



python™

# Podatkovni tipi

- Števila
  - Cela (Integer)
    - 3; 236411; -345698;
  - Realna (Real)
    - 3,14; -3456.12; 3e3;
  - Kompleksna
    - -5+2j; 1.7-5.4j; 0.5j;
- Bool
  - Logične vrednosti
    - True (1), False (0)
- Niz
  - Niz znakov v narekovaju
    - "Pozdravljeni" '1 + 1'
- Tabela
  - [5,56,88,2345]

# Številski izrazi

- + seštevanje  $2+3=5$
- odštevanje  $5-2=3$
- \* množenje  $2*3=6$
- / deljenje  $5/2=2.5$
- // celoštevilčno deljenje  $5//2=2$
- % ostanek pri celoštevilčnem deljenju  $6\%4=2$
- \*\* potenciranje  $2**3=8$
- += doda stari vrednosti
- () spreminjanje vrstnega reda računanja  $(2+5)*4=28$

# Izrazi z nizi

<b>+</b>	združevanje	<code>"Janez"+"Novak"="JanezNovak"</code>
<b>*</b>	ponavljanje	<code>"Janez"*3 = "JanezJanezJanez"</code>
<b>[a]</b>	indeksiranje	<code>"Janez"[1] = "a"</code>
<b>[-a]</b>	indeksiranje od konca	<code>"Janez"[-2] = "e"</code>
<b>[a:b]</b>	odrezovanje	<code>"Janez"[2:4]="ne"</code>
<b>len(a)</b>	štetje znakov	<code>len("Janez") = 5</code>
<b>&gt;</b>	primerjanje	<code>"A" &lt; "B" = True</code>
<b>in</b>	iskanje	<code>"ane" in "Janez" = True</code>

# Nizi

- "hello"+"world" "helloworld" # konkatencija
- "hello"\*3 "hellohellohello" # ponavljanje
- "hello"[0] "h" # indeksiranje
- "hello"[-1] "o" # (od konca)
- "hello"[1:4] "ell" # del
- len("hello") 5 # velikost
- "hello" < "jello" 1 # primerjanje
- "e" in "hello" 1 # iskanje
- "escapes: \n etc, \033 etc, \if etc"
- 'single quotes' ""triple quotes"" r"raw strings"

# Logični izrazi

- >** večji kot  $5 > 3 \Rightarrow \text{True}$
- <** manjši kot  $A < B \Rightarrow \text{True}$
- ==** je enak  $\text{"Janez"} == \text{"Miha"} \Rightarrow \text{False}$
- !=** različen  $300 != 3e2 \Rightarrow \text{False}$
- >=** večji ali enak  $\text{len}(\text{"Janez"}) \Rightarrow 5$
- <=** manjši ali enak  $\text{"A"} < \text{"B"} \Rightarrow \text{True}$
- and** logični in  $\text{True and False} \Rightarrow \text{False}$
- or** logični ali  $\text{True or False} \Rightarrow \text{True}$
- not** logični ne  $\text{not True} \Rightarrow \text{False}$

# Spremenljivke

- Ime spremenljivke je poljubno
  - Prvi znak mora biti črka
  - Ne sme vsebovati presledka, matematičnih znakov ...
  - A in a sta različni spremenljivki

```
>>> ime = "Janez"  
>>> A = 5  
>>>a=6  
>>> A  
5
```

- 
- Med izvajanjem programa spremenljivka spreminja svojo vrednost

```
>>> a=5  
>>> a  
5  
>>> a=6  
>>> a
```

# Spremenljivke

- Spremenljivko opredelimo tako, da ji priredimo vrednost

```
>>> a = 5
>>> b = 3,14
>>> c = True
>>> d = a + b
>>> e = [1, 2, 3, 4]
```

- 
- Med izvajanjem programa spremenljivka lahko spremeni svoj tip

```
>>> a = 5
>>> a = 3,14
>>> a = [1, 2, 3, 4]
```



# Program

Izdelaj program, ki prebere dve števili in izpiše njun produkt

- Program najprej napišemo

**File>New**

```
a=int(input("Prvo število:")) # v spremenljivko a se vpiše celo število
```

```
b=int(input("Drugo število:"))
```

```
print (a*b)
```

- Shranimo

**File>Save**

- Poženemo

**File>Run**

## FUNKCIJA ZA IZPIS PODATKOV NA ZASLON

`print(a)` – izpiše se vrednost spremenljivke `a`

`print('a')` – izpiše se znak `a`

## FUNKCIJA ZA BRANJE PODATKOV IZ TASTATURE

`a = input()` – prebere tekst iz tastature in ga spravi v spremenljivko `a`

`a = input('Vstavi vrednost za spremenljivko a: ')` – prebere tekst iz tastature in ga spravi v spremenljivko `a`

