

# Programski jezik Python



# Python

- 1970. Niklaus Wirth – Pascal
- 1989. Nizozemac Guido van Rossum – Python-naziv po grupi Monty Python's Flying Circus
- Uz proceduralni pristup, objektno usmjereni pristup programiranju
- Besplatan
- <http://www.python.org/download/>
- Python jedan od tri službena programska jezika u Google

# Programsko okruženje za pripremanje programa - IDLE



- -Guido van Rossum nazvao IDLE
- **I**ntegrated **D**eve**L**oment **E**nvironment (prezime komičara iz grupe Monthly Python – Eric Idle)
- Prozor koji čini interaktivno sučelje s Pythonom

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

Sučelje Pythona

# Python



- Interaktivno sučelje
- Traka izbornika sučelja: File, Edit, Shell, Debug, Options, Windows, Help
- Help, IDLE Help pojedine aktivnosti
- Python Docs – upućuje na opsežnu dokumentaciju
- Na početku novog reda >>> znak upita (prompt)
- Sučelje sa znakom upita traži od korisnika odgovor
- Sučelje omogućuje interaktivni rad

# Jednostavni tipovi podataka



- Osnovni tipovi podataka: cijeli broj, broj s pomičnom točkom, logički ili Booleov tip, znakovni niz ili string
- `>>> 120`
- `120`
- Utipkana vrijednost se ispisuje plavom bojom
- Boju ispisa – Options, Configure IDLE

# Cijeli brojevi



- Naziv za cjelobrojni tip podataka u Pythonu je int
- U Pythonu nije ograničen broj znamenaka cijelog broja

```
>>> 27
27
>>> -302
-302
>>> 1234567890123456781234567812345678
1234567890123456781234567812345678
>>>
```

# Cijeli brojevi



- Binarni brojevni sustav

```
>>> 0b00001010
10
>>> 0b11111111
255
```

- Binarni zapis pretvara u dekadski
- Heksadekadski brojevni sustav

```
>>> 0x0A
10
>>> 0xFF
255
|
```

# Cijeli brojevi



- Cijeli broj pretvara u binarni

```
>>> bin(15)
'0b1111'
>>> bin(240)
'0b11110000'
```

- Cijeli broj pretvara u heksadekadski

```
>>> hex(15)
'0xf'
>>> hex(240)
'0xf0'
```



# Zapisivanje realnih brojeva



```
>>> 35.  
35.0  
>>> .0035  
0.0035  
>>> 0.00035  
0.00035  
>>> 0.000035  
3.5e-05  
>>> 0.0000356789  
3.56789e-05  
>>> 1e4  
10000.0  
>>> 1e8  
100000000.0  
>>> 1e32  
1e+32
```

# Zapisivanje realnih brojeva



- Norma IEEE 754
- Vrijednosti  $5.0 \cdot 10^{-324}$  do  $1.7 \cdot 10^{308}$
- Broj koji ima eksponent veći od +308 je za Python veliki broj i zapisuje se kao inf (beskonačno)
- Mali broj prikazuje se kao nula
- Broj `1.2345678901234567890e-323` u Pythonu `0.0`

# Logički ili Booleov tip podataka



- U Pythonu se logički tip naziva bool
- Poprima dvije vrijednosti: True i False
- True i true nije jednako
- `int(True)` 1      `int(False)` 0
- `bool(1)` True      `bool(0)` False
- `bool(100)` True      `bool(-100)` True

# Znakovni niz - string



- Osnovni tip podataka za prikaz teksta
- `str` - naziv u Pythonu
- Vrijednost se obilježava jednostrukim ili dvostrukim navodnim znacima na početku i kraju niza
- Dvostruki navodnik se ne smije pisati kao dva uzastopna jednostruka navodnika

# Posebni znaci za oblikovanje teksta



| Posebni znak     | Opis djelovanja               |
|------------------|-------------------------------|
| <code>\n</code>  | Prijelaz u novi red           |
| <code>\t</code>  | Tabulator                     |
| <code>\\</code>  | Ispisati lijevo ukošenu crtu  |
| <code>\'</code>  | Ispisati jednostruki navodnik |
| <code>\''</code> | Ispisati dvostruki navodnik   |

```
>>> print('prvi redak\n drugi redak\n treci redak')
prvi redak
drugi redak
treci redak
```

```
>>> print('prvi redak\t drugi redak\t treci redak')
prvi redak      drugi redak      treci redak
```

# Kodovi ASCII i Unicode



- Funkcija `ord()` , `hex()`, `bin()`

```
>>> print('A', ord('A'), hex(ord('A')), bin(ord('A')))
A 65 0x41 0b1000001
>>> |
```

```
>>> print('Č', ord('Č'), hex(ord('Č')), bin(ord('Č')))
Č 268 0x10c 0b100001100
>>> |
```

UTF-8 kodu kodiranje naših dijakritičkih znakova obavlja s dva bajta

Sada su veliko i malo slovo jedno do drugog

# Python kao veliki kalkulator – aritmetički izrazi



| Operator | Opis djelovanja              |
|----------|------------------------------|
| +        | Zbrajanje                    |
| -        | Oduzimanje ili negacija      |
| *        | Množenje                     |
| **       | Potenciranje                 |
| /        | Dijeljenje                   |
| //       | Cjelobrojno dijeljenje       |
| %        | Računanje ostatka dijeljenja |

# Python kao veliki kalkulator – aritmetički izrazi



```
>>> 2+3
5
>>> 2-3
-1
>>> 3*4
12
>>> 4/2
2.0
>>> -7
-7
```

```
>>> 17//3
5
>>> 17%3
2
>>> 18//3
6
>>> 18%3
0
>>> 5//7
0
>>> 5%7
5
```



# Python kao veliki kalkulator – aritmetički izrazi



- Koliko je dana, sati, minuta i sekundi 1 000 000 sekundi

```
>>> ((11*24 + 13)* 60 + 46)* 60 + 40  
1000000
```

| Operator    | Značenje simbola              |
|-------------|-------------------------------|
| **          | Potenciranje                  |
| -           | Negacija                      |
| *, /, //, % | Množenje, dijeljenje, ostatak |
| +, -        | Zbrajanje, oduzimanje         |

# Primjeri izraza u Pythonu



```
>>> 3+12/3*(60-20)
163.0
>>> 3 + 12/3*60 -20
223.0
>>> 3+12/(3*60)-20
-16.933333333333334
>>> 3 + 12//3 * 60 -20
223
>>> 2 + 12//(3*60)-20
-18
>>> |
```

# Primjeri izraza u Pythonu



- Python pamti zadnju izračunatu vrijednost
- Koristimo u sljedećem koraku , pišemo znak `_`

```
>>> 3
3
>>> _+4
7
>>> _+5
12
>>> _+6
18
```

# Varijable i znak pridruživanja



- = nazivamo znakom pridruživanja

```
>>> x= 1000
>>> x=x+1
>>> print (x)
1001
>>> |
```

- Ugrađene funkcije: print(), bin(), hex(), int(), ord(),...
- abs(-3)    3        abs(2.4)    2.4    abs(-4.7)    4.7

