

Sveučilište u Splitu  
Prirodoslovno-matematički fakultet

# Programiranje 1 2013./2014. Vježba 1.



Programiranje 1 2013./2014.  
Vježba 1

## Sadržaj

- Prijava na sustav Moodle
- Instalacija Pythona
- Rad u Pythonu
- Osnovni tipovi podataka
- Aritmetički izrazi



# Moodle

- Prijava na sustav Moodle

[moodle.pmfst.hr](http://moodle.pmfst.hr)

- Korisničko ime i lozinka – isti kao za e-mail (vidjeti papirić u indeksu)

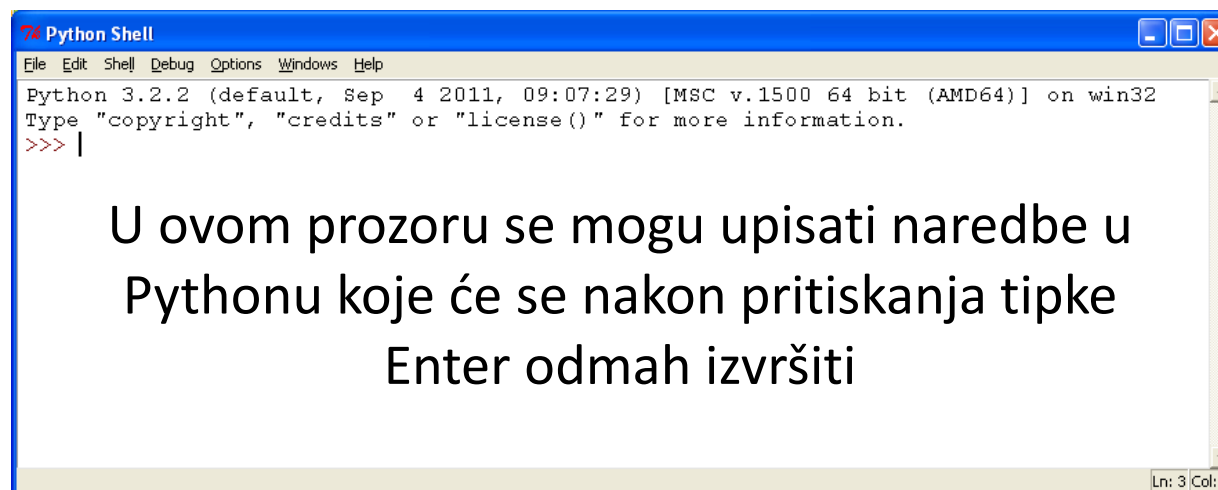


# Python 3.3.2 – Instalacija i pokretanje

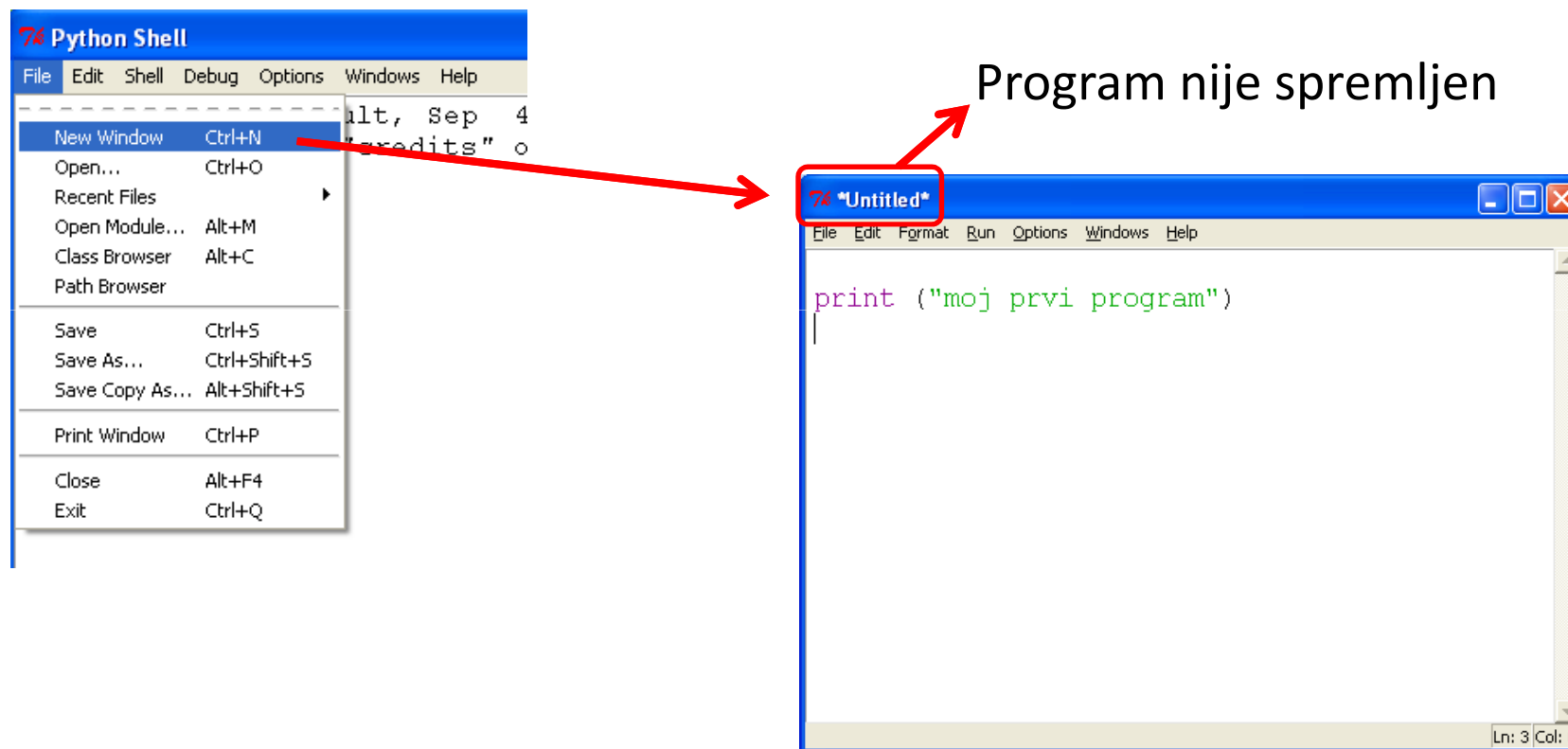
1) Link za download:

