



OSNOVE PYTHONA I ANALIZA I VIZUALIZACIJA PODATAKA

Virtualna radionica

Mentori: Tijana Paunović, **Jelena Mitrović** i Siniša Bubonja

UPOZNAJ PYLADIES!

- [PyLadies](#)
- [Pyladies Dobj](#)

O PROGRAMIRANJU

Ejda Bajron (prvi programer)

Programerima nije potrebno veliko poznavanje matematike

Biti dobar programer slično je kao biti dobar u rešavanju sudoku
slagalice

Programiranje je kreativna aktivnost – LEGO



OBIM KVADRATA

Algoritam

1. učitaj stranicu kvadrata
2. izračunaj obim po formuli $O = 4 \cdot a$
3. prikaži rezultat

Pitanja implementacije algoritma

- Kako se učitava podatak? (naredba input)
- Kako se pamti ? (dodela vrednosti =)
- Kako se računa formula? (*aritmetički izraz*)
- Kako se prikazuje rezultat? (naredba print)

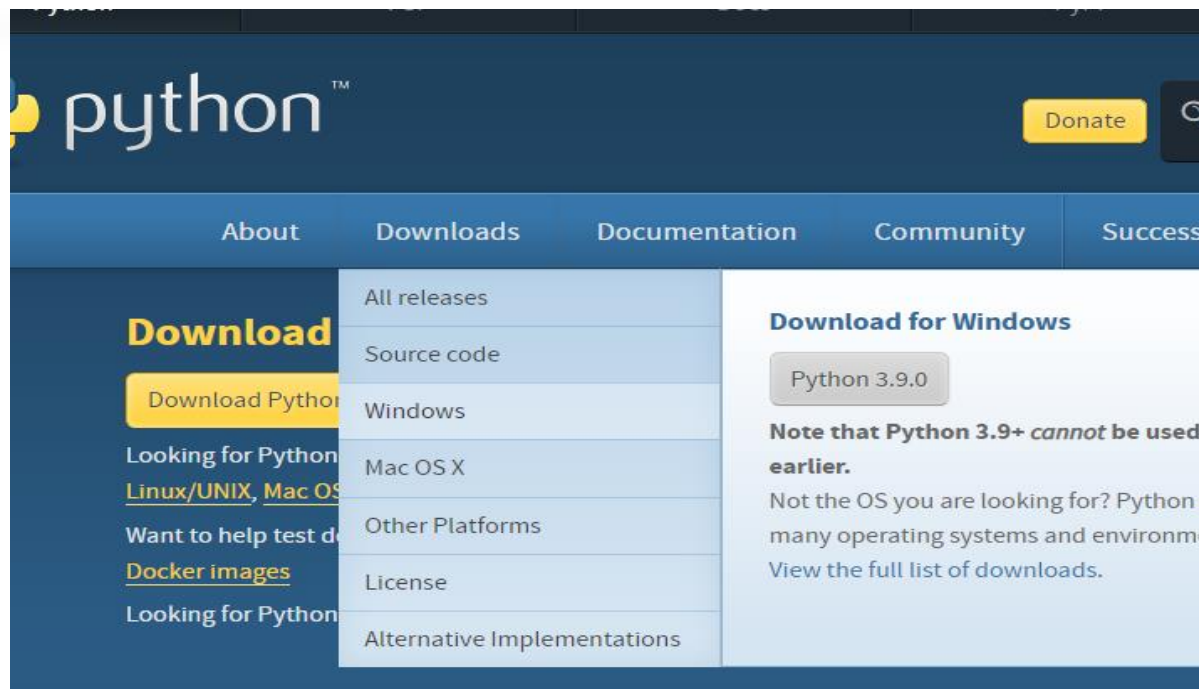


OSNOVE PYTHONA

INSTALACIJA PYTHON-A

Koristićemo aktuelnu verziju 3.9.0

Zavisno od vrste računara i operativnog sistema, preuzima se i pokreće odgovarajući instalacioni program



RADNO OKRUŽENJE – ZA POČETAK IDLE

IDLE je integrisano razvojno okruženje (Integrated Development Environment – IDE) koje dolazi sa Python instalacijom.

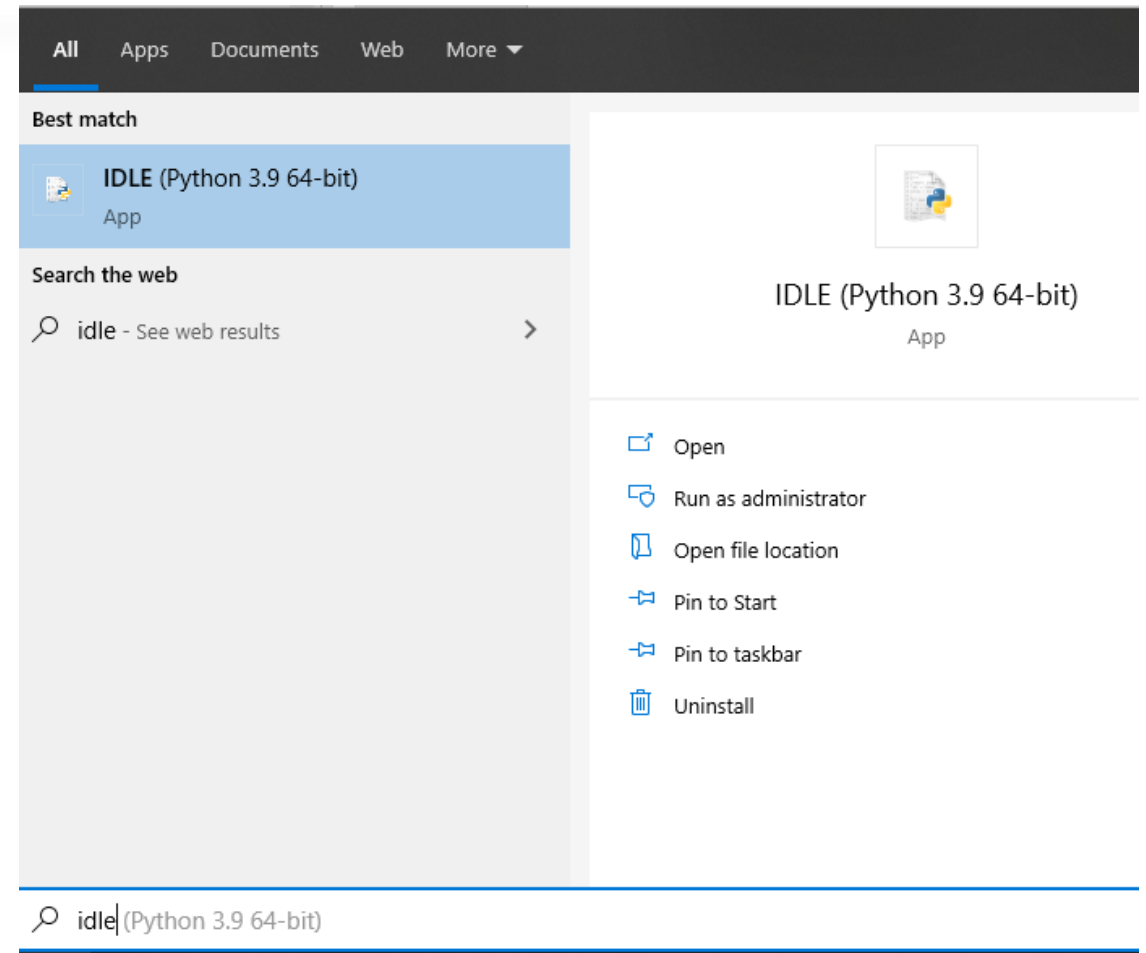
To je program koji vam omogućava da editujete i izvršavate Python programe.