# Varijable i znak pridruživanja



- Ugrađena funkcija `divmod()`

```
>>> x=170
>>> y= 40
>>> kolicnik, ostatak =divmod(x,y)
>>> print(kolicnik, ostatak)
4 10
```

```
>>> sekunde = 1000000
>>> minute, sekunde = divmod(sekunde,60)
>>> sati, minute = divmod(minute,60)
>>> dani,sati = divmod(sati,24)
>>> print(dani,sati,minute,sekunde)
11 13 46 40
```

# Zadaci za ponavljanje

1. Nabroji osnovne tipove podataka u Pythonu.
2. Dekadni zapis broja 1011 dobit ćemo naredbom:
  - a. `b00001011`
  - b. `ob00001011`
  - c. `00001011b`
  - d. `00001011bo`
3. Koje od sljedećih prefiksa trebamo upisati ispred heksadekadnog zapisa broja kako bismo dobili pripadni dekadni zapis tog broja?
  - a. `x`
  - b. `o`
  - c. `ox`
  - d. `xo`

# Zadaci za ponavljanje

4. Pythonova funkcija koja će vratiti binarni zapis dekadnog broja je \_\_\_\_\_, dok heksadekadni zapis dekadnog broja vraća funkcija \_\_\_\_\_.
5. Koji je naziv standarda za zapis realnih brojeva u računalu?
6. Naziv logičkog tipa u Pythonu je \_\_\_\_\_.
7. Osnovne logičke vrijednosti su: \_\_\_\_\_ i \_\_\_\_\_.
8. Odredi vrijednosti sljedećih izraza: a. `int(False)`  
b. `int(True)` c. `bool(0)` d. `bool(5)` e. `bool(-10)`

# Zadaci za ponavljanje



9. Kako nazivamo znakovni tip u Pythonu?

10. Tekst ćemo u Pythonu pisati unutar znakova \_\_\_\_\_ ili \_\_\_\_\_.

11. Slijedećim posebnim znakovima pridruži pripadne opise:

| Posebni znak |  | Opis djelovanja               |
|--------------|--|-------------------------------|
| \n           |  | Ispisati jednostruki navodnik |
| \t           |  | Prijelaz u novi redak         |
| \'           |  | tabulator                     |

12. Funkcija ord() vraća \_\_\_\_\_.



# Zadaci za ponavljanje

13. Odredi vrijednost sljedećih izraza:

a.  $2 + 7$     b.  $-5 + 2$     c.  $6/2$     d.  $10/4$     e.  $8//4$

f.  $8//5$     g.  $9\%5$     h.  $2 + 3*7$     i.  $3**2*2$

j.  $7*3\%5$     k.  $15// 8 \%3 + 2 ** 3 \% 4$

l.  $15 \% 8 * 3 // 4 * 2 \% 6$

14. Što će biti ispisano na zaslonu nakon sljedećeg niza naredbi napisanih u Pythonu?

$2 + 3$

$_{-} -1$

$_{-} *4$

$_{-} /7$

# Zadaci za ponavljanje

15. Odredi vrijednost varijable  $t$  nakon izvođenja sljedećeg niza naredbi ako je vrijednost varijable  $t$  postavljena na 10 ( $t = 10$ ) ?

$$r = 2 * t$$

$$t = r \% 7$$

$$t = r + t$$

16. Koja će biti vrijednost varijable  $k$  nakon sljedećeg niza naredbi?

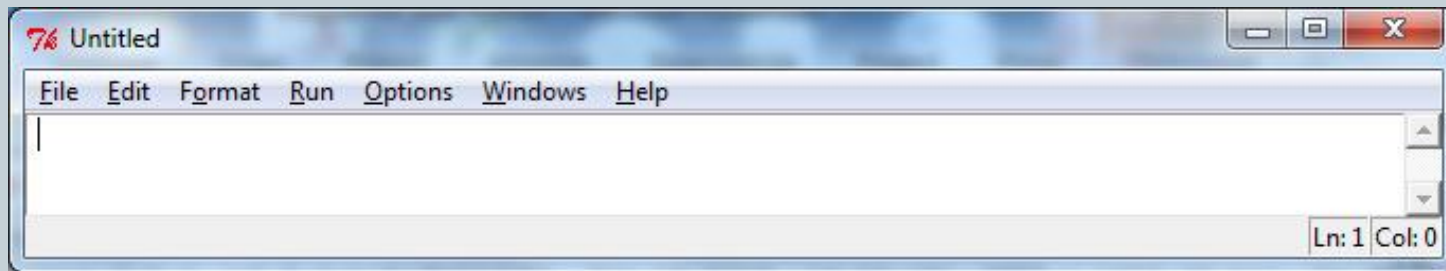
$$a, b = \text{divmod}(48, 10)$$

$$k = a + b$$

# Priprema i izvođenje programa

## Interaktivni i uređivački dio sučelja IDLE

- Programi u Pythonu, novi prozor – uređivački dio sučelja IDLE, otvaramo ga da u interaktivnom sučelju (Python Shell), File -> New Window ili (Ctrl) + (N)
- Izgled prozora



### Uređivački dio sučelja IDLE

- U traci izbornika nalaze se nazivi padajućih izbornika: **File, Edit, Format, Run, Options, Windows i Help.**

# Priprema i izvođenje programa



- Help -> IDLE Help – opis funkcija pojedinih naredbi.
- File -> Save as pohranjujemo napisani program.
- Naziv programa te iza njega sufiks **.py**
- Modul – datoteka koja ima sufiks
- Open – otvaramo program
- Program se pokreće **Run** i naredbom **Run Module** ili **F5**

A screenshot of a Python IDE window. The title bar reads "7% primjer\_1.py - C:/Python33/primjer\_1.py". The menu bar includes "File", "Edit", "Format", "Run", "Options", "Windows", and "Help". The code editor contains two lines of Python code: "#Program pozdrav od Pythona - primjer\_1" and "print('Lijep pozdrav od Pythona!')". The status bar at the bottom right shows "Ln: 2 Col: 33".