<http://www.python.org/download>

2) Pokrenuti: IDLE (Python GUI)

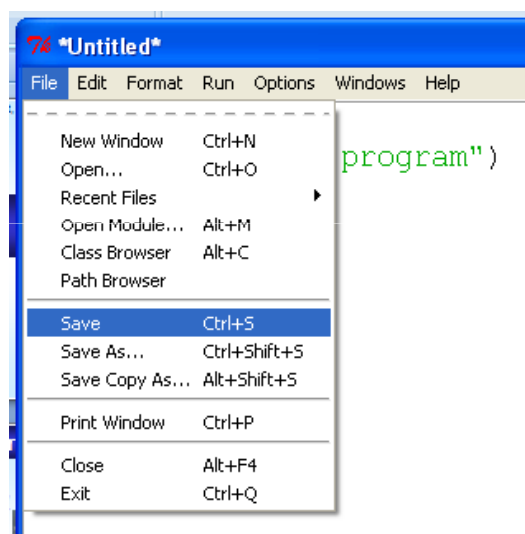


# Python 3.3.2 – Otvaranje novog programa

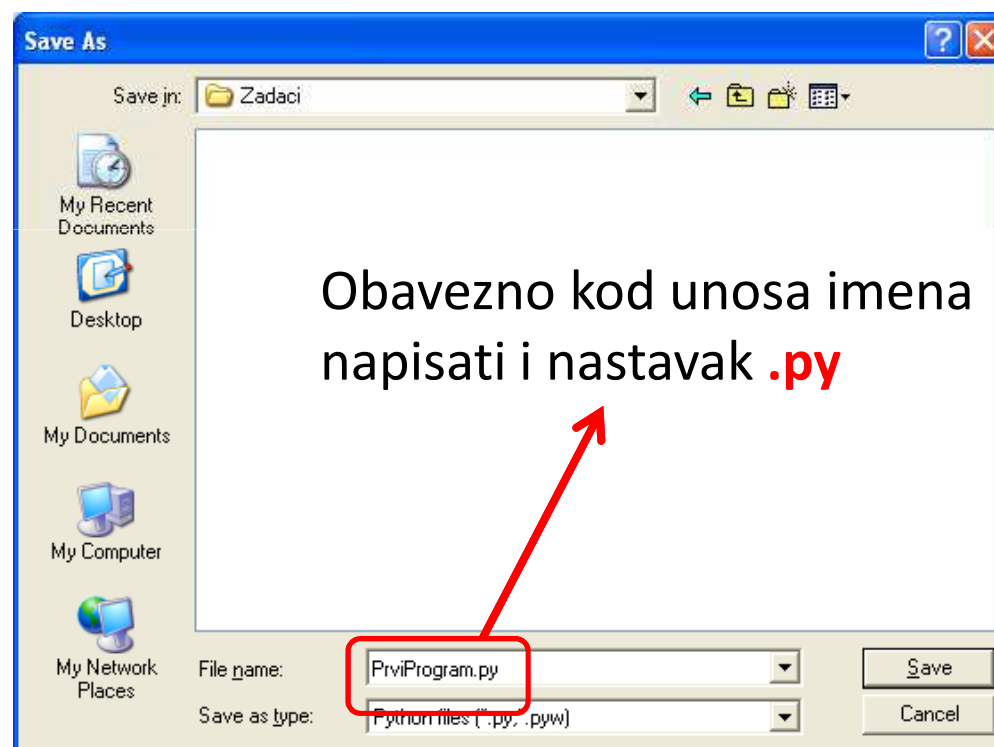


# Python 3.3.2 – Spremanje programa

1) Save



2) Save As



# Python Shell

- Omogućuje **interaktivni** rad između računala i korisnika:
  - Na početku novog reda pojavljuju tri strelice **>>>**.
  - **Odzivni znak (eng. prompt )** - poruka kojom se od korisnika traži neka akcija ili odgovor.
  - Korisnik odgovara utipkavanjem nekog niza znakova koji treba biti zaključen tipkom Enter.
  - Python obavlja akciju i u novom redu ispiše odzivni znak.
- Kopiranje cijelog retka
  - Kliknite na kraj retka koji želite kopirati i pritisnite Enter
- Python razlikuje veliko ili malo slovo (eng. **case sensitive**).



## Python Shell

- Kada se nakon prompta utipka neka vrijednost Python će tu istu vrijednost ispisati:

```
>>> 120
```

```
120
```

- Zbog lakšeg prepoznavanja na zaslonu monitora Python piše svoj ispis plavom bojom.





# Osnovni tipovi podataka

- **int** (cijeli broj)
  - **float** (broj s pomičnom točkom)
  - **bool** (logički ili Booleov tip)
  - **str** (znakovni niz, string)
- 
- Postoji jedna posebna vrijednost koja se zove **None** i označava ništa.



## Cijeli brojevi

- Naziv za cjelobrojni tip podataka u Pythonu je int.
- Broj znamenaka cijelog broja u Pythonu nije ograničen – cijeli broj može bit proizvoljne veličine.