Postoje i druga razvojna okruženja za Python, ali za početak preporučujemo IDLE-a zbog njegove jednostavnosti.



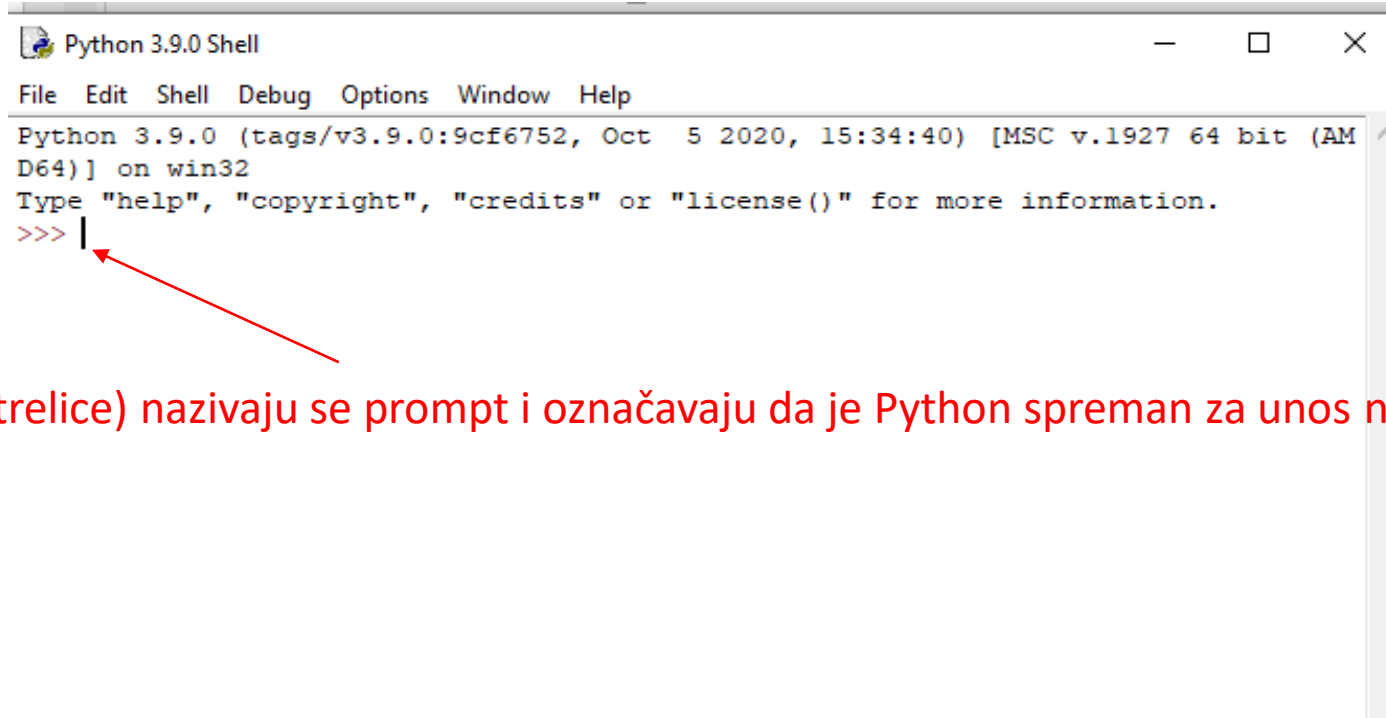
RADNO OKRUŽENJE- IDLE

Pokretanje interpretera/editora



RADNO OKRUŽENJE

Kada startujete IDLE, otvoriće se poseban **interaktivni** prozor (shell) u kome možete da unostite Python naredbe, kao i da odmah vidite rezultat izvršavanja tih naredbi. Ovaj prozor se najčešće koristi kao kalkulator ili za testiranje kratkih kodova - programa.