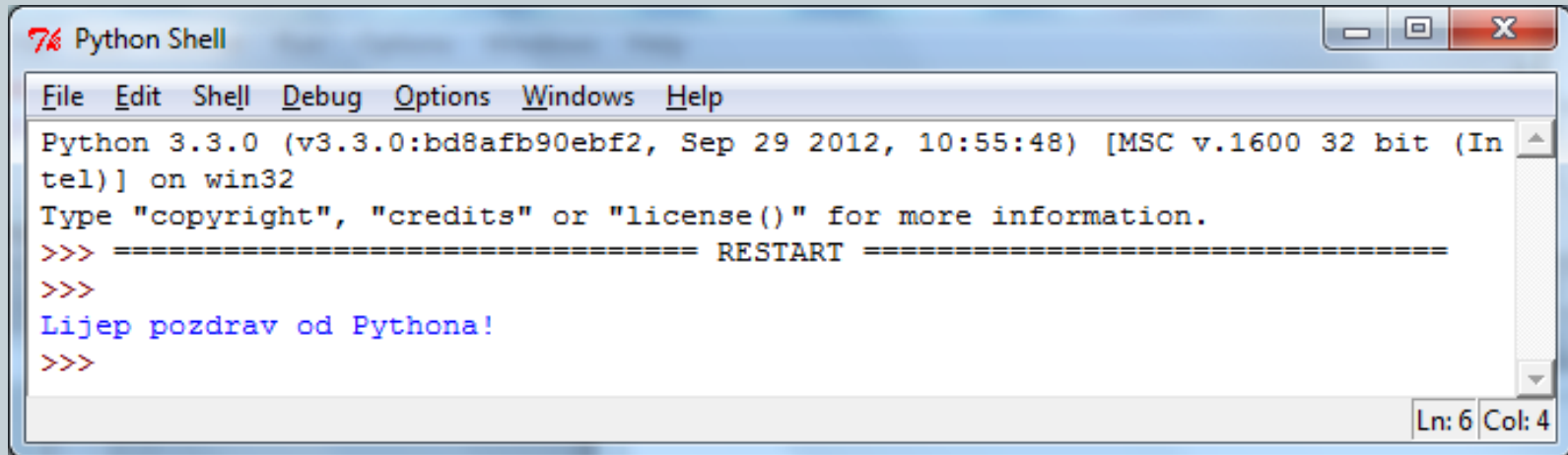
```
7% primjer_1.py - C:/Python33/primjer_1.py
File Edit Format Run Options Windows Help
#Program pozdrav od Pythona - primjer_1
print('Lijep pozdrav od Pythona!')
Ln: 2 Col: 33
```

Prozor pohranjenog programa primjer\_1

# Priprema i izvođenje programa



- # označivanje komentara
- Nakon pokretanja programa



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Lijep pozdrav od Pythona!
>>>
Ln: 6 Col: 4
```

## Ispis programa

- Funkcija `print()` služi za ispisivanje sadržaja na zaslon monitora.

# Priprema i izvođenje programa



- **input()** - za unošenje sadržaja

```
7% primjer_2.py - C:/Python33/primjer_2.py
File Edit Format Run Options Windows Help
#Program za osobni pozdrav
ime= input('Molim utipkajte svoje ime: ')
print(ime, ', lijep pozdrav od Pythona!', sep='')
Ln: 3 Col: 47
```

sep=", između ispisanih sadržaja neće biti razmaka

```
7% Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Molim utipkajte svoje ime: Dominik
Dominik, lijep pozdrav od Pythona!
>>>
Ln: 7 Col: 4
```

# Formatirani ispis



- `print('st_1{br_1}st_2{br_2},...,st_n{br_n}'.format(v_1,v_2,...,v_n))`

pri čemu vrijedi:

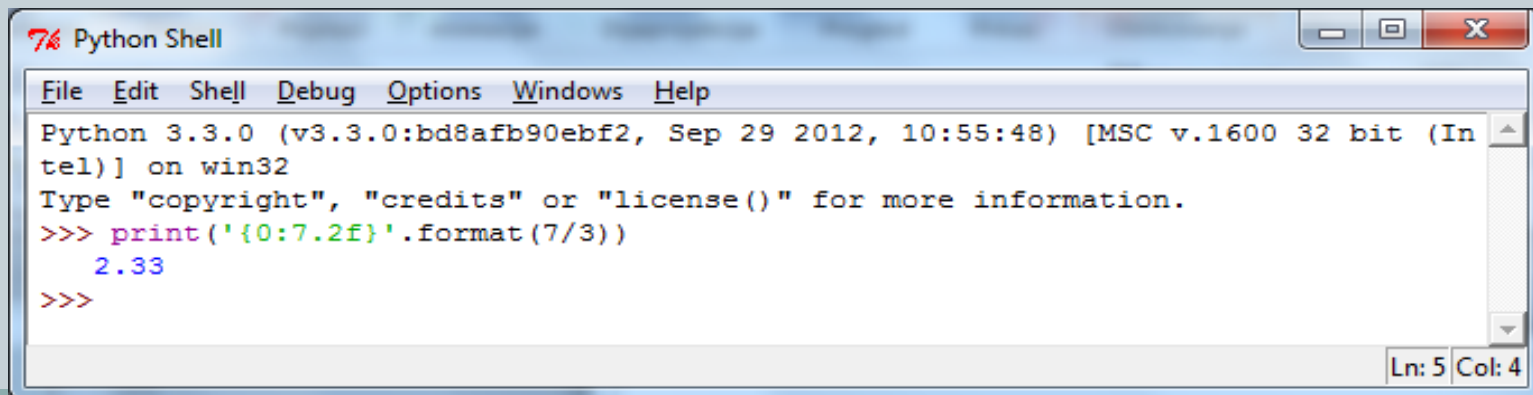
- `st_1, st_2, ..., st_n` - proizvoljni tekst,
- `v_1, v_2, ..., v_n` - popis varijabli koje želimo ispisati,
- `br_1, br_2, ..., br_n` - predstavlja redne brojeve varijabli unutar popisa.

```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Zbroj brojeva {0} i {1} je: {2}'.format(3,4,3+4))
Zbroj brojeva 3 i 4 je: 7
>>> print('Zbroj brojeva {} i {} je: {}'.format(3,4,3+4))
Zbroj brojeva 3 i 4 je: 7
>>> print('Svojstvo brojeva {0} i {1} prema kojem je {0} + {1} = {1} i {0} nazivamo
komutativnošću'.format(3,4))
Svojstvo brojeva 3 i 4 prema kojem je 3 + 4 = 4 i 3 nazivamo komutativnošću
>>> print('{0} {2} {1} = {3}'.format(3, 4, '+', 3+4))
3 + 4 = 7
>>>
```

Ln: 11 Col: 4

# Formatirani ispis

- d – ako ispisujemo cijeli broj,
- f – ako ispisujemo realni broj,
- s – ako ispisujemo string.
- Ispis cijelih brojeva moguće je formatirati naredbom: `{nd}`, n neki prirodni broj ili 0, označava broj na koliko će se mjesta ispisati određeni prirodni broj.
- `{n.mf}`, n ukupni broj mjesta na koji će se realni broj ispisati (uključujući i decimalnu točku), m – broj znamenki iza decimalne točke koje će biti ispisane.

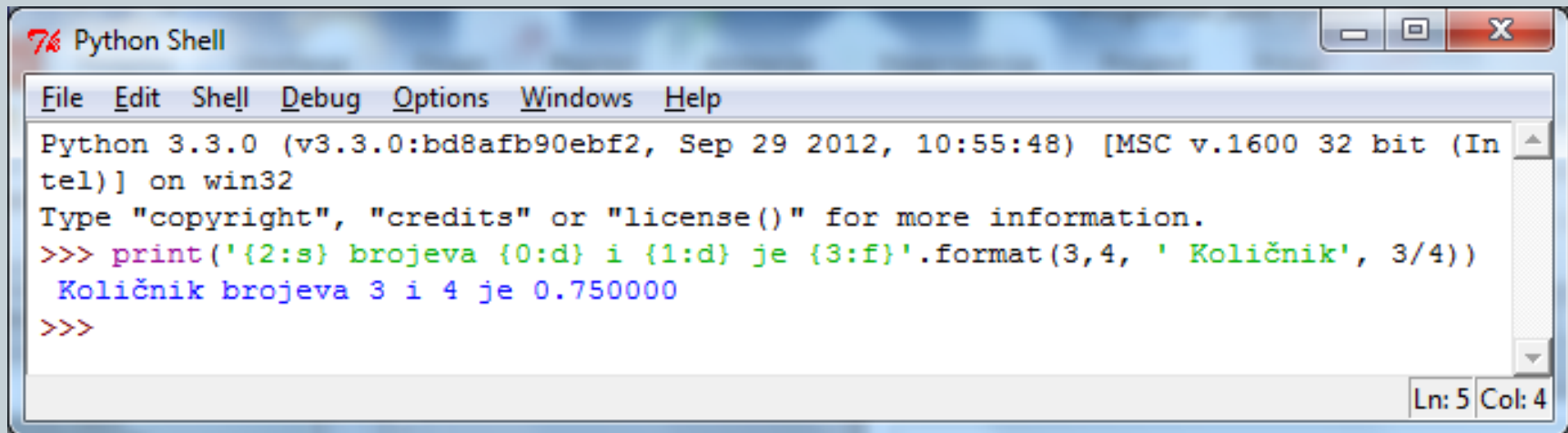


