Ana  
*****  
Ana  
Ana  
Ana  
*****  
Ana  
Ana  
Ana  
*****
```

```
for x in range (1,4):  
    for y in range (1,4):  
        a=a+1  
        b=b+a  
        c=c+b  
  
print (a,b,c)
```

```
9  
45  
165
```

- Napisati program koji će na ekran ispisati sve brojeve koji su djeljivi sa 3, a nisu djeljivi sa 7 između 1000 i 10000.

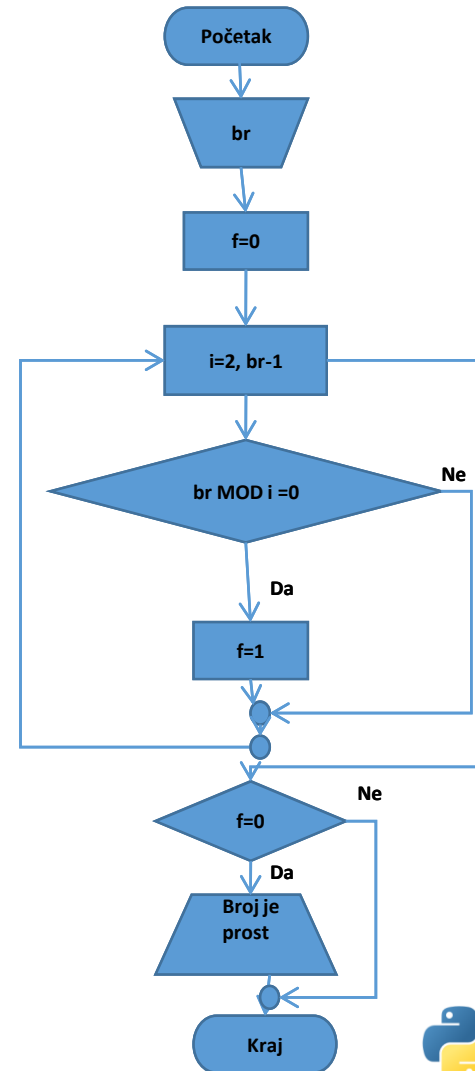


```
za x=1000 do 10000
    ako je (x MOD 3 = 0) onda
        ako je (x MOD 7 > 0) onda
            ispis(x)
```

```
for x in range(1000,10001):
    if (x%3 == 0):
        if (x%7 > 0):
            print(x)
```

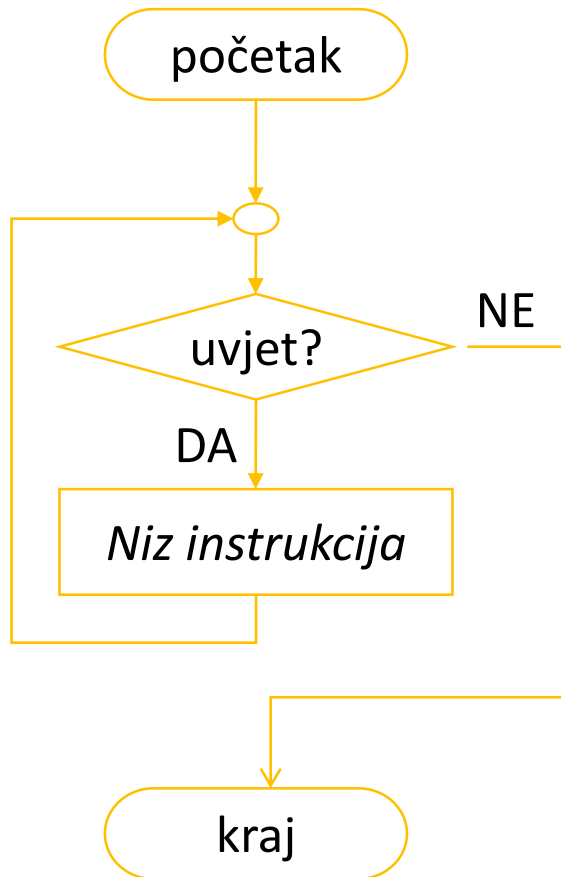
- Upisati broj. Ispitati da li je uneseni broj prost.