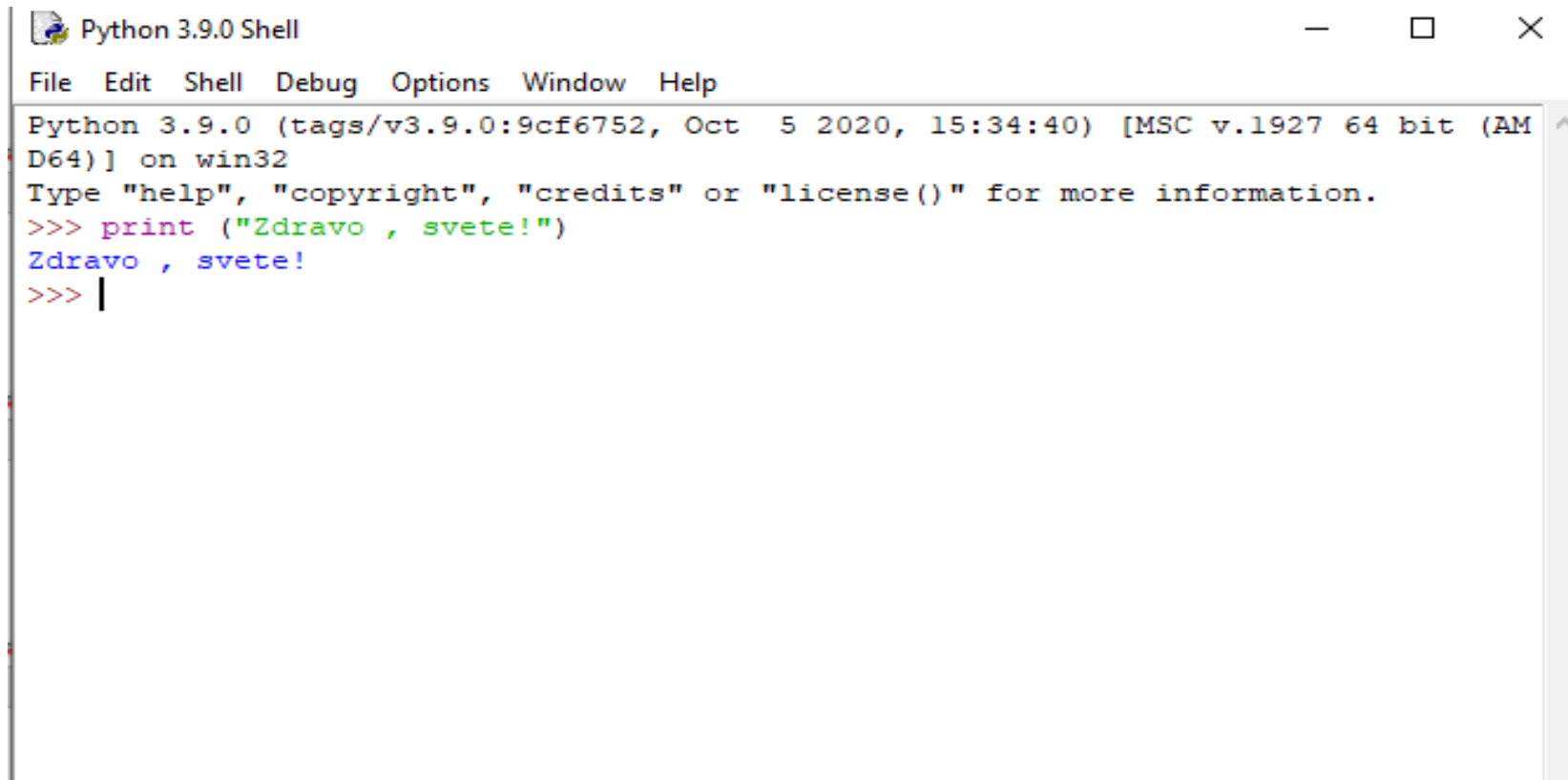
```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
```

>>> (tri strelice) nazivaju se prompt i označavaju da je Python spreman za unos naredbe



PRIMER: HELLO, WORLD!

Postoji tradicija da se na početku učenja programskih jezika uvijek prvo napiše program koji ispisuje "Zdravo, svete!" na ekran.



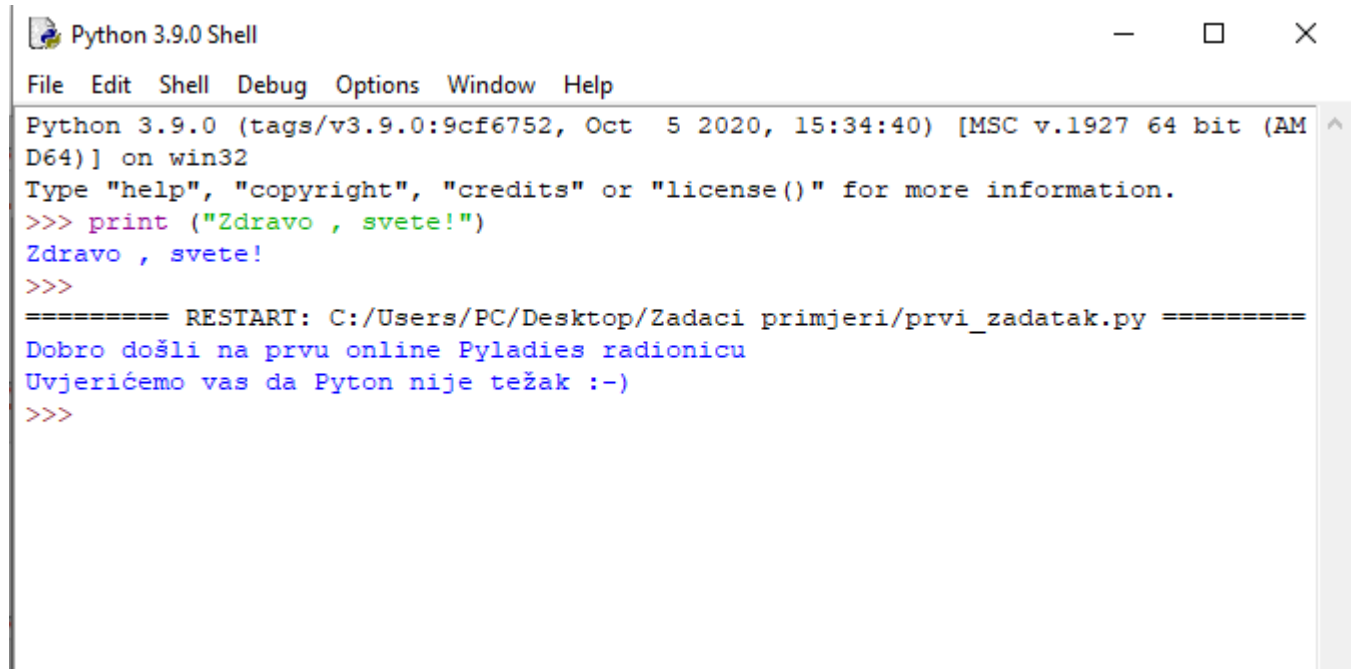
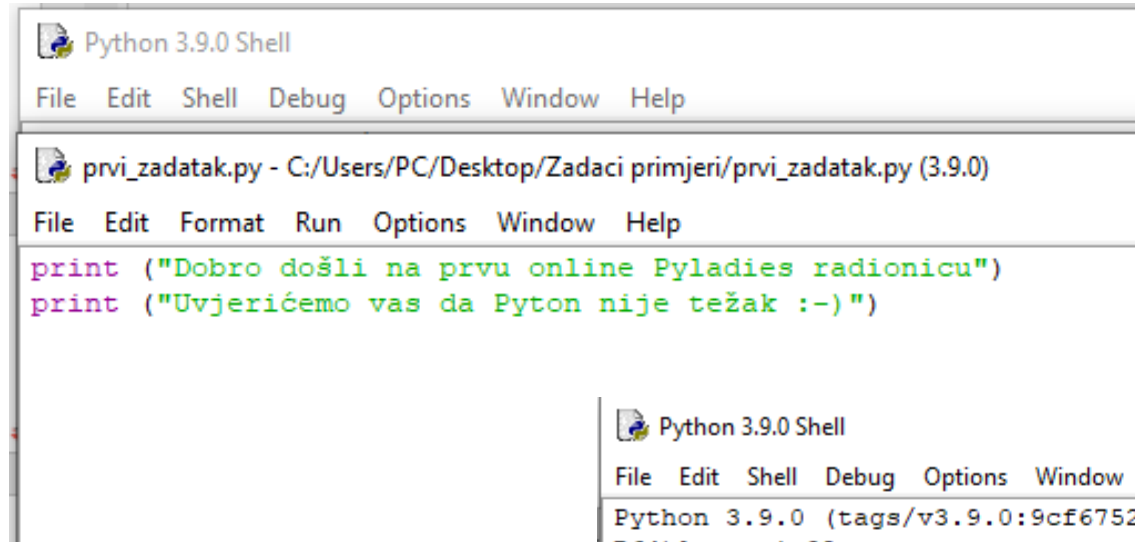
```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("Zdravo , svete!")
Zdravo , svete!
>>> |
```