- Cijeli broj u binarnom obliku: ispred binarnog prikaza napiše se 0b.
- Cijeli broj u heksadekadskom obliku: ispred binarnog prikaza napiše se 0x.



## Cijeli brojevi

➤ Primjeri cijelih brojeva:

```
>>> 27
```

```
27
```

```
>>> - 302
```

```
-302
```

```
>>> 1234567890123456781234567812345678
```

```
1234567890123456781234567812345678
```



## Cijeli brojevi

- Katkada (iako rijetko) će biti prikladno cijeli broj utipkati u binarnom obliku. To se čini tako da se ispred binarnog prikaza napiše 0b:

```
>>> 0b00001111
```

```
15
```

```
>>> 0b11111111
```

```
255
```

```
>>> 0b0001010
```

```
10
```

- Kao što vidimo, Python će binarni zapis pretvoriti dekadni i ispisati cijeli broj u dekadnom obliku.



## Cijeli brojevi

- Cijele brojeve koje smo zapisali binarno možemo zapisati i u heksadekadskom sustavu koristeći **0x**:

```
>>> 0x0F
```

```
15
```

```
>>> 0xF0
```

```
240
```

```
>>> 0xFF
```

```
255
```

```
>>> 0x0A
```

```
10
```



## Realni brojevi

- Naziv za realni tip podataka u Pythonu je **float**.
- Prilikom utipkavanja racionalnog broja mora se upotrijebiti **decimalna točka** prema kojoj Python prepoznaje da se radi o tipu **float**.
- Brojevi s pomičnom točkom dvostruke preciznosti mogu poprimiti vrijednosti o granicama od  **$5.0 \times 10^{-324}$**  pa do  **$1.7 \times 10^{308}$** .
- Broj koji bio imao eksponent veći od +308 je za Python beskonačno veliki broj i Python ga zapisuje kao **inf** (od eng. infinite – beskonačno).