```
br = int(input("Unesi broj: "))  
f = 0  
  
for i in range(2,br):  
    if br%i == 0:  
        #ako smo nasli djelitelja  
        f = 1  
        break  
  
if f==0:  
    print(br, "je prost")  
else:  
    print(br, "nije prost")
```



Petlja s ispitivanjem uvjeta ponavljanja na početku

dok je (uvjet)
niz_instrukcija



- koristimo je ako broj ponavljanja petlje nije unaprijed poznat nego čekamo da se ispuni neki uvjet
- kada je uvjet ispunjen petlja se prekida
- petlja prvo provjerava uvjet, ako uvjet vrijedi blok naredbi se izvršava
- blok naredbi se ne mora izvršiti niti jednom, jer se na početku ispituje uvjet, i ukoliko nije ispunjen blok naredbi se ne izvršava

WHILE

```
while (uvjet) :  
    <naredbe>
```

- WHILE petlja se izvršava dok je uvjetni izraz istinit
- Izlazi iz petlje kad je uvjet lažan

```
x = 10  
while (x < 15):  
    print (x)  
    x = x + 1
```

```
10  
11  
12  
13  
14
```

```
n=4
z = 1
i = 1
dok je ne(i = n)
    i = i + 1
    z = z * i
ispis (z - i)
```

20

```
i = 0
z = 0
dok je ne(i > 16)
    i = i + 1
    ako je i < 5 onda z = z + 1
ispis (i - z)
```

13

```
i=1
t=0
dok je (i<=5)
    t=t+2
    i=i+1
ispis (i,t)
```

6, 10

```
x=2
y=1
dok je (x<=10)
    x=x+2
    y=y+x
ispis (x,y)
```

12, 41

- Beskonačna **while** petlja

```
while 1:  
    print("Pozdrav!")
```

- Zaustavlja se kombinacijom tipki: CTRL+C

- Kontrolirani unos podataka

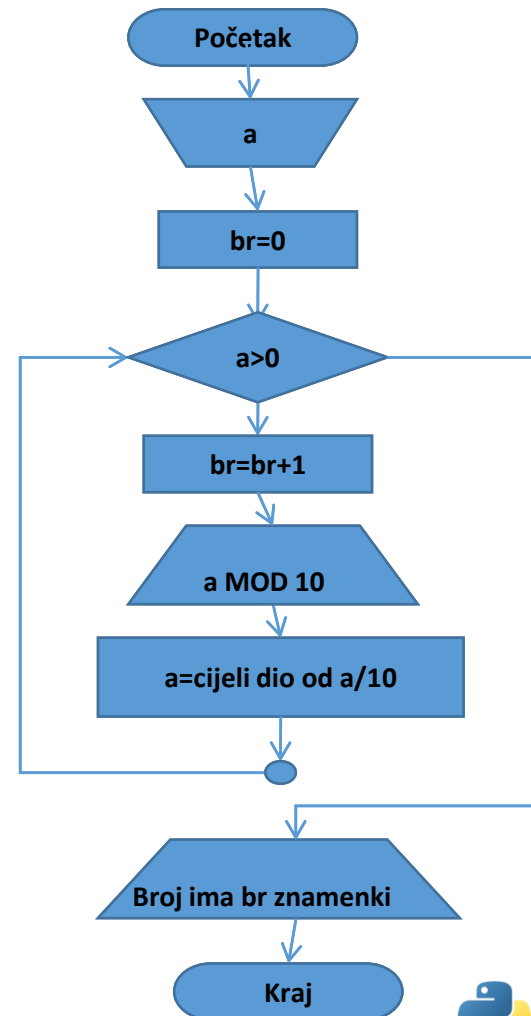
```
#uvjet ovisi o korisniku
x = -1
while x<=0:
    x = int(input("Unesi poz.broj: "))
```

- Možemo i programirati izlaz iz petlje uz neki uvjet kombinacijom **if** i **break**

```
while 1:
    x = int(input("Unesi paran broj: "))
    if x % 2 == 0:
        break
```

- Učitati prirodan broj. Ispisati taj broj. Ispisati znamenke tog broja i koliko ih ima.

```
a=int(input())  
br=0  
while (a>0):  
    br=br+1  
    print (a%10)  
    a=int(a/10)  
print ("Broj ima",br,"znamenki")
```



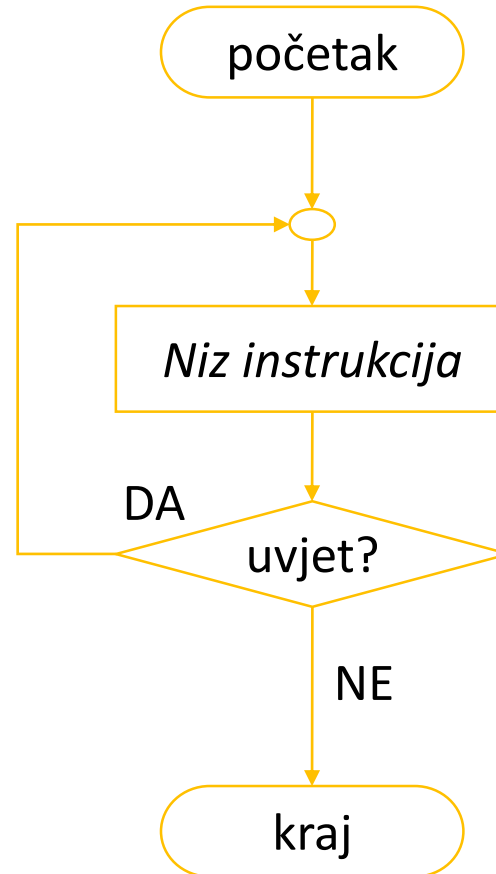
- Napisati program koji će učitati prirodan broj. Ispisati taj broj. Pronaći i ispisati koliko puta se ponovila znamenka 4 u tom broju.

