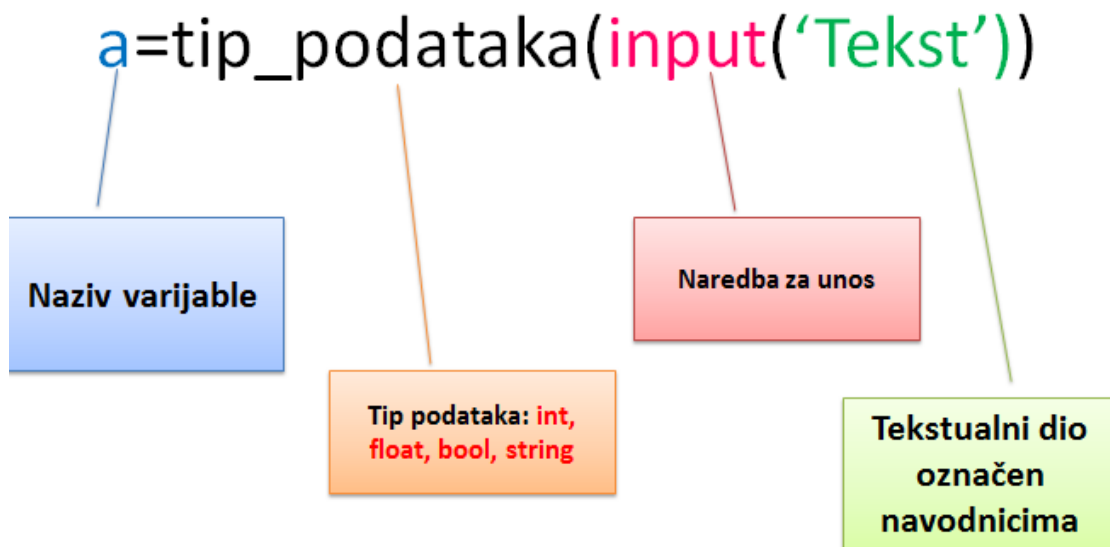


# NAREDBE UNOSA I ISPISA

---

## INPUT

Da bi mogli manipuirati podacima i nad njima vršiti različite operacije, potrebno je dobiti podatke nad kojima će se moći vršiti različite operacije. Kako bi omogućili upisivanje tekstualne ili broježane vrijednosti u Python programski jezik, potrebno je upotrijebiti naredbu **input**.



Naredba **input** sama za sebe ne nači ništa. Nju moramo pridružiti nekoj varijabli kao običnu broježanu ili tekstualnu vrijednost. Navikli smo različitim varijablama pridruživati različite vrijednosti no uobičajenim pridruživanjem vrijednosti nekoj varijabli (na primjer: `a=5`), odredili smo vrijednost unutar skripte programskog jezika. Kako bi omogućili korisniku da preko tipkovnice unosi podatke u Python shell sučelje za izvršavanje programa, potrebno je upotrijebiti upravo naredbu **input**. Kao što smo dosad pridruživali neku vrijednost varijabli pomoću znaka pridruživanja „=“, sada ćemo naredbu **input** zajedno s njezinom sintaksom pridružiti nekoj varijabli **a**. Kako u Pythonu unaprijed nije zadan tip podataka, on svaki unos interpretira kao niz znakova, a ne broježanu vrijednost. Stoga prije naredbe `input`, potrebno je odrediti tip podataka za rad s brojevima (**int**, **float**).

Primjeri unosa vrijednosti primjenom različitih tipova podataka:

```
>>> b=float(input('unesi realan broj'))
unesi realan broj3.14
>>> b
3.14
```

```
>>> a =int(input('unesi cijeli broj:'))
unesi cijeli broj:100
>>> a
100
```

```
>>> c =input('unesi ime:')
unesi ime:Ana
>>> c
'Ana'
```

Svaki tip podataka nužno zahtijeva pisanje zagrada.