## Realni brojevi

- Racionalni brojevi, kojima aproksimiramo i realne brojeve, u računalima se zapisuju u skladu s dogovorenom međunarodnom normom koja se zove IEEE 754 u formatu s pomičnom točkom (eng. Floating point number), iz čega je nastao naziv tipa podataka – float.
- Primijetimo da je Python brojeve manje od 0.0001 počeo pisati u **eksponencijalnom** obliku.

```
>>> 350
```

```
350
```

```
>>> 35.
```

```
35.0
```

```
>>> 3.5
```

```
3.5
```

```
>>> 0.35
```

```
0.35
```

```
>>> 0.035
```

```
0.035
```

```
>>> .0035
```

```
0.0035
```

```
>>> 0.00035
```

```
0.00035
```

```
>>> 0.000035
```

```
3.5e-05
```

```
>>> 0.0000356789
```

```
3.56789e-05
```



## Realni brojevi

- Pogledajmo sada kako će Python ispisivati velike brojeve:

```
>>> 1e12
```

```
1000000000000.0
```

```
>>> 1e15
```

```
1000000000000000.0
```

```
>>> 1e16
```

```
1e+16
```

```
>>> 1e32
```

```
1e+32
```

- Vidimo da je broj 1e15 ispisan s petnaest znamenki lijevo od decimalne točke, dok je broj 1e16 ispisan u eksponencijalnom obliku.





## Logički ili Booleov tip

- Naziv za logički tip podataka u Pythonu je **bool**.
- Tip bool ima dvije vrijednosti: **True** i **False**.
  - Vrijednosti logičkog tipa mogu se zapisati i kao brojevi 1 (True) i 0 (False).
  - Python će svaki broj različit od 0 smatrati da je True, dok se samo broj 0 smatra False.
  - True i true nije jednako (Python je case sensitive).

# Logički ili Booleov tip

➤ Pogledajmo što će *Python* ispisati kada utipkamo te vrijednosti:

```
>>> True
```

```
True
```

```
>>> False
```

```
False
```

```
>>> true
```

```
Traceback (most recent call last):
```

```
File "<pyshell#4>", line 1, in <module>
```

```
true
```

```
NameError: name 'true' is not defined
```

```
>>> false
```

```
Traceback (most recent call last):
```

```
File "<pyshell#83>", line 1, in <module>
```

```
false
```

```
NameError: name 'false' is not defined
```



## Stringovi

- Osnovni tip podatka za prikaz **teksta** je znakovni niz ili string. Engleski naziv string toliko se udomaćio da je postao sastavni dio hrvatske stručne terminologije. U Pythonu se taj tip zove **str**, a vrijednost znakovnog niza obilježava se **jednostrukim ili dvostrukim navodnim znacima** na početku i kraju niza.
- Preporuka: upotrebljavati jednostruke navodnike (ukoliko je jednostruki navodnik dio teksta koristi se dvostruki navodnik)

```
>>> 'Dobriša Cesarić'
'Dobriša Cesarić'
>>> 'Pjesma "Voćka poslije kiše" je lijepa pjesma.'
'Pjesma "Voćka poslije kiše" je lijepa pjesma.'
>>> "U Pythonu se tekst piše ovako: 'Ovo je tekst.'"
"U Pythonu se tekst piše ovako: 'Ovo je tekst.'"
```



## Stringovi

- Naglasimo da za jednostruki i dvostruki navodnik postoje dvije tipke na tipkovnici. Dvostruki navodnik se ne smije pisati kao dva uzastopno napisana jednostruka navodnika. Evo što bi se u suprotnom ispisalo: `>>> 'Pjesma "Voćka poslije kiše" je lijepa pjesma.'`
- `'Pjesma Voćka poslije kiše je lijepa pjesma.'`
- Python je shvatio da su to dva nezavisna teksta ograđena jednostrukim navodnim znakom napisana jedan iza drugoga i spojio ih je u jedan! Zapazimo da drugi string započinje prazninom i zbog toga postoji razmak između riječi kiše i je.
- Kada pažljivo pogledamo ispis vidimo da je *Python ispisao vrijednost koja je pohranjena kao znakovni niz zajedno s navodnim znacima koji obrubljuju tekst!*



# Stringovi

- U Pythonu ne postoji tip koji se sastoji od jednog znaka – **char**.
- Znak je naprosto vrijednost tipa str duljine jedan.
- Posebni znakovi za oblikovanje teksta:

Posebni znak	Opis djelovanja
\n	prijelaz u novi redak
\t	tabulator
\\	ispisati lijevo ukošenu crtu
\'	ispisati jednostruki navodnik
\"	ispisati dvostruki navodnik

## Aritmetički izrazi

- Operandi mogu biti cijeli brojevi (tip int) i brojevi s pomičnim točkom (tip float).
- Operatori:

Operator	Opis djelovanja
+	zbrajanje
-	oduzimanje ili negacija
*	množenje
**	potenciranje
/	dijeljenje
//	cjelobrojno dijeljenje
%	računanje ostatka dijeljenja

