

# While petlje

Igor Buzov, dipl. inf, viši predavač

Centar umjetne inteligencije Lipik

## Kada koristimo for petlje?

- For petlje koristimo kada nam je poznat konačan broj iteracija

- Primjer 1:

```
for i in range(4):
```

```
    print(i)
```

- Primjer 2:

```
gradovi = ["zagreb", "nOVA gRADIŠKA", "OSIJEK"]
```

```
for mjesto in gradovi:
```

```
    print (mjesto.title())
```

# Kada koristimo while petlju?

- While petlja se koristi dokle god je zadovoljen određeni uvjet
- While petlja se koristi također kada želimo beskonačnu petlju iz koje se može izaći po želji korisnika

## Primjer 1

```
current_number = 1
```

```
while current_number <= 5:
```

```
    print(current_number)
```

```
    current_number += 1
```

- Kao i druge petlje, završava dvotočkom, a blok koda ispod petlje se uvlači

```
def zbrajanje (a,b):
```

```
    return a + b
```

```
print ("Dobro došli jednostavnu kalkulator aplikaciju")
```

```
while True:
```

```
    print ("Za zbrajanje dva broja, unesite riječ 'zbrajanje'")
```

```
    print ("Upišite 'quit' ako želite završiti")
```

```
    odgovor = input ("Vaš odgovor: ")
```

```
    if odgovor == "quit":
```

```
        print ("Zahvaljujem na korištenju")
```

```
        break
```

```
    if odgovor == "zbrajanje":
```

```
        broj1 = float(input("Unesite prvi broj"))
```

```
        broj2 = float(input("Unesite drugi broj"))
```

```
        rezultat = zbrajanje (broj1, broj2)
```

```
        print ("Zbroj vaša dva broja glasi:", rezultat)
```

```
print ("kaj programa")
```

## Primjer 2

## Zadatak 1

- Želite napraviti program koji korisnika pita za ime grada koji bi htio posjetiti i pohranjuje odgovor u varijablu 'grad'
- Kao odgovor, program ispisuje tekst:
  - 'grad' je prekrasan u ovo doba godine!
- Na početku programa je izbornik gdje korisnik može odabrati 'početak programa' ili 'kraj programa'
- Program se izvršava dokle god korisnik ne upiše 'kraj programa'

# Naredba import

- Naredba import nam omogućuje da koristimo posebne alate koje Python već ima, a da ih mi sami ne moramo pisati.
- Želimo koristiti funkcije, varijable i druge stvari koje su napisane i spremljene negdje drugdje
- Kad bi sve funkcije i alati ikada stvoreni (od matematičkih operacija, preko grafike, mrežnih veza, rada s bazama podataka, umjetne inteligencije itd.) bili dio "osnovnog Pythona", program bi bio gigantski
- Dodatni sadržaji Pythona su pohranjeni u modulima. Svaki modul ima svoj naziv

# Modul random

- Modul random je jedan od ugrađenih modula u Pythonu. To je "kutija s alatom" koja sadrži funkcije za rad sa slučajnošću.
- Koristimo ga da bismo generirali nešto što se čini nasumičnim ili nepredvidivim (npr. bacanje kocke, izvlačenje karte, odabir broja bez predvidivog uzorka)
- Ključne funkcije unutar random modula:
  - `random.randint(a, b)`: Vraća slučajan cijeli broj između a i b, uključujući a i b. (npr. `random.randint(1, 10)` daje cijeli broj od 1 do 10).
  - `random.random()`: Vraća slučajan decimalni broj (float) između 0.0 (uključivo) i 1.0 (isključivo).
  - `random.choice(lista)`: Slučajno odabire jedan element iz zadane liste. (npr. `random.choice(['jabuka', 'banana', 'trešnja'])`).



## Primjer 3

```
import random
```

```
slucajni_broj = random.randint(1, 100)
```

```
print (slucajni_broj)
```

## Zadatak 2 – grupni rad

- Koršitenjem `random.randint ()` funkcije, generirajte neki slučajni broj
- Zatražite od korisnika da pokuša pogoditi vaš broj
- Korisnik pogađa na način da unosi broj
- Kao odgovor, dobija je li pogodio, je li njegov broj prevelik ili premali
- Kad naponon pogodi, program se završava
- Korisnik ima mogućnost pogađanja dokle god ne pogodi broj

## Zadatak 3 – grupni rad

- Napravite aplikaciju za izradu To do liste prema uputama koje ćete dobiti