```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('{0:7.2f}'.format(7/3))
      2.33
>>>
```

Ln: 5 Col: 4



# Formatirani ispis



The screenshot shows a Python Shell window titled "Python Shell" with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The main text area displays the following content:

```
Python 3.3.0 (v3.3.0:bd8afb90ebf2, Sep 29 2012, 10:55:48) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('{2:s} brojeva {0:d} i {1:d} je {3:f}'.format(3,4, ' Količnik', 3/4))
    Količnik brojeva 3 i 4 je 0.750000
>>>
```

The status bar at the bottom right indicates "Ln: 5 Col: 4".

Izgled formatiranog ispisa prilikom unosa podataka:

```
>>> n=int(input('Zbroj brojeva {0} i {1} = {2} '.format(12,25, 12+ 25)))
Zbroj brojeva 12 i 25 = 37
```

# Osnovna pravila pisanja programa



- Pravila za pisanje imena:
  - Imena se sastoje od proizvoljnog broja slova, znamenki i podvlaka, ne smiju početi znamenkom.
  - Python razlikuje velika i mala slova, a i A različita imena

## Ključne riječi:

|        |          |         |          |        |
|--------|----------|---------|----------|--------|
| False  | class    | finally | is       | return |
| None   | continue | for     | lambda   | try    |
| True   | def      | from    | nonlocal | while  |
| and    | del      | global  | not      | with   |
| as     | elif     | if      | or       | yield  |
| assert | else     | import  | pass     |        |
| break  | except   | in      | raise    |        |

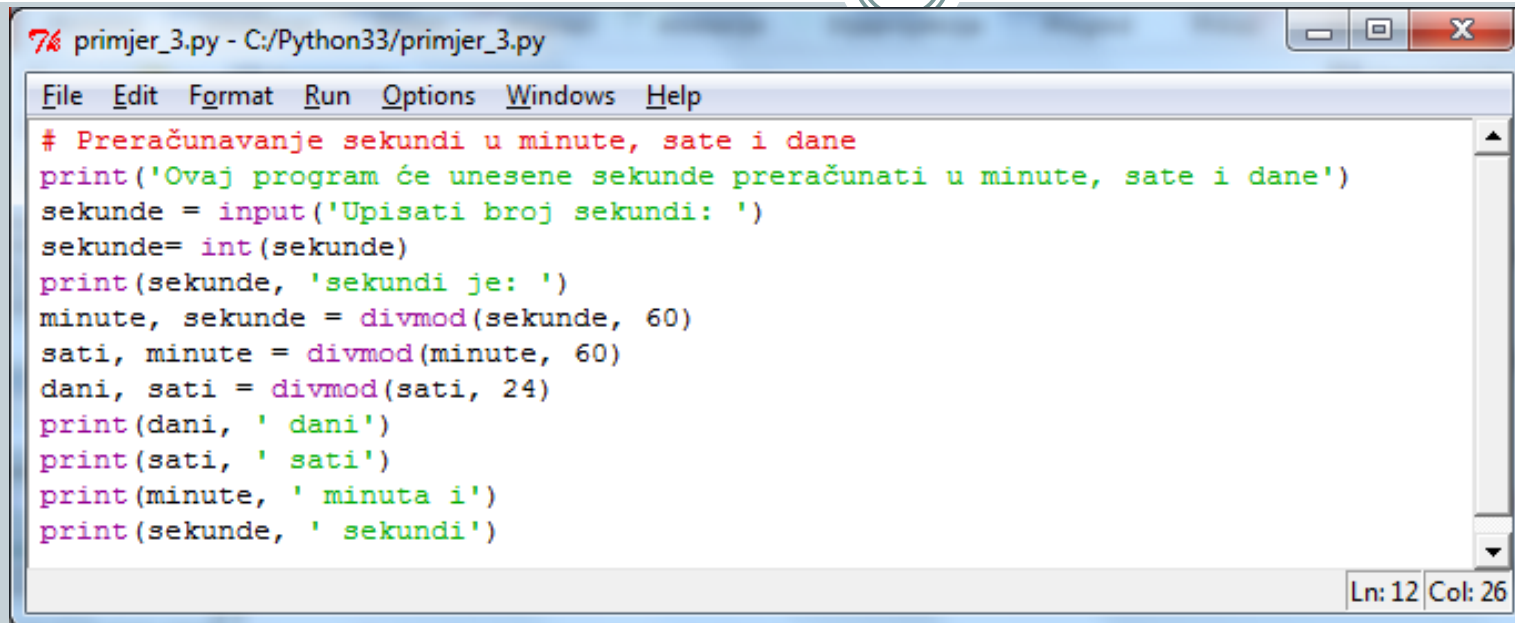
# Naredbe višestrukog pridruživanja



varijabla\_1, varijabla\_2, ... = izraz\_1, izraz\_2, ...