`a = int(input('Vstavi vrednost za spremenljivko a: '))` – prebere celo število iz tastature in ga spravi v spremenljivko `a`

`a = float(input('Vstavi vrednost za spremenljivko a: '))` – prebere realno število iz tastature in ga spravi v spremenljivko `a`

`# To je komentar, tolmac tega ne tolmaci`

# Linearni program

Izdelaj program, ki prebere dve števili in izpiše njun produkt

```
a=int(input("Prvo število:"))  
b=int(input("Drugo število:"))  
print (a*b)
```

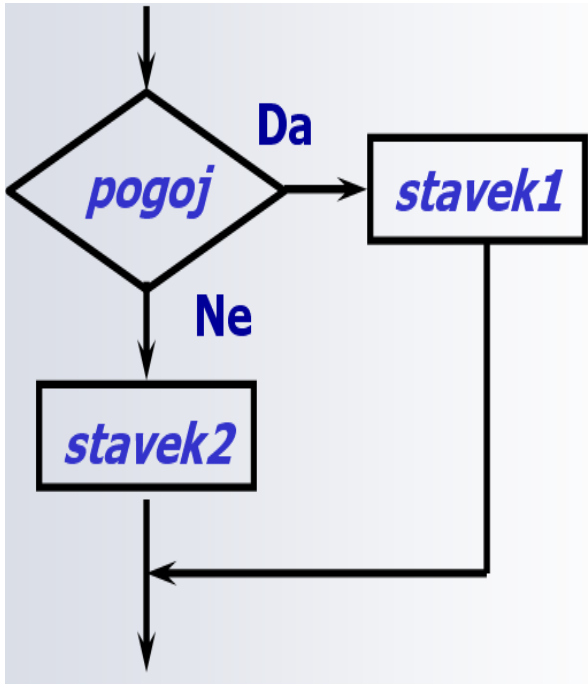
# Pogojni stavek

Za vejitev v Pythonu uporabimo stavek if:

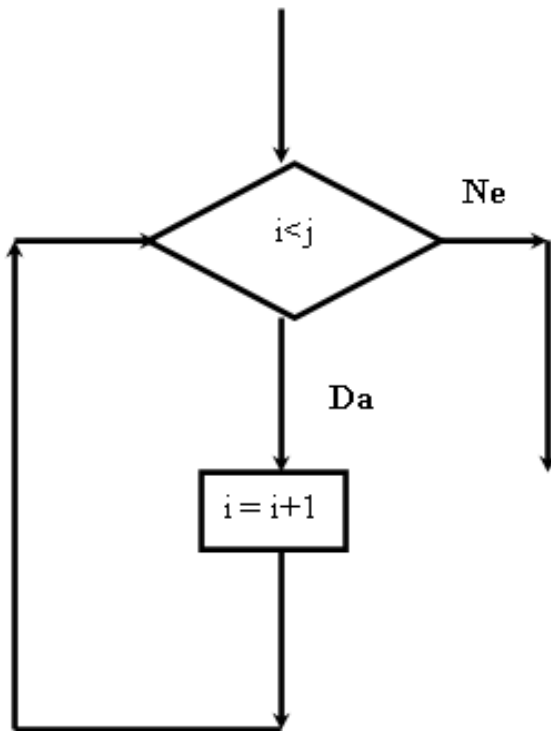
```
if <pogoj>:  
    prvi blok stavkov..  
else:  
    drugi blok stavkov..
```

ali pa

```
if <pogoj 1>:  
    prvi blok stavkov..  
elif <pogoj 2>:  
    drugi blok stavkov..  
else:  
    tretji blok stavkov..
```



# Zanka while



**while <pogoj>:**

**prvi blok stavkov..**

**ali**

**while <pogoj>:**

**prvi blok stavkov..**

**else:**

**drugi blok stavkov..**

# Zanka for

for i in S:

