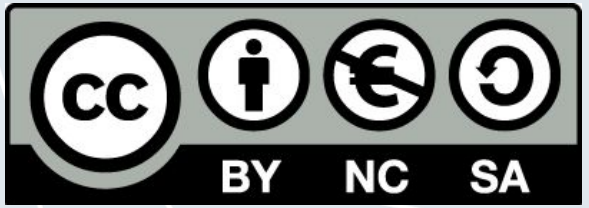
SVEUČILIŠTE U MOSTARU

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

PROGRAMIRANJE 1

While petlja
Operatori za rad sa listama

doc. dr. sc. Tomislav Volarić
mag. ing. comp. Robert Rozić



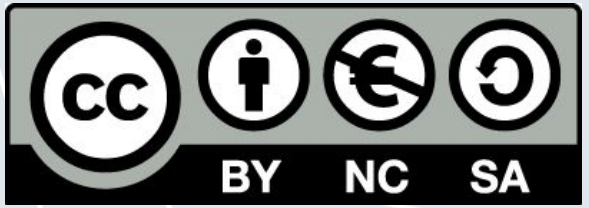
SVEUČILIŠTE U MOSTARU

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

Creative Commons

- **Slobodno smijete:**
  - **dijeliti** - umnožavati, distribuirati i javnosti priopćavati djelo
  - **remiksirati** - prerađivati djelo
- **pod sljedećim uvjetima:**
  - **Imenovanje**. Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
  - **Nekomercijalno**. Ovo djelo ne smijete koristiti u komercijalne svrhe.
  - **Dijeli pod istim uvjetima**. Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.
Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.
Tekst licence preuzet je s https://creativecommons.org/.



Okruženja za rad u Python-u

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

IDE - Integrated Development Environment

- Službeni Python IDLE (Integrated Development and Learning Environment)
- Thonny IDE
- Online interpreteri npr. Programiz
- Linux Python interpreter

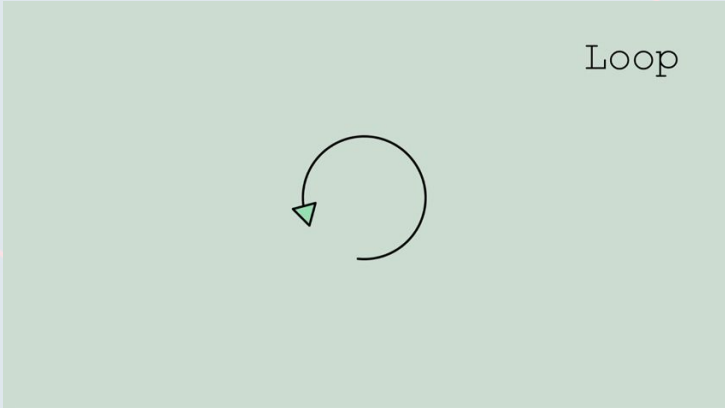
Ciklička struktura - loop, petlja

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

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.

Python - for i **while** petlje



Python while petlja

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

While petlja u Pythonu koristi se za ponavljanje bloka koda sve dok je određeni uvjet ispunjen.

Ovu petlju koristimo kada ne znamo koliko puta želimo izvršiti određeni blok koda.

```
while testni_uvjet:
    tijelo petlje
```

petlja se izvršava sve dok je uvjet ispunjen

Koji je uvjet za kraj izvršavanja for petlje ?