```
>>> a = 100
>>> b = 15
>>> zbroj, razlika = a+b, a-b
>>> zbroj
115
>>> razlika
85
>>> količnik, ostatak = a//b, a%b
>>> količnik
6
>>> ostatak
10
>>> print('zbroj = {} razlika = {}'.format(zbroj, razlika))
zbroj = 115 razlika = 85
```

# Program za preračunavanje sekundi u dane, sate i minute



```
primjer_3.py - C:/Python33/primjer_3.py
File Edit Format Run Options Windows Help
# Preračunavanje sekundi u minute, sate i dane
print('Ovaj program će unesene sekunde preračunati u minute, sate i dane')
sekunde = input('Upisati broj sekundi: ')
sekunde = int(sekunde)
print(sekunde, 'sekundi je: ')
minute, sekunde = divmod(sekunde, 60)
sati, minute = divmod(minute, 60)
dani, sati = divmod(sati, 24)
print(dani, ' dani')
print(sati, ' sati')
print(minute, ' minuta i')
print(sekunde, ' sekundi')
Ln: 12 Col: 26
```

```
Ovaj program će unesene sekunde preračunati u minute, sate i dane
Upisati broj sekundi: 1000000
1000000 sekundi je:
11 dani
13 sati
46 minuta i
40 sekundi
```

# Jednostavniji zapis



```
sekunde=input('Upisati broj sekundi: ')
```

```
sekunde= int(sekunde)
```

## **Jednostavnije pisanje:**

```
sekunde=int(input('Upisati broj sekundi: '))
```

**File -> Recent Files** , aktivira pomoćni padajući izbornik s nizom programa, odnosno modula koje smo u posljednje vrijeme upotrebljavali

# Zadaci za vježbu



1. Napiši program koji će unositi duljine stranica pravokutnika i ispisivati:

a) opseg pravokutnika

*Primjer:*

| Unos | Ispis |
|------|-------|
| 2    | 14    |
| 5    |       |

b) površina pravokutnika

*Primjer:*

| Unos | Ispis |
|------|-------|
| 2    | 10    |
| 5    |       |

# Zadaci za vježbu



2. Napiši program koji će unositi prirodan broj  $n$  i ispisivati površinu kvadrata čija stranica ima duljinu  $n$ .

*Primjer:*

| Unos | Ispis |
|------|-------|
| 6    | 36    |

3. Napiši program koji će unositi ocjene iz pet predmeta, a ispisivati prosjek ocjena.

| Unos | Ispis |
|------|-------|
| 3    | 4.2   |
| 4    |       |
| 5    |       |
| 4    |       |
| 5    |       |

# Zadaci za vježbu



4. Ivana već dugo želi kupiti novu pernicu koja stoji  $n$  kuna. Za rođendan je od bake dobila  $m$  kuna ( $m < n$ ). Nakon što je djedu ispričala priču s pernicom, djed joj je obećao da će joj dati ostatak novaca. Kako još ne zna najbolje računati, moli te za pomoć. Napiši program koji će unositi cijenu pernice ( $n$ ) i iznos koji je dobila od bake ( $m$ ), a ispisivati iznos koji joj treba dati djed kako bi mogla kupiti pernicu.

*Primjer:*

| Unos | Ispis |
|------|-------|
| 132  | 82    |
| 50   |       |



# Zadaci za vježbu



5. Napiši program koji će unositi dva prirodna broja  $a$  i  $b$  te ispisivati njihov zbroj, razliku, umnožak i količnik (drugi je broj uvijek različit od nule). Ispis programa treba biti „punog” oblika, primjerice, za unos brojeva 5 i 7 te operaciju  $+$  ispis treba biti  $5 + 7 = 12$ .

*Primjer:*

| Unos | Ispis     |
|------|-----------|
| 4    | $4+2=6$   |
| 2    | $4-2=2$   |
|      | $4*2=8$   |
|      | $4/2=2.0$ |

# Zadaci za vježbu



6. Napiši program koji će unositi iznos odobrenog potrošačkog kredita  $c$ , godišnju kamatnu stopu  $p$ , broj mjeseci  $m$ , a ispisivati kamate prema formuli:

$$k = \frac{cp(m+1)}{2400}$$

*Primjer:*

| Unos | Ispis |
|------|-------|
| 1000 | 43.33 |
| 8    |       |
| 12   |       |

# Zadaci za vježbu



7. Ivica ima  $n$  kuna za koje želi kupiti prijateljicama čokolade. Jedna čokolada stoji  $m$  kuna. Ivicu zanima koliko će najviše čokolada moći kupiti te koliko će mu novca nakon toga još preostati. Pomogni Ivici i napiši program koji će unositi iznos novca kojim Ivica raspolaže te cijenu jedne čokolade, a ispisivati koliko maksimalno čokolada Ivica može kupiti te koliko će mu novca nakon toga preostati.

*Primjer:*

| Unos | Ispis |
|------|-------|
| 100  | 5     |
| 17   | 15    |

# Moduli – zbirke funkcija



- *import* – *f* funkcije iz modula koje želimo upotrijebiti moramo najprije najaviti
- u interaktivnom prozoru

```
>>> x=16
>>> import math
>>> y=math.sqrt(x)
>>> y
4.0
```

```
>>> from math import sqrt
>>> x=25
>>> y=sqrt(x)
>>> y
5.0
>>> print(sqrt(81))
9.0
```

# Moduli – zbirke funkcija



- *Uvoz funkcija može se obaviti tako da se umjesto imena funkcije stavi znak \*. Znak \* simbolizira sve funkcije koje se nalaze u modulu.*

```
>>> from math import *
```

```
>>> print(sqrt(2))
```

```
1.4142135623730951
```

```
>>> x=3
```

```
>>> y=4
```

```
>>> print(sqrt(x**2 + y**2))
```

```
5.0
```

# Moduli – zbirke funkcija



```
>>> r=1
>>> from math import pi
>>> print('Površina kruga s polumjerom {0} iznosi {1}'.format(r,pi*r**2))
Površina kruga s polumjerom 1 iznosi 3.141592653589793
```

Preporuka za uvoz modula koristiti se naredbom:

- `import ime_modula` ili
- `from ime_modula import ime_funkcije_1, ime_funkcije_2,...`
- Najčešće korištene funkcije modula `math` su: `ceil(x)`, `floor(x)`, `exp(x)`, `log(x,b)`, `sin(x)`, `cos(x)`, `tan(x)`, `asin(x)`, `acos(x)`.

# Moduli – zbirke funkcija



| Funkcija          | Opis  | Primjer  |
|-------------------|---|--|
| $\text{ceil}(x)$  | Najmanji cijeli broj veći od ili jednak $x$ | $\text{ceil}(3.4)=4$<br>$\text{ceil}(4.9)=5$     |
| $\text{floor}(x)$ | najveći cijeli broj manji ili jednak $x$    | $\text{floor}(3.2) = 3$<br>$\text{floor}(7.8)=7$ |
| $\text{exp}(x)$   | $e^x$                                       | $\text{exp}(1)=2.71828$                          |
| $\text{log}(x,b)$ | $\log_b x$                                  | $\text{log}(100,10)=2.0$                         |
| $\text{sin}(x)$   | $\sin x$                                    | $\text{sin}(\pi)= 0$                             |
| $\text{cos}(x)$   | $\cos x$                                    | $\text{cos}(\pi)= -1$                            |
| $\text{tan}(x)$   | $\text{tg } x$                              | $\text{tan}(0)=0.0$                              |
| $\text{asin}(x)$  | Arcus sinus                                 | $\text{asin}(-1)= 1.57079$                       |
| $\text{acos}(x)$  | Arcus kosinus                               | $\text{acos}(-1)=3.14159$                        |

# Moduli – zbirke funkcija



- Unutar modula **math** ne nalaze se funkcije:  
**abs(x)**, **round(x)**, one su ugrađene funkcije u Pythonu
- Funkcija **round(x)** vraća cijeli broj koji je najbliži realnom broju  $x$
- $\text{round}(7.2)=7$ ,  $\text{round}(9.8)=10$ ,  $\text{round}(-3.2)= -3$
- Modul **random** sadržava funkcije kojima se može generirati nasumične brojeve i modelirati slučajne događaje i pojave.
- U modulu **random** postoji funkcija **randint(a,b)**



# Moduli – zbirke funkcija