**prvi blok stavkov ...**

#stavki1 se izvajajo, dokler je vrednost spremenljivke i v okviru seznama S

**else:**

**drugi blok stavkov ...**

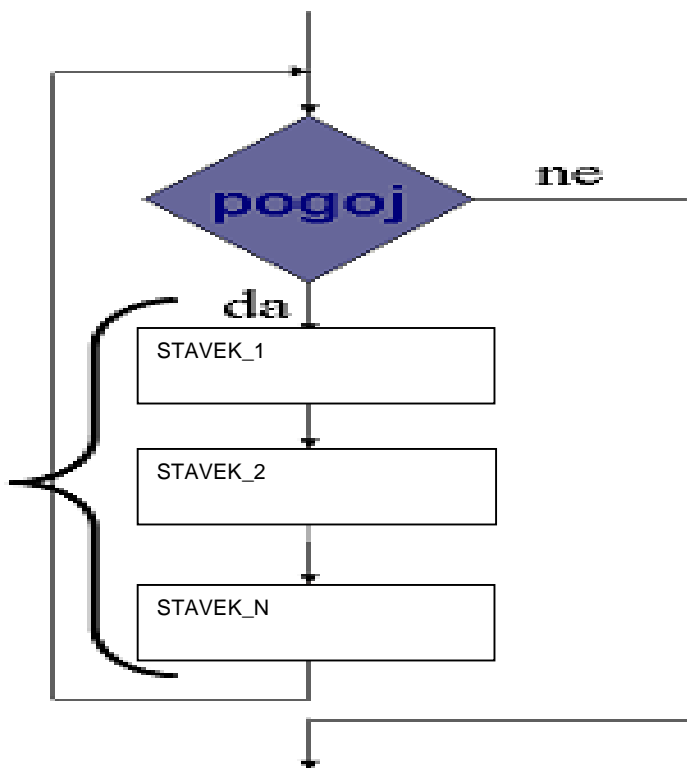
#stavki2 se izvajajo na koncu, ko zgornji pogoj ni več izpolnjen, razen če med stavki1 ni bil uporabljen stavek break

**for i in range(a,b,k)**

Zanka se izvaja, dokler velja :

**$a \leq i < b$**

**i se večja za k**



*dokler je pogoj resničen, se izvajajo ti stavki*

# Tabele

- je zaporedje podatkov, znotraj oglatih oklepajev ločenih z vejico:

```
starost = [15, 14, 16, 17, 14]
```

```
dijaki = ['Mija', 'Tine', 'Jože', 'Nika', 'Pika']
```

- V tabeli so lahko podatki različnih podatkovnih tipov:

```
starost_dijakov = ['Mija', 15, 'Tine', 14, 'Jože', 16, 'Nika', 17,  
                  'Pika', 14]
```

# Tabele

- Elementi tabele so dostopni preko **indeksov**:

```
>>> dijaki = ["Mija", "Tine", "Jože", "Nika", "Pika"]
>>> starost=[15, 14, 16, 17, 14]
>>>
>>> dijaki[0]
'Mija'
>>> starost[1]
14
>>> dijaki[-1]
'Pika'
>>> starost[4]
14
```



# Tabele

- Številu elementov v tabeli pravimo **dolžina** tabele. Pridobimo jo s funkcijo `len`:

```
>>> dijaki = ["Mija", "Tine", "Jože", "Nika", "Pika"]
>>> starost=[15, 14, 16, 17, 14]
>>>
>>> len(dijaki)
5
>>> len(starost)
5
```

# Tabele

- Z izrazom *tabela[a:b]* dobimo **odsek tabele**, ki je sestavljen iz elementov z indeksi od *a* do vključno *b - 1*.

```
>>> dijaki = ["Mija", "Tine", "Jože", "Nika", "Pika"]
>>> dijaki[2:4]
['Jože', 'Nika']
>>> dijaki[1:2]
['Tine']
>>> dijaki[3:]
['Nika', 'Pika']
>>> dijaki[:2]
['Mija', 'Tine']
....
```

# Sprehod po elementih tabele

```
dijaki = ["Mija", "Tine", "Jože", "Nika", "Pika"]
```

```
for dijak in dijaki:  
    print(dijak)
```

```
>>>  
RESTART: C:/Users/Maja/AppData/Local/Programs/Python/Python35-32/sprehod_po_tab  
eli.py  
Mija  
Tine  
Jože  
Nika  
Pika  
>>> |
```

# Sprehod po indeksih tabele

```
dijaki = ["Mija", "Tine", "Jože", "Nika", "Pika"]  
  
for indeks in range(len(dijaki)):  
    print(indeks+1, ". dijak je: ", dijaki[indeks], sep="")
```

```
RESTART: C:/Users/Maja/AppData/Local/Programs/Python/Python35-32/sprehod_po_indeksih_tab  
ele.py  
1. dijak je: Mija  
2. dijak je: Tine  
3. dijak je: Jože  
4. dijak je: Nika  
5. dijak je: Pika  
>>> |
```

# Funkcije

- **Funkcija** ali **podprogram** je zaporedje stavkov.
- Z uporabo funkcij program razdelimo na manjše in bolj obvladljive dele.
- Funkciji damo ime, s katerim jo nato lahko pokličemo nekje iz programa.
- Pri klicu lahko funkcija dobi tudi vhodne podatke (parametre) in na koncu lahko vrne nek rezultat.

# Funkcije

```
def vecje(a, b): #stavek def pove, da gre za deklaracijo funkcije
```

```
    if a > b:  
        return a  
    else:  
        return b
```

```
a=int(input("Vpiši prvo število: "))  
b=int(input("Vpiši drugo število: "))  
print(vecje(a,b))
```



klic funkcije vecje

# Literatura

- <https://lusy.fri.uni-lj.si/ucbenik/>