Python while petlja

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

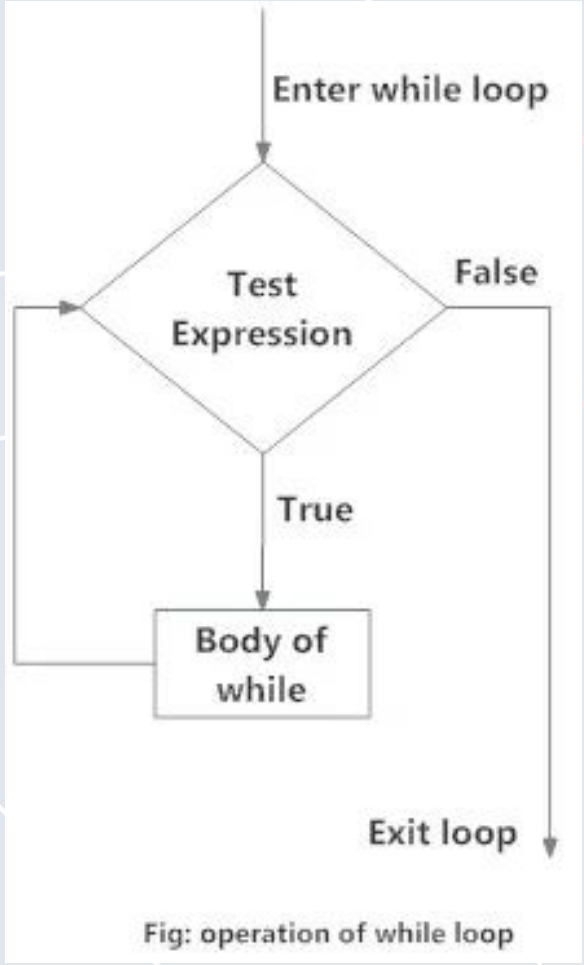


Fig: operation of while loop

Python while petlja

Primjer

Program za izračun zbroja brojeva do određenog broja

```
n = 10
suma = 0
i = 1
while i <= n:
    suma = suma + i
    i = i+1

print("Suma je", sum)
```

Kolekcije podataka u Python-u - Liste

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

Lista je struktura koja može pohraniti kolekciju elemenata istog tipa.
npr. String - niz znakova (engl. characters)

Svi podaci u nekoj kolekciji većinom su istog tipa.

- lista (Python, LOGO)
- niz (eng. array) (Qbasic, C)

```
n = 10
brojevi = [0] * n
for i in range(n):
    brojevi[i] = int(input('Unesi broj: '))
```

Liste

Python liste su dinamičke - mogu se mijenjati tijekom izvođenja programa

```
brojevi = []
unos = int(input('Unesi broj: '))
while(unos > 50):
    brojevi.append(unos)
    unos = int(input('Unesi broj: '))

for broj in brojevi:
    print(broj)
```

Operatori za rad sa listama

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

	Operator	Primjena	Rezultat
+	spajanje	>>>“danas” + “sutra”	danassutra
*	umnožavanje	>>> “danas” * 3	danasdanasdanas
[]	indeksiranje znakovnog niza	>>> “danas”[0] >>> “danas”[1]	d a
[:]	podniz	>>> “danas”[0:3] >>> “danas”[2:4]	dan nas
in	prvi string sadržan u drugom	>>> “dan” in “sutra”	False
not in	prvi string nije sadržan u drugom	>>> “dan” not in “sutra”	True

Indeksi u listi omogućavaju pristup pojedinom elementu liste odnosno znaku u stringu.
Pritom prvi element liste ima indeks 0, drugi ima indeks 1 itd.

Operatori za rad sa listama

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

Python podržava i negativne indekse u listi.
Tako zadnji element u listi ima indeks -1, predzadnji znak ima indeks -2….

>>> s=input(‘Unesi riječ: ‘)	Unesi riječ: danas
>>> s[-1]	‘s’
>>> s[-2]	‘a’
>>> s[::2]	‘dns’

Ako želimo dohvatiti svaki drugi znak u stringu onda to radimo na ovaj način: **s[::2] | ‘dns’**
Ovdje koristimo drugu dvotočku za zadavanje koraka po kojem dohvaćamo znakove u nekom rasponu.
Na primjer, korak 2 je za dohvaćanje svakog drugog znaka u rasponu.

Primjer zadatka

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

Što će ispisati program?

```
a='kolokvij'
b='test'
if a[2:7] != 'olo':
        if b[3] == 't':
            b=str(2**2-3)
elif b[::2] == 'ts':
        b=str(6%3)

print(b + a)
```

Literatura

- **Think IT 1, (2019.) Udžbenik iz informatike za prvi razred gimnazije, ALFA d. d.**
- **https://www.programiz.com/python-programming/while-loop**

sumarum.sum.ba

SUMARUM
sustav e-učenja

FPMOZ
Fakultet prirodoslovno-matematičkih i odgojnih znanosti
Sveučilište u Mostaru

Hvala na pozornosti!

tomislav.volaric@fpmoz.sum.ba

robert.rozic@fpmoz.sum.ba