RADNO OKRUŽENJE-IDLE EDITOR

- pokrenuti Python razvojno okruženje IDLE
- u razvojnom okruženju IDLE otvoriti novi projekat (opcija File/New)
- u editoru koji se pojavi otkucati željeni kod
- snimiti program u datoteku na lokaciji koju ste odabrali pod odgovarajućim imenom sa ekstenzijom .py (opcija File/Save as...)
- pokrenuti program (opcija Run/Run Module tj. F5 na tastaturi)



RADNO OKRUŽENJE-IDLE EDITOR



SINTAKSA

Program u jeziku Python je *niz naredbi*, čiji delimiter je kraj linije teksta.

Razmaci (Spaces)

Razmaci su značajni na početku linije, ali ne posle toga.

Dok u drugim programskim jezicima uvlačenje u kodu služi samo za bolju čitljivost, uvlačenje u Python-u je vrlo važno.

Python koristi uvlačenje za označavanje bloka koda.

Velika i mala slova

U Python-u print, Print, i PRINT su različite stvari. Znači velika i mala slova su važna. Za sada, držaćemo se malih slova, pošto većina Python naredbi koristi takva slova.



KOMENTARI

- Komentari se mogu koristiti za objašnjenje koda. Oni se prilikom izvršavanja koda ignorišu što omogućava upotrebu i za sprečavanje izvršavanja koda.
- Komentari u jednoj liniji počinju znakom “hash” # i protežu se do kraja linije.
- Komentari u više linija počinju i završavaju se sa tri jednostruka ili tri dvostruka navodnika.

```
# Ovo je komentar u jednoj liniji

"""
    Ovo je
    komentar u
    više
    linija
"""

# print ("Hello, World!")

print ("Hello, World!") # Ovo je komentar u jednoj liniji
```

Izvršiće se samo ovaj deo koda

Ispred znaka za komentar u više redova ne smije pisati nikakav kod tj. mora biti napisan na početku reda.



PROMENLJIVE

Promenljive su “skladišta” za čuvanje podataka.

Za razliku od drugih programskih jezika Python nema naredbu za deklarisanje promenljivih.

Promenljiva se kreira (definiše) kada joj se prvi put dodeli vrednost.



PROMENLJIVE

Ime promenljive sme sadržati samo

- Brojeve
- Velika i mala slova
- Donju crticu _
- Ne sme biti ključna reč
- Ime ne sme početi brojem
- Python razlikuje velika i mala slova

Neka od dozvoljenih imena:

jabuka, slova123, A4, ime_psa

Neka od nedozvoljenih imena:

ime psa, 123slova, jabuka#

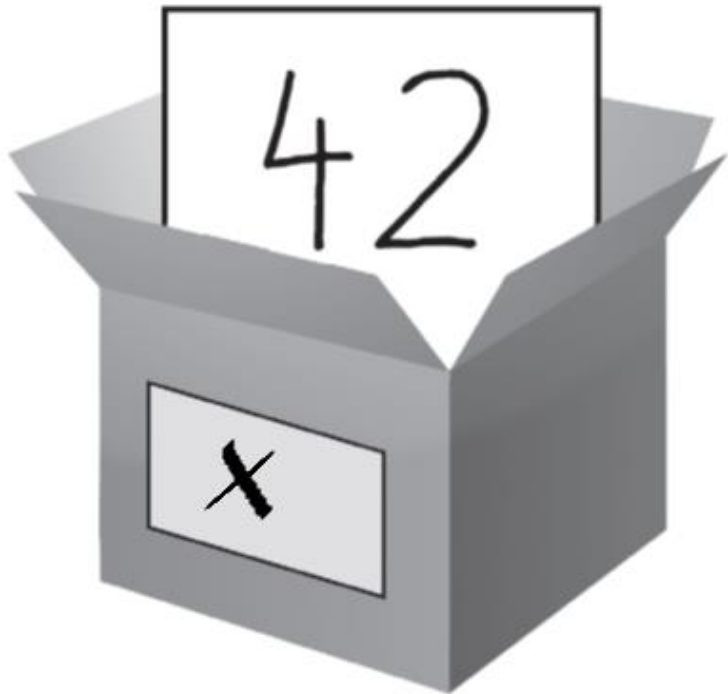
and def exec if not return
assert del finally import or try
break elif for in pass while
class else from is print yield
continue except global lambda raise



PROMENLJIVE - NAREDBA DODELE VREDNOSTI

Naredba dodele vrednosti je oblika

<ime promenljive> = <vrednost koju želimo da sačuvamo>



Promenljivu možete zamisliti kao kutiju sa nazivom u koju je postavljena vrednost

Izraz $x = 42$ znači da kažete programu:

„Promenljiva x sada ima u sebi vrednost celog broja 42“

OSNOVNI TIPOVI PROMENLJIVIH

Ceo broj (int)

Decimalni broj (float)

Logička vrednost (boolean)

Tekst (str tj. string)

Tip promenljive definisan je tipom dodeljene vrednosti, npr.

```
x = 1
```

```
x = 1.0
```

```
x = 2*x
```

```
ime = "Jovana"
```

```
pozdrav = "Zdravo " + ime
```



NIZOVI ZNAKOVA

Niz znakova (string) može se zapisati na nekoliko načina.