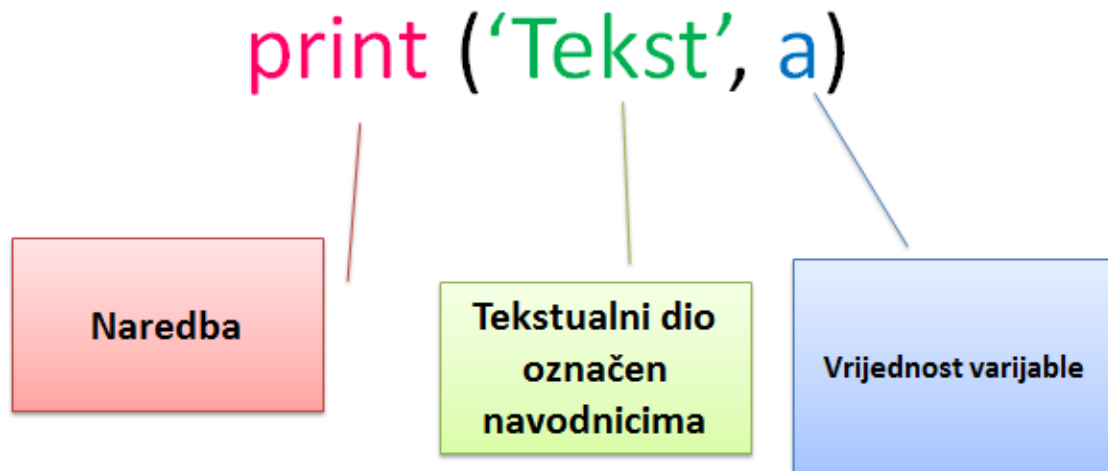
```
>>> ime=input()
asd
>>> ime
'asd'
>>> a=input()
100
>>> a
'100'
```

u varijablu **ime** je spremljen niz znakova 'asd'  
u varijablu **a** je spremljen niz znakova '100'

Naredba input bez teksta

U slučaju da želimo raditi sa znakovnim nizovima, onda nije potrebno naredbi **input** pridruživati tip podataka. Nakon tipa podataka (ukoliko za to postoji potreba), dodaje se ključna riječ **input**. Iznimno je bitno za naredbu **input** pisanje zagrada. Unutar tih zagrada, moguće je pisati tekst. Nije nužno pisati tekst, ali pomoću teksta jasnije možemo korisniku odrediti što se od njega traži: da li se traži unos brojsane vrijednosti ili znakovnog niza.

## PRINT



Naredba za ispisivanje vrijednosti međutim ne predstavlja neko veliko odstupanje od sintakse naredbe za unos. Naredba za unos aktivira se ključnom riječi **print**. Nakon te riječi otvaraju se i zatvaraju zagrade. Unutar zagrada, moguće je ispisati tekst, ali i vrijednosti varijabli.

**Glavna je razlika što se tekst uvijek stavlja unutar navodnika.** To je znak Pythonu da se radi o poruci koju korisnik želi ispisati. Vrijednost varijable se ispisuje na način da se ta varijabla pozove unutar naredbe **print** na način da se upiše njezino slovo na odgovarajuće mjesto. Sve tekstualne vrijednosti unutar naredbe `print` kao i vrijednosti varijabli, moguće je odvojiti zarezom „`,`“. Dakle naredba ispisivanja omogućuje:

- ispis vrijednosti na standardni izlaz (zaslon monitora)
- višestruke vrijednosti odvojene zarezom
- **print** sam dodaje razmak između višestrukih vrijednosti

### Primjer naredbe `print` u Pythonu

```
>>> print('ovo je neki tekst', 'a ovdje je i broj', 33)
ovo je neki tekst a ovdje je i broj 33
```

# ZADACI

## Zadatak 1:

- Upisati jednu riječ i umnožiti je 5 puta

## Primjer ispisa:

```
Unesite 1. rijec'python'  
python python python python python
```

## Rješenje:

```
a=input("Unesite 1. rijec")  
  
c=(a+' ') *5  
|  
print(c)
```

U rješenju programa, primijenjen je razmak ' ' koji se označava jednostrukim navodnicima.

### Zadatak 2:

- Upišite dvije riječi i spojite ih u niz.

### Primjer ispisa:

```
Unesite 1. rijec 'python'  
Unesite 2. rijec' je super'  
python je super  
>>> |
```

### Rješenje:

```
a=input("Unesite 1. rijec")  
b=input("Unesite 2. rijec")  
  
c=a+' '+b  
print(c)
```

### Zadatak 3:

- Unesite troznamenkasti broj
- Kod ispisa ispisati:
  - Broj koji unosite
  - Broj u binarnom brojevnom sustavu (baza 2)

### Primjer ispisa:

```
====  
Unesite troznamenkasti broj325  
( 'Vas unos je', 325, 'Binarni broj je', '0b101000101')  
>>> |
```

### Rješenje:

```
a=int(input("Unesite troznamenkasti broj"))  
  
b=bin(a)  
  
print("Vas unos je",a,"Binarni broj je", b)
```

#### Zadatak 4:

- Unesite znak preko tipkovnice (slovo, brojku, specijani znak) i ispišite ASCII kod tog znaka

#### Primjer ispisa:

```
Unesite znak';'  
(vase znak:', ';', 'ASCII kod', 59)  
>>> |
```

---

#### Rješenje:

```
a=input("Unesite znak")  
  
b=ord(a)  
  
print("vase znak:",a, "ASCII kod",b)
```

POMOĆ:  
**ord(a)**

Kod ovog zadatka, primijenjena je posebna naredba `ord()` pomoću koje se dohvaća vrijednost nekog znaka napisana u ASCII kodnom sustavu.