```
unesi (broj)
br=0

dok je broj>0
    ako je (broj MOD 10 = 4) onda
        br=br+1
    broj=broj DIV 10
ispis ("Znamenka 4 se pojavila";br;"puta")
```

Petlja s ispitivanjem uvjeta ponavljanja na kraju

```
ponavljaj  
    niz_instrukcija  
dok je (uvjet)
```

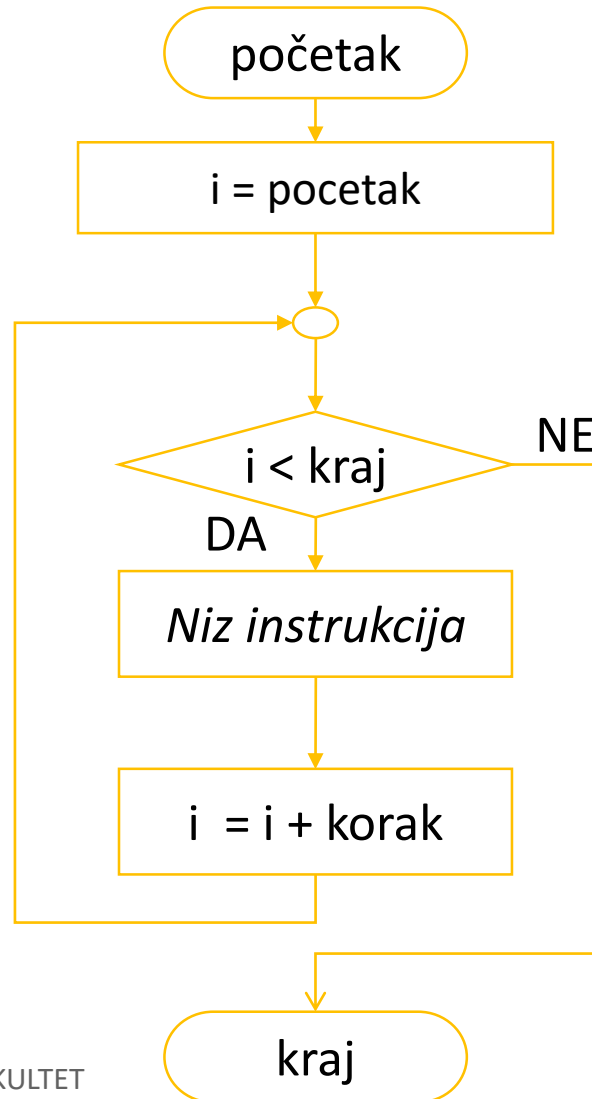


- koristi se kada broj ponavljanja petlje nije unaprijed poznat
- petlja prvo izvršava blok naredbi, pa tek onda ispituje uvjet
- blok naredbi se izvršava barem jednom, čak i ako uvjet od početka nije ispunjen

```
i=1
a=1
ponavljaaj
    a=a+2
dok je (i<=10)
    ispis (a,i)
```

```
beskonačna
petlja
```

Petlja s određenim brojem ponavljanja kao petlja s uvjetom



Izračunavanje N faktoriijela

- Za svaki prirodni broj n definira se faktoriijela na slijedeći način:

$$n! = 1 \cdot 2 \cdot 3 \cdots n$$

Pritom vrijedi $0! = 1$ i $1! = 1$

Npr.:

$$2! = 1 \cdot 2 = 2$$

$$5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$$

- Rješenje s petljom s poznatim brojem ponavljanja

```
učitaj n  
fakt = 1
```

```
za i = 1 do n  
    fakt = fakt * i
```

```
ispiši n, fakt
```

- Rješenje s petljom u kojoj se uvjet ispituje na početku

```
učitaj n
fakt=1
i=1
dok je (i <= n)
    fakt = fakt * i
    i = i+1
ispiši n,fakt
```

- Rješenje s petljom u kojoj se uvjet ispituje na kraju

```
učitaj n
fakt = 1
i = 1
ponavljaaj
    fakt = fakt * i
    i = i+1
dok je (i <= n)
ispisi n,fakt
```

Zadaci

- Obavezno pročitati do sljedećih vježbi i predavanja:
“Rješavanje problema programiranjem u Pythonu”
stranice: 84-117