```
>>> from random import randint
```

```
>>> print(randint(0,1))
```

```
1
```

```
>>> print(randint(5,15))
```

```
12
```

```
>>> print(randint(25,150))
```

```
114
```

# Uvjetne naredbe



if, if – else, if – elif – else

Opći oblik naredbe grananja :

```
if uvjet_1:
```

```
    naredba_1
```

```
elif uvjet_2:
```

```
    naredba_2
```

```
...
```

```
elif uvjet_n:
```

```
    naredba_n
```

```
else:
```

```
    naredba
```

# Uvjetne naredbe



## *if...else*

```
x,y=17,25
```

```
print('Koliki je zbroj brojeva {0} i {1}?' .format(x,y))
```

```
z=int(input('Utiskati odgovor: '))
```

```
if z == x+y:
```

```
    print('Točno!')
```

```
else:
```

```
    print('Netočno!')
```

# Uvjetne naredbe



## *if...else*

```
x,y=17,25
```

```
print('Koliki je zbroj brojeva {0} i {1}?' .format(x,y))
```

```
z=int(input('Utipkati odgovor: '))
```

```
if z == x+y:
```

```
    print('Točno!')
```

```
else:
```

```
    print('Netočno!')
```

# Uvjetne naredbe



```
import random
a,b=1,9
x,y=random.randint(a,b),random.randint(a,b)
operator = random.randint(0,1)
if operator:
    oper='umnožak'
    t=x*y
else:
    oper='zbroj'
    t=x+y
print('Koliki je {0} brojeva {1} i {2}?.format(oper,x,y))
z=int(input('Utipkati odgovor: '))
if z == t:
    print('Točno!')

else:
    print('Netočno!')
```

# Uvjetne naredbe



*Primjer 1.*

***Napiši program koji će od zadana tri broja ispisati najveći od njih.***

```
#program if_primjer_1  
from math import*  
a=int(input("a = "))  
b=int (input("b = "))  
c=int (input("c = "))  
max = a  
if (b>max):  
    max=b  
if(c>max):  
    max=c  
print('Max= {0:d}'.format(max))
```

# Relacijski operatori



## *Relacijski operatori*

| Operator | Značenje simbola     |
|----------|----------------------|
| >        | veće od              |
| <        | manje od             |
| >=       | veće od ili jednako  |
| <=       | manje od ili jednako |
| ==       | jednako              |
| !=       | nije jednako         |

# Logički operatori



## *Logički operatori*

| Operator | Naziv operacije                   | Algebarski simbol |
|----------|-----------------------------------|-------------------|
| and      | I operacija, konjunkcija          | $\wedge$          |
| or       | ILI operacija, disjunkcija        | $\vee$            |
| not      | NE operacija,<br>komplementiranje | $\neg$            |

## *Redosljed izvođenja logičkih operacija*

| Redosljed | Operacija |
|-----------|-----------|
| 1.        | not       |
| 2.        | and       |
| 3.        | or        |



# Redosljed izvođenja operacija



*Redosljed izvođenja operacija*

| Redosljed | Operacija   |
|-----------|-------------|
| 1.        | aritmetički |
| 2.        | relacijski  |
| 3.        | logički     |

```
>>> 2+3<4
```

```
False
```

```
>>> 2*3 -11>3*3 -5
```

```
False
```

```
>>> 3+4>5 and 3+5 >4 and 4 + 5 >3
```

```
True
```

```
>>> False or not False
```

```
True
```

```
>>> False and True
```

```
False
```

```
>>>
```

# Donošenje odluka u programima



*Primjer 2.*

**Zadaj prirodni broj. Ispiši je li taj broj djeljiv sa sedam.**

```
# djeljivost broja sa 7
print('Program će ispisati je li neki utipkani broj djeljiv sa 7')
broj=int(input('Utipkati prirodni broj: '))
if broj % 7 ==0:
    print('Broj {} je djeljiv sa sedam.'.format(broj))
else:
    print('Broj {} nije djeljiv sa sedam.'.format(broj))
```

Za uneseni broj želimo ustanoviti je li manji od nule, jednak nuli ili je veći od nje.



# Izbor jedne od tri mogućnosti s **if-else** strukturom

```
broj=int(input('Utipkati željeni broj:'))
```

```
if broj== 0:
```

```
    print('Broj je jednak nuli.')
```

```
else:
```

```
    if broj>0:
```

```
        print('Broj je veći od nule.')
```

```
    else:
```

```
        print('Broj je manji od nule.')
```

Za uneseni broj želimo ustanoviti je li manji od nule, jednak nuli ili je veći od nje, upotreba **elif**



```
# Izbor jedne od tri mogućnosti s if-elif-else strukturom  
broj=int(input('Utipkati željeni broj:'))  
if broj== 0:  
    print('Broj je jednak nuli.')  
elif broj>0:  
    print('Broj je veći od nule.')  
else:  
    print('Broj je manji od nule.')
```

# Kiselost i lužnatost otopina izražavaju se pH-vrijednostima.



*Napiši program koji će za utipkanu pH – vrijednost napisati kakva je otopina.*

| <b>pH</b> | <b>Opis</b>   |
|-----------|---------------|
| <0,4.5]   | jako kisela   |
| <4.5,6.5] | slabo kisela  |
| <6.5,7.5] | neutralna     |
| <7.5,9.5] | slabo lužnata |
| <9.5,14]  | jako lužnata  |

# Kiselost i lužnatost otopina izražavaju se pH-vrijednostima.

*# Klasifikacija otopina po pH- vrijednostima*

```
print('Program će ovisno o utipkanoj vrijednosti napisati je li otopina jako  
kisel, slabo kisel, neutralna, slabo lužnata ili jako lužnata')
```

```
ph=float(input('Upisati pH-vrijednost: '))
```

```
if ph > 0 and ph <= 4.5:
```

```
    print('Otopina s pH-vrijednošću {0:0.1f} je jako kisel.'.format(ph))
```

```
elif ph > 4.5 and ph <= 6.5:
```

```
    print('Otopina s pH-vrijednošću {0:0.1f} je slabo kisel.'.format(ph))
```

```
elif ph > 6.5 and ph <= 7.5:
```

```
    print('Otopina s pH-vrijednošću {0:0.1f} je neutralna.'.format(ph))
```

```
elif ph > 7.5 and ph <= 9.5:
```

```
    print('Otopina s pH-vrijednošću {0:0.1f} je slabo lužnata.'.format(ph))
```

```
elif ph > 9.5 and ph <= 14.0:
```

```
    print('Otopina s pH-vrijednošću {0:0.1f} je jako lužnata.'.format(ph))
```