Upotrebom:

- jednostrukih navodnika

Npr. 'Zdravo!'

- dvostrukih navodnika

npr. "Zdravo!"

- trostrukih navodnika

npr. Omogućavaju pisanje niza znakova u nekoliko linija.



PRIMER – PROMENLJIVE DODELA VREDNOSTI

```
poruka= "Programiranje je lepo"  
pi = 3.14  
n = 5  
print (poruka)  
print (pi)  
print (n)
```



TIPOVI I VREDNOSTI

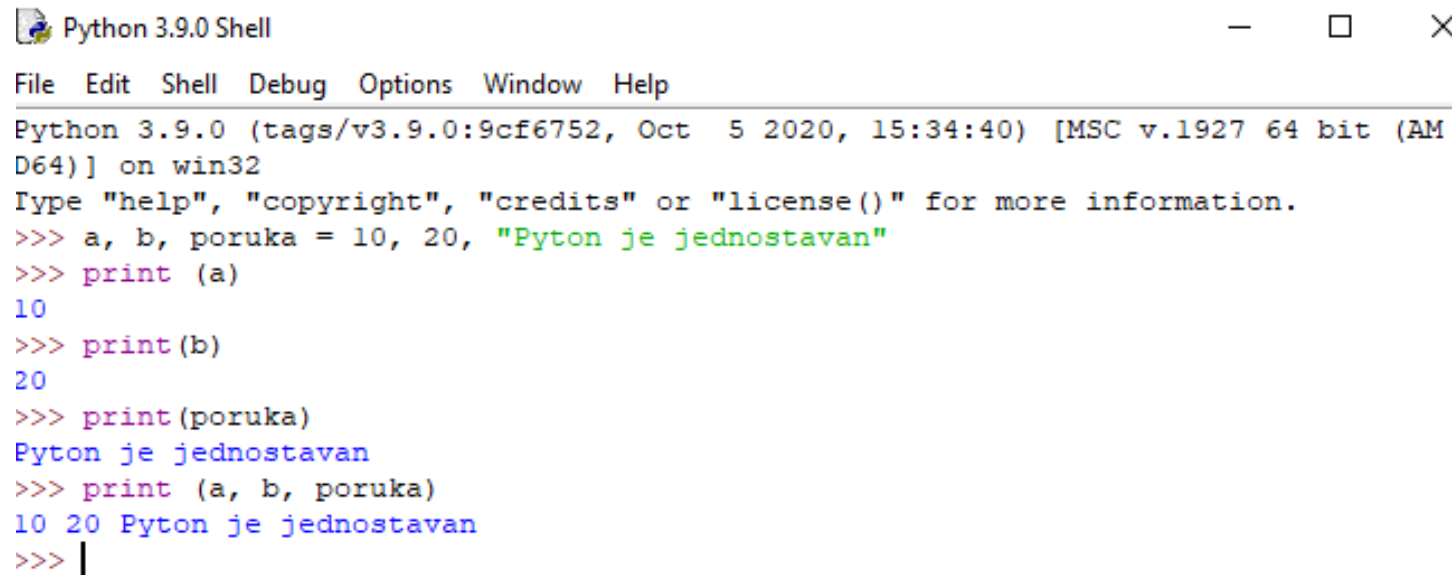
Dodelom vrednosti se određuje tip neke promenljive. Ukoliko nismo sigurni kog je tipa neki podatak, interpreter nam to može reći (dodamo sledeći kod)

```
print (type(poruka))  
print (type(pi))  
print (type(n))
```



PRIMER - VIŠESTRUKA DODELA VREDNOSTI

a, b, poruka = 10, 20, Pyton je jednostavan



```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> a, b, poruka = 10, 20, "Pyton je jednostavan"
>>> print(a)
10
>>> print(b)
20
>>> print(poruka)
Pyton je jednostavan
>>> print(a, b, poruka)
10 20 Pyton je jednostavan
>>> |
```



OPERATORI

Operatori se koriste za izvođenje operacija na promenljivim i vrednostima.

Aritmetički operatori se koriste nad numeričkim vrednostima tj. za izvođenje uobičajenih matematičkih operacija.

Kao i u matematici, u programiranju je potrebno voditi računa o prednostima operacija. Operacije množenja i deljenja imaju veći prioritet izvršavanja od sabiranja i oduzimanja.

Prilikom pisanja programa najčešće koristimo male zagrade kako bismo lakše sagledali prioritete.



ARITMETIČKI OPERATORI

Operator	Naziv	Upotreba	Rezultat za x=5 i y=2
+	Sabiranje ili spajanje stringova	$x+y$	7
-	Oduzimanje	$x-y$	3
*	Množenje	$x*y$	10
/	Deljenje	x/y	2.5
%	Modul (ostatak od deljenja)	$x\%y$	1
**	Stepenovanje	$x**y$	25
//	Celobrojno deljenje (decimalni deo se odbacuje)	$x//y$	2