#### Zadatak 5:

- Unesite vrijednost varijable i ispišite memorijsku adresu te varijable

#### Primjer ispisa:

```
Unesite vrijednost a:25  
(mem. lokacija je:', 38368808)
```

POMOĆ:  
**id(a)**

#### Rješenje:

```
a=int(input("Unesite vrijednost a:"))  
  
print("mem. lokacija je:", id(a))
```

---

U rješenju je primijenjena posebna naredba `id()` kojom je moguće dohvatiti adresu memorijske lokacije gdje je trenutno spremljen podatak u memoriji.

### Zadatak 6:

- Zbrojite dva broja i rezultat ispišite kao decimalni broj

### Primjer ispisa:

====

```
Prvi broj 2.6
Drugi broj5.8
('Zbroj je:', 8.4)
|
```

### Rješenje:

```
a=float(input("Prvi broj"))
b=float(input("Drugi broj"))
c = a+b
print("Zbroj je:", c)
|
```

### Zadatak 7:

- Unesite dva broja i izračunajte njihovu sumu, razliku, količnik, umnožak

### Primjer ispisa:

```
Unesite 1. broj 2
Unesite 2. broj 3
('Zbroj je:', 5)
('Razlika je :', -1)
('Umnozak je:', 6)
('Kolicinik je:', 0)
```

### Rješenje:

```
a=int(input("Unesite 1. broj"))
b=int(input("Unesite 2. broj"))
c=a+b
d=a-b
e=a*b
f=a/b
print("Zbroj je:", c)
print("Razlika je :", d)
print("Umnozak je:",e)
print("Kolicinik je:",f)
```



### Zadatak 8:

- Unesite dva broja i izračunajte njihovu **sumu, razliku, količnik, umnožak**
- Sumu ispisati u **binarnom sustavu**
- Razliku ispisati u **heksadekadskom sustavu**
- Umnožak ispisati u **oktalnom sustavu**
- Količnik ispisati u dekadskom sustavu

### Primjer ispisa:

```
Unesite 1. broj 25
Unesite 2. broj 5
('Zbroj je:', '0b11110')
('Razlika je :', '0x14')
('Umnozak je:', '0175')
('Kolicinik je:', 5)
```

### Rješenje:

```
a=int(input("Unesite 1. broj"))
b=int(input("Unesite 2. broj"))

c=a+b
d=a-b
e=a*b
f=a/b

print("Zbroj je:", bin(c))
print("Razlika je :", hex(d))
print("Umnozak je:", oct(e))
print("Kolicinik je:", f)
```

U rješenju ovog programskog primjera, korištene su naredbe **bin()**, **hex()**, **oct()** kojima omogućavamo da se neki broj u dekadskom brojevnom sustavu pretvori u binarni, heksadekadski i oktalni brojevni sustav.

### Zadatak 9:

- Napisati program koji ispisuje rezultat na temelju izraza:  $x=b^2-4ac$

### Primjer ispisa:

```
1.broj2
2.broj3
3.broj4
('Rezultat', -23)
```

### Potenciranje u Pythonu

**a\*\*2**

### Rješenje:

```
a=int(input("1.broj"))
b=int(input("2.broj"))
c=int(input("3.broj"))

x=b**2-4*a*c

print("Rezultat", x)
```

### Zadatak 10:

- Izračunati aritmetičku sredinu sedam brojeva
- NAPOMENA: u jednoj liniji koda napraviti unos vrijednosti 7 varijabli

### Primjer ispisa:

```
Unesi niz brojeva1,2,3,4,5,6,7
('aritm.sredina je:', 4)
```

### Rješenje:

```
a,b,c,d,e,f,g=input("Unesi niz brojeva")

sredina=(a+b+c+d+e+f+g)/7

print("aritm.sredina je:", sredina)
```

U rješenju ovog programa, vidljivo je da nije naznačen broječni tip podataka. Iako je naredba unosa napisana bez pripadajućeg tipa podataka, python je uspješno uspio izračunati aritmetičku sredinu. U istom retku, napravljen je upis u 7 različitih varijabli.