```
else:
```

```
    print('Zadana vrijednost pH = {0:0.1f} nije uobičajena.'.format(ph))
```

# Ponavljanje bčpkova naredbi



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

```
    blok_naredbi
```

```
    ...
```

ponavlja se od 0 do n-1

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

```
    print(i)
```

**0**

**1**

**2**

**3**

**4**

# Ponavljanje bčpkova naredbi



```
>>> for i in range(5):  
    print(i, end='')
```

0 1 2 3 4

```
>>> for i in range(100):  
    print(i, end=' ')  
    if not((i+1) % 10):  
        print()
```

```
0 1 2 3 4 5 6 7 8 9  
10 11 12 13 14 15 16 17 18 19  
20 21 22 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37 38 39  
40 41 42 43 44 45 46 47 48 49  
50 51 52 53 54 55 56 57 58 59  
60 61 62 63 64 65 66 67 68 69  
70 71 72 73 74 75 76 77 78 79  
80 81 82 83 84 85 86 87 88 89  
90 91 92 93 94 95 96 97 98 99
```



# Ponavljanje bčpkova naredbi



```
>>> for i in range(100):  
    print('{:3d}'.format(i),end=' ')  
    if not((i+1) % 10):  
        print()
```

```
0  1  2  3  4  5  6  7  8  9  
10 11 12 13 14 15 16 17 18 19  
20 21 22 23 24 25 26 27 28 29  
30 31 32 33 34 35 36 37 38 39  
40 41 42 43 44 45 46 47 48 49  
50 51 52 53 54 55 56 57 58 59  
60 61 62 63 64 65 66 67 68 69  
70 71 72 73 74 75 76 77 78 79  
80 81 82 83 84 85 86 87 88 89  
90 91 92 93 94 95 96 97 98 99
```

**Napiši program koji će unositi prirodan broj n i ispisivati sve njegove djelitelje.**



```
# Ispis svih djelitelja broja n  
n= int(input('Unesi prirodni broj: '))  
for i in range(1,n+1):  
    if n % i == 0:  
        print(i,end=' ')
```

**Unesi prirodni broj: 15  
1 3 5 15**

Funkcija range(). Može se pozvati i s **tri parametra:**

```
>>>for i in range(1,10,3)  
    print(i)  
  
1  
4  
7  
>>>
```

# Range



```
>>>for i in range(1,10,3)
    print(i)
```

```
1
4
7
```

```
>>>for i in range(10,4,-2):
    print (i)
```

```
10
8
6
```

# Range



```
>>>for i in range(10,12,-1)  
    print(i)
```

prvi parametar funkcije  
range() je manji od drugog,  
pa nema nikakvog ispisa

## Napiši program koji će unositi prirodni broj $n$ i provjeravati je li prost broj

```
# Provjerava je li učitani broj prost
n = int(input('Unesi broj: '))
brojac= 0
for i in range(1,n+1):
    if n%i == 0:
        brojac+=1
if brojac ==2 and n>1:
    print('Broj {} je prost'.format(n))
else:
    print('Broj {} nije prost'.format(n))
```

Unesi broj: 17  
Broj 17 je prost

Unesi broj: 15  
Broj 15 nije prost

Prosti brojevi: 2,3,5,7,11,13,... Ako broj nije prost, za njega ćemo reći da je složen. Broj 1 nije niti prost niti složen.

# Uvjetno ponavljanje bloka naredbi



...

```
while uvjet:  
    blok_naredbi
```

...

```
while i<10:  
    print(i,end=' ')  
    i+=1
```

0 1 2 3 4 5 6 7 8 9

## Uvjetno ponavljanje bloka naredbi



```
# Srednja visina uz korištenje while petlje  
print('Izračunavanje srednje visine proizvoljnog broja osoba')  
n=0  
s=0  
ima_brojeva='da'  
while ima_brojeva == 'da':  
    v=float(input('Upisati visinu u m: '))  
    s +=v  
    n +=1  
    ima_brojeva=input('Ima li još unosa (utipkati da ili ne)? ')  
print('\n Srednja visina {} osoba u metrima je {:.4f}'.format(n,s/n))
```

## Uvjetno ponavljanje bloka naredbi

Izračunavanje srednje visine proizvoljnog broja osoba

Upisati visinu u m: 1.78

Ima li još unosa (utipkati da ili ne)? da

Upisati visinu u m: 1.56

Ima li još unosa (utipkati da ili ne)? da

Upisati visinu u m: 1.83

Ima li još unosa (utipkati da ili ne)? da

Upisati visinu u m: 1.72

Ima li još unosa (utipkati da ili ne)? da

Upisati visinu u m: 1.65

Ima li još unosa (utipkati da ili ne)? da

Upisati visinu u m: 1.69

Ima li još unosa (utipkati da ili ne)? ne

**Srednja visina 6 osoba u metrima je 1.7050**



## Uvjetno ponavljanje bloka naredbi



```
# Srednja visina uz korištenje while petlje
print('Izračunavanje srednje visine proizvoljnog broja osoba')
n=0
s=0
v=float(input('Upisati visinu u metrima (negativni broj za kraj: ')
while v > 0:
    s +=v
    n +=1
    v=float(input('Upisati visinu u metrima (negativni broj za kraj): '))

print("\n Srednja visina { } osoba u metrima je {:.4f}'.format(n,s/n))
```

## Uvjetno ponavljanje bloka naredbi



Izračunavanje srednje visine proizvoljnog broja osoba

Upisati visinu u metrima (negativni broj za kraj): 1.78

Upisati visinu u metrima (negativni broj za kraj): 1.56

Upisati visinu u metrima (negativni broj za kraj): 1.83

Upisati visinu u metrima (negativni broj za kraj): 1.72

Upisati visinu u metrima (negativni broj za kraj): 1.65

Upisati visinu u metrima (negativni broj za kraj): 1.69

Upisati visinu u metrima (negativni broj za kraj): -1

Srednja visina 6 osoba u metrima je 1.7050

# Ugrađene funkcije



Upoznali smo funkcije: *int()*, *float()*, *bin()*, *hex()*, *ord()*, *input()*, *print()*, *abs()*, *round()*

**Ugrađene funkcije:**

| Funkcija                  | Primjer               |
|---------------------------|-----------------------|
| $\text{min}(a,b,c,\dots)$ | $\text{min}(2,1,5)=1$ |
| $\text{max}(a,b,c,\dots)$ | $\text{max}(2,1,5)=5$ |
| $\text{pow}(a,b)$         | $\text{pow}(2,5)=32$  |
| $\text{pow}(a,b,n)$       | $\text{pow}(2,5,3)=2$ |

***(kolicnik,ostatak) = divmod(a,b)***

# Ugrađene funkcije



**import math**

**help(math)** – ispisati popis svih funkcija i ostalih podataka o modulu, slijedi dio ispisa:

FUNCTIONS

`acos(...), acos(x)`

Return the arc cosine (measured in radians) of  $x$ .

`acosh(...), acosh(x)`

Return the hyperbolic arc cosine (measured in radians) of  $x$ .

`asin(...), asin(x)`

Return the arc sine (measured in radians) of  $x$ .

`asinh(...), asinh(x)`

Return the hyperbolic arc sine (measured in radians) of  $x$ .

# Definiranje vlastitih funkcija

```
def ime_funkcije(popis_parametara):  
    blok_naredbi  
    return vrijednost
```

**Primjer 1:** Napiši funkciju koja za zadani prirodni broj provjerava je li broj paran te ispisuje odgovarajuću poruku.