OSNOVNE MATEMATIČKE OPERACIJE

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 2+2
4
>>> 3-3
0
>>> 4/2
2.0
>>> (3-1)+2
4
>>> 4**3
64
>>> 8%2
0
>>> 9//4
2
>>> 4*4*4
64
>>>
```



IZRAZI

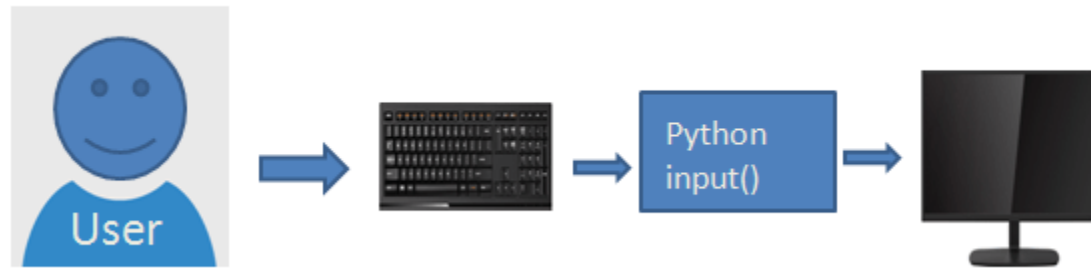
Izrazi su pravilna kombinacija *vrednosti* (konstanti), *promenljivih*, *funkcija* i *operatora*.

Izrazi nakon izračunavanja (evaluacije) daju rezultat određenog tipa, koji se može dodeliti nekoj promenljivoj

Rezultat može biti numerički, nenumerički ili logički (podtip int)



STANDARDNI ULAZ/IZLAZ



Funkcija `input()` čeka da korisnik ukuca tekst na tastaturi i pritisne **ENTER**.

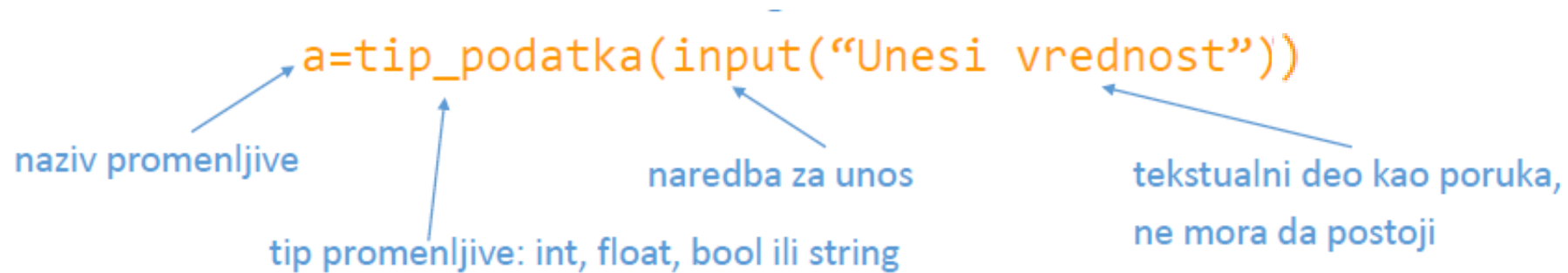
Funkcije uvek vraćaju neku vrednost kao rezultat izvršavanja

- `input()` kao rezultat izvršavanja vraća *string* koji je korisnik interaktivno uneo
- `print()` kao rezultat vraća vrednost `None`

STANDARDNI ULAZ/IZLAZ

Upisivanje tekstualne ili brojčane vrednosti je moguće pomoću naredbe `input()`. Sama naredba ništa ne znači ukoliko nije pridružena nekoj promenljivoj.

Tip prihvaćenih podataka je uvek string, ukoliko se ne zahteva drugačije.

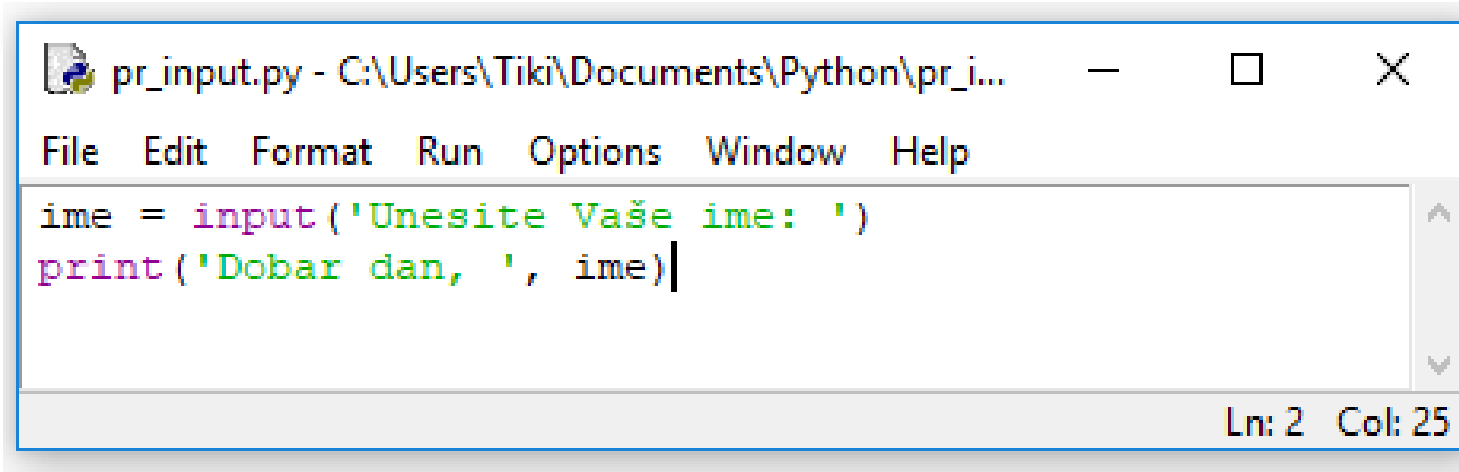


Funkcija za ispis je `print()`, na primer:

```
print ("Ovo je tekst ", "a ovo je broj ", 5)
```



ULAZ PODATAKA (INPUT)



The screenshot shows a window titled 'pr_input.py - C:\Users\Tiki\Documents\Python\pr_i...'. The menu bar includes 'File', 'Edit', 'Format', 'Run', 'Options', 'Window', and 'Help'. The code editor contains the following Python code:

```
ime = input('Unesite Vaše ime: ')\nprint('Dobar dan, ', ime)|
```

The status bar at the bottom right indicates 'Ln: 2 Col: 25'.

Svrha funkcija **input** je da traži da korisnik ukuca nešto i da se to preuzme (dodeli promenljivoj). Onaj deo koji je pod znacima navoda će biti prikazan korisniku kao uputstvo šta se od njega očekuje.

Uneseni tekst je uvek *string*, a konverzija u odgovarajući tip vrednosti vrši se pomoću ugrađenih funkcija, npr. `int()` i `float()`



FUNKCIJA PRINT()

Funkcija `print()` prikazuje vrednost niza unutar zagrada na ekranu.

Linija `print ('Pozdrav!')` znači „Odštampaj tekst u nizu 'Pozdrav!'“.

Kada Python izvrši ovu liniju, možete da kažete da je *pozvao* funkciju `print()` i da je vrednost niza prosleđena funkciji.

Vrednost koja je prosleđena funkciji naziva se *argument*. Videćete da navodnici nisu odštampani na ekranu. Oni samo označavaju gde je početak i kraj niza; nisu deo vrednosti niza.



ZADACI

zadatak1_o

Milena je na pijaci prvog dana kupila 10 jabuka, drugog dana 7, a trećeg 8 jabuka. Koliko je Milena u proseku kupovala jabuka? (napisati program)

Napomena: koristiti promenljive.

zadatak2_o

Miloš ima 10 klikera. Petar ima 2 puta više klikera od Miloša, a Nikola ima 5 klikera više od Miloša. Koliko ukupno klikera imaju Petar i Nikola?

(napisati program)

Napomena: koristiti promenljive.



ZADACI

zadatak3 o

Izračunati površinu kvadrata stranice 5cm, koristeći operator za stepenovanje. Dobijenu površinu podeliti sa 2 i ispisati posebno ceo deo broja, a posebno ostatak. (napisati program). Napomena: koristiti promenljive.

zadatak4 o

Napisati program koji za unete vrednosti (masu i visinu) izračunava index telesne mase (body mass index, bmi)

Masa u kilogramima (prirodan broj)

Visina u metrima (realan broj)

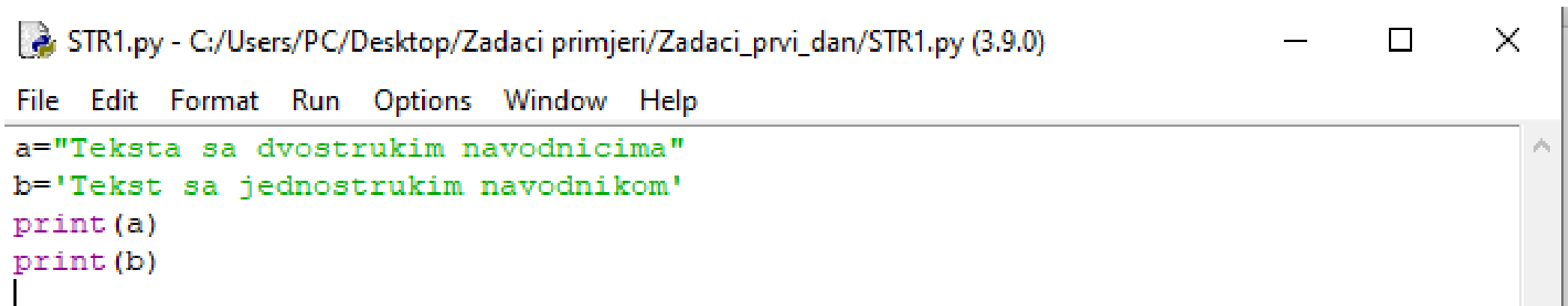
Indeks telesne mase ($bmi = \frac{masa}{visina^2}$)



STRINGOVI

RAD SA STRINGOVIMA

String je niz karaktera ograničen jednostrukim ili dvostrukim navodnicima



The screenshot shows a window titled "STR1.py - C:/Users/PC/Desktop/Zadaci primjeri/Zadaci_prvi_dan/STR1.py (3.9.0)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
a="Teksta sa dvostrukim navodnicima"  
b='Tekst sa jednostrukim navodnikom'  
print(a)  
print(b)  
|
```

Teksta sa dvostrukim navodnicima

Tekst sa jednostrukim navodnikom



RAD SA STRINGOVIMA

String mora početi i završiti se istim navodnicima

```
s = 'On je rekao, "Zdravo, svete!"'  
t = "Vise navodnika'''' ali i dalje ispravno."  
print(s)  
print(t)
```

On je rekao, "Zdravo, svete!"

Vise navodnika'''' ali i dalje ispravno.



SPAJANJE STRINGOVA

Dva stringa se spajaju u jedan korišćenjem operatora +

```
s1 = "Zdravo"  
s2 = "Svete"  
print(s1+s2)  
print(s1+' '+s2)  
print("Halo," + "gde si?")  
s1 = "Prvi string"  
s2 = ", drugi string"  
print(s1+s2)
```



KOPIRANJE STRINGOVA

```
s = 'Ha'  
print (s * 10)  
a = 'Ćao'  
print(a * 3)
```

HaHaHaHaHaHaHaHaHa

ĆaoĆaoĆao

Množenje nulom i negativnim brojem ne daje nikakav rezultat

Nije dozvoljeno množenje razlomljenim brojem i sabiranje sa celim brojem

```
print('Pozdrav' * 8.1)  
print('123' + 4)
```



PRIMER

- Ako želimo da šampamo dugačak red crtica, to možemo da uradimo na sledeći način:

```
print('_' *25)
```

GRANANJE

LOGIČKI OPERATORI

Operator	Naziv	Upotreba	Opis
<i>and</i>	Logičko I, konjukcija	izraz1 <i>and</i> izraz2	Rezultat je TRUE ako i samo ako oba izraza imaju vrednost TRUE, u ostalim slučajevima FALSE.
<i>or</i>	Logičko II, disjunkcija	izraz1 <i>or</i> izraz2	Rezultat je TRUE, ako je bar jedan izraz TRUE, inače daje FALSE.
<i>not</i>	Negacija	<i>not</i> izraz	Unarni operator kod kojeg je rezultat TRUE samo ako je izraz FALSE i obrnuto.



OPERATORI POREĐENJA

Operator	Naziv	Upotreba	Rezultat za x=5 i y=10	Rezultat za x=10 i y=10
<	Manje od	$x < y$	TRUE	FALSE
<=	Manje od ili jednako	$x \leq y$	TRUE	TRUE
>	Veće od	$x > y$	FALSE	FALSE
>=	Veće od ili jednako	$x \geq y$	FALSE	TRUE
!=	Različito	$x \neq y$	TRUE	FALSE
==	Jednako	$x == y$	FALSE	TRUE



PRIMERI LOGIČKIH I OPERATORA POREĐENJA

```
Python 3.9.0 Shell
File Edit Shell Debug Options Window Help
Python 3.9.0 (tags/v3.9.0:9cf6752, Oct 5 2020, 15:34:40) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> x,y=10,20
>>> x,y
(10, 20)
>>> x<y
True
>>> x<0 and y<0
False
>>> x>0 or y>1
True
>>> x>0 and not y>1
False
>>> 7>5
True
>>> 3!=3
False
>>> 3==3
True
>>> "Marko" > "Markovic"
True
>>> |
```



GRANANJE

Neke naredbe se izvršavaju samo ako je neki uslov ispunjen. Da bi se opisalo uslovno izvršavanje nekih naredbi koristi se naredba if

```
if uslov:  
    naredba_1  
    ...  
    naredba_k
```

```
if uslov:  
    naredba_1  
    ...  
    naredba_m  
else:  
    naredba_1  
    ...  
    naredba_n
```

```
if uslov1:  
    naredba_1  
    ...  
    naredba_m  
elif uslov2:  
    naredba_1  
    ...  
    naredba_n  
else:  
    naredba_1  
    ...  
    naredba_k
```

- Zgrade u uslovu nisu OBAVEZNE.
- Dvotačka je obavezna nakon navođenja uslova
- Naredba uslova će vratiti tačno/netačno
- Ako je tačan uslov koji se zadaje izvršiće naredba_1...