```
#program 1
```

```
def paran(n):  
    if n%2==0:  
        print('Broj {0} je paran'.format(n))  
    else:  
        print('Broj {0} je neparan'.format(n))  
    return  
n=int(input("unesi broj n:"))  
paran(n)
```

# Definiranje vlastitih funkcija



Definiraj više brojeva i za njih provjeri jesu li parni:

```
#program 2  
def paran(n):  
    if n%2==0:  
        print('Broj {0} je paran'.format(n))  
    else:  
        print('Broj {0} je neparan'.format(n))  
    return  
  
n=int(input("Koliko brojeva:"))  
for i in range(n):  
    a=int(input("Zadaj broj:"))  
    paran(a)
```

```
Koliko brojeva:5  
Zadaj broj:3  
Broj 3 je neparan  
Zadaj broj:6  
Broj 6 je paran  
Zadaj broj:2  
Broj 2 je paran  
Zadaj broj:7  
Broj 7 je neparan  
Zadaj broj:99  
Broj 99 je neparan
```

# Definiranje vlastitih funkcija



```
def armstrongov(n):  
#funkcija True ako je n Armstrongov broj, a inače vraća False  
    s=0  
    t=n  
    while n>0:  
        s+=(n%10)**3  
        n//=10  
    if s==t:  
        return True  
    else:  
        return False  
for i in range (1,501):  
    if armstrongov(i):  
        print(i)
```

|            |
|------------|
| <b>1</b>   |
| <b>153</b> |
| <b>370</b> |
| <b>371</b> |
| <b>407</b> |

# Definiranje vlastitih funkcija



```
def armstrongov(n):  
    #funkcija vraća True ako je n Armstrongov broj, a inače vraća  
    False  
    s=0  
    t=n  
    while n>0:  
        s+=(n%10)**3  
        n//=10  
    if s==t:  
        return True  
    else:  
        return False  
def main():  
    for i in range (1,501):  
  
        if armstrongov(i):  
            print(i)  
    return
```

1  
153  
370  
371  
407



# Definiranje vlastitih funkcija

```
def armstrongov(n):  
    #funkcija True ako je n Armstrongov broj, a inače vraća False  
    s=0  
    t=n  
    while n>0:  
        s+=(n%10)**3  
        n//=10  
    if s==t:  
        return True  
    else:  
        return False  
def main():  
    for i in range (1,501):  
        if armstrongov(i):  
            print(i)  
    return  
main()
```

1  
153  
370  
371  
407

## Definiranje vlastitih funkcija

```
def armstrongov(n):  
#funkcija True ako je n Armstrongov broj  
    s=0  
    t=n  
    while n>0:  
        s+=(n%10)**3  
        n//=10  
    if s==t:  
        return True  
    else:  
        return False  
def prost(n):  
    for i in range(2,round(n**(1/2)+1)):  
        if n%i==0:  
            return False  
    if n>1:  
        return True  
    else:  
        return False
```

```
def main():  
    n=int(input('Unesi prirodan broj:  
'))  
    b=0  
  
    for i in range (1,n+1):  
        if armstrongov(i)and prost(i):  
            b+=1  
            print(i)  
    if b==0:  
        print('Nema')  
  
    return  
main()
```

## Razni zadaci s funkcijama:



1. Napiši program koji će izračunati tangens kuta zadanog u stupnjevimama, minutama i sekundama, uz pomoć funkcije.
2. Napiši program koji će za zadana dva kompleksna broja ispisati njihov zbroj i umnožak.
3. Napiši program koji će za zadane stranice i dijagonalu pravokutnika izračunati drugu stranicu, opseg i površinu.
4. Napišite program koji će izračunati i ispisati imaginarni dio umnoška dva kompleksna broja.
5. Napišite program koji će ispisati zbroj i umnožak znamenki učitanoog prirodnog broja.
6. Napišite program koji će od tri zadana realna broja ispisati njihovu aritmetičku i harmonijsku sredinu.

## Razni zadaci s funkcijama:



7. Napiši program koji će ispisati opseg i površinu kvadrata ako je poznata duljina njegove dijagonale.
8. Napiši program koji će uz pomoć funkcije ispisati najmanji od triju zadanih cijelih brojeva.
9. Napiši program koji će na odsječku  $[200, 400]$  ispisati zbroj brojeva, broj brojeva djeljivih sa sedam i koliko ih je neparnih.
10. Napišite program koji će ispisati zbroj brojeva djeljivih sa 13 na odsječku  $[200, 400]$ .
11. Napišite program uz pomoć funkcije koji će za učitani realni broj ispisati njegov cijeli i decimalni dio.
12. Napišite funkciju čiji će ulazni parametri biti dva prirodna broja. Funkcija treba vraćati zbroj tih dvaju brojeva.

## Rješenja nekih zadataka:



Napišite funkciju čiji će ulazni parametri biti dva prirodna broja. Funkcija treba vraćati zbroj tih dvaju brojeva.

### **#program 12**

```
def zbroj(a,b):
    x=a+b
    return x
def main():
    a=int(input("Prvi broj:"))
    b=int(input("Drugi broj:"))
    print('Zbroj brojeva {} i {} iznosi {}'.format(a,b,zbroj(a,b)))

main()
```

# Rješenje prvog zadatka



Napiši program koji će izračunati tangens kuta zadanog u stupnjevima, minutama i sekundama, uz funkcije.

```
#program 1a
from math import *
def tg(stu,min,sek):
    kut=stu+min/60+sek/3600
    rad=kut*pi/180
    x=sin(rad)/cos(rad)
    return x
def main():
    stu=int(input("Zadaj stupnjeve:"))
    min=int(input("Zadaj minute:"))
    sek=int(input("Zadaj sekunde:"))
    print('Za zadani kut {0} stu {1} min i {2} sek tangens iznosi: {3:.2f}'.format(stu,min,sek,tg(stu,min,sek)))
main()
```

## Rješenje trećeg zadatka

```
#program 3
from math import *
def b(a,d):
    x=sqrt(pow(d,2)-pow(a,2))
    return x
def o(a,d):
    y=2*(a+b(a,d))
    return y
def p(a,d):
    z=a*b(a,d)
    return z
def main():
    a=int(input("Zadaj stanicu a=:"))
    d=int(input("Zadaj dijagonalu d=:"))
    print('Za zadanu stanicu a = {0} i dijagonalu d = {1}, stranica b = {2:
.2f}, opseg = {3: .2f}, površina = {4: .2f}
'.format(a,d,b(a,d),o(a,d),p(a,d)))
    return
main()
```