GRANANJE

```
if uslov:                # ako je uslov ispunjen
    naredba_1            #   izvrši naredbu 1
    ...                 #   ...
    naredba_k            #   izvrši naredbu k
```



GRANANJE

```
if uslov:                # ako je uslov ispunjen
    naredba_1             #   izvrši naredbu 1
    ...                  #   ...
    naredba_m             #   izvrši naredbu m
else:                    # u suprotnom:
    naredba_1             #   izvrši naredbu 1
    ...                  #   ...
    naredba_n             #   izvrši naredbu n
```

PRIMERI

- Za uneti broj ispisati da li je manji od 10

```
a = int(input("Unesi broj "))  
if a < 10:  
    print ('Broj je manji od 10')
```

Za uneti broj ispisati da li je paran ili neparan

```
if a%2 == 0:  
    print ('Broj je paran')  
else:  
    print ('Broj je neparan')
```



ZADACI

zadatak1_g

Unesi neki prirodan broj i ispiši da li je taj broj deljiv sa sedam, u suprotnom ispiši da nije.

zadatak2_g

Unesite dva broja. Saberite dva broja. Ako je zbir veći od 20 ispišite : **Suma je veća od 20** u suprotnom ispišite: **Suma je manja od 20**



PRIMER - BLOK KOMANDI

Za unete dužine stranica pravougaonika ili kvadrata (ukoliko su unete vrednosti jednake) izračunati površinu.

```
a=int(input("Unesite stranicu a: "))
```

```
b=int(input("Unesite stranicu b: "))
```

```
if a==b:
```

```
    p=a**2
```

```
    print("Povrsina kvadrata iznosi: ", p)
```

```
else:
```

```
    p=a*b
```

```
    print("Povrsina pravougaonika iznosi: ", p)
```

Više od
jedne
naredbe



LANČANI USLOVI

Primer: Odrediti odnos dva uneta broja

```
x = int(input("Unesi broj x "))
y = int(input("Unesi broj y "))
if x < y:
    print ('x je manje od y')
elif x > y:
    print ('x je vece od y')
else:
    print ('x i y su jednaki')
```



POREĐENJE STRINGOVA

```
ime= input()
if ime == "Marko":
    print("Da, moje ime je Marko")
if ime < "Marko":
    print("Uneto ime je ispred Marka")
elif ime > "Marko":
    print("Uneto ime je iza imena Marko")
else:
    print("Uneto ime je Marko")
```



PRIMER - KALKULATOR

Unetu operaciju primeniti na zadate vrednosti promenljivih **a** i **b**

```
a = 25
b = 5
c = input ('uneti operaciju: ')
if (c == '+'):
    print (a + b)
elif (c == '*'):
    print (a * b)
elif (c == '-'):
    print (a - b)
elif (c == '/'):
    print (a / b)
```



**PONAVLJANJE
(ITERACIJA)**

FOR PETLJA

PONAVLJANJE (ITERACIJA)

Ako je potrebno ispisati jednu rečenicu, brojeve ili naredbe više puta potrebno je dosta vremena.

Petlja je deo koda koji se ponavlja.

Upotrebom petlje štedite vreme, jer ne morate stalno da kucate jedno te isto.



PONAVLJANJE (ITERACIJA)

Naredba ponavljanja u jeziku Python omogućava višestruko izvršavanje određenog bloka naredbi.

Broj ponavljanja može biti unapred određen (naredba for) ili se može postaviti logički uslov za izvršavanje ponavljanja

Logički uslov u jeziku Python proverava se *pre* prvog izvršavanja bloka naredbi (naredba while)



NAREDBA FOR

Ponavljanje zadani broj puta:

```
for promenljiva in skup_vrednosti:  
    telo_petlje
```

Telo_petlje (blok naredbi) se izvršava za svaku pojedinačnu vrednost iz zadanog skupa vrednosti

Skup vrednosti može biti eksplicitno zadan ili proizveden, npr. pomoću funkcije range()

funkcija range(...) omogućava generisanje skupa vrednosti za koje se vrši svako ponavljanje, npr.

range(1, 10, 1) generiše skup vrednosti 1, 2, 3, 4, 5, 6, 7, 8, 9

range(10,1,-2) generiše skup vrednosti 10,8,6,4,2



UGRAĐENA FUNKCIJA RANGE()

- Funkcija range generiše sekvencu celih brojeva:

- `range(start, stop, korak)`

Početna
vrednost
brojača

Završna vrednost
brojača
(nije uključen)

Tipične forme upotrebe funkcije su:

```
range(10) # skup vrednosti 0,1,...,9  
range(1,11) # skup vrednosti 1,2,...,10  
range(0,10,3) # skup vrednosti 0,3,6,9  
range(0,-10,-1) # skup vrednosti 0,-1,-2,...,-9
```



NAREDBA FOR - PRIMER

```
for i in range(5):  
    print("Zdravo")
```

Zdravo

Zdravo

Zdravo

Zdravo

Zdravo

```
for i in range(5):  
    print(i)
```

0

1

2

3

4



NAREDBA FOR - PRIMER

```
for i in range(1,6):  
    print (i)
```

```
for i in range(25,4,-5):  
    print (i)
```

Izlaz

1

2

3

4

5

25

20

15

10

5



NAREDBA FOR - PRIMER

Primer

```
for x in "banana":  
    print(x)
```

Izlaz

b
a
n
a
n
a

PRIMER

Šta je rezultat sledećeg koda?

a)

```
for i in range(3,9):  
    print (i)
```

b)

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

c)

```
for x in "Informatika":  
    print(x)
```

ZADACI

zadatak1 for

Ispisati samo parne brojeve brojevnog niza od prvih 10 brojeva.

zadatak2 for

Iskoristite for petlju da šampate pravougaonik (prikazan ispod). Omogućite korisniku da odredi dužinu i širinu pravougaonika, tako što će te vrednosti uneti na početku programa.

Uputstvo: `print('*'*10)` štampa 10 zvezdica