# Aritmetički izrazi

➤ Primjeri jednostavnijih izraza:

```
>>> 2 + 3
```

```
5
```

```
>>> 2 - 3
```

```
-1
```

```
>>> 3 * 4
```

```
12
```

```
>>> 4 / 2
```

```
2.0
```

```
>>> - 7
```

```
-7
```



## Aritmetički izrazi

- Operator **dijeljenja** / djeluje tako da uvijek daje rezultat **float**.
- Pri cjelobrojnom dijeljenju traži se koliko puta se djeliteľ može oduzeti od djeljenika s tim da ostatak bude pozitivan ili jednak nuli:
  - Npr. rezultat cjelobrojnog dijeljenja broja 13 s brojem 4 daje rezultat 3 i ostatak 1
- Pažnja! Ako je barem jedan od operandata tipa float, operator **cjelobrojnog dijeljenja** dat će rezultat tipa float.



## Aritmetički izrazi

- Primjeri cjelobrojnog dijeljenja i modula – ostatka pri dijeljenju:

```
>>> 17 // 3
```

```
5
```

```
>>> 17 % 3
```

```
2
```

```
>>> 18 // 3
```

```
6
```

```
>>> 18 % 3
```

```
0
```

```
>>> 19 // 3
```

```
6
```

```
>>> 19 % 3
```

```
1
```

```
>>> 20 // 3
```

```
6
```

```
>>> 20 % 3
```

```
2
```

```
>>> 21 // 3
```

```
7
```

```
>>> 21 % 3
```

```
0
```



## Aritmetički izrazi

- Operator **cjelobrojnog dijeljenja** `//` dat će rezultat tipa float ako je barem jedan od operatora tipa float:

```
>>> 4. // 2
```

```
2.0
```

```
>>> 4 // 2.
```

```
2.0
```



## Aritmetički izrazi

- Cjelobrojno dijeljenje može biti korisno za različita izračunavanja. Primjerice, cjelobrojno dijeljenje nam omogućava jednostavno preračunavanje trajanja vremenskog razdoblja izraženog u sekundama u minute, sate i dane:

```
>>> 1000000 // 60
```

```
16666
```

```
>>> 1000000 % 60
```

```
40
```

```
>>> 16666 // 60
```

```
277
```

```
>>> 16666 % 60
```

```
46
```

```
>>> 277 // 24
```

```
11
```

```
>>> 277 % 24
```

```
13
```

- Prema tome, 1000000 sekundi je 11 dana, 13 sati, 46 minuta i 40 sekundi. Provjerimo je li rezultat točan:

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

```
1000000
```



## Aritmetički izrazi

- Analizom sljedećih primjera i samostalnim eksperimentiranjem utvrdimo načine izračunavanja izraza:

```
>>> 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
```



## Aritmetički izrazi

- Pri pisanju složenijih izraza dobro je zbog preglednosti upotrijebiti **zgrade**, ako i nisu potrebne.
- Isto tako, napisani izrazi bit će čitljiviji ako se između operandi i operatora umetnu **razmaci**.
- Primjer:

```
>>> 13.+6.3-13.2/4*21-17
```

```
-67.0
```

```
>>> (13. + 6.3) - (13.2 / 4) * 21 - 17
```

```
-67.0
```



## Aritmetički izrazi

- Python u svom interaktivnom načinu rada uvijek pamti zadnju izračunatu vrijednost. Ta se vrijednost može iskoristiti koristiti u sljedećem koraku izračunavanja tako da se napiše znak **\_** (**podvlaka**, engl. *underscore*):

```
>>> 3
```

```
3
```

```
>>> _ + 4
```

```
7
```

```
>>> _ + 5
```

```
12
```

- To može biti praktično, no tako se može zapamtiti samo jedna vrijednost i to samo ona iz zadnjeg koraka računanja!



Programiranje 1 2013./2014.

Vježba 1

## Zadatak 1.1.

Izračunajte vrijednosti sljedećih izraza:

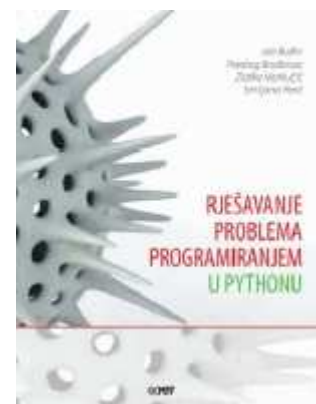
$2+7$ ,  $-14+6$ ,  $10/4$ ,  $8//4$ ,  $11//4$ ,  $9\%5$ ,  $15\%3$ ,  $3**2*2$ ,  $7+3\%5$ ,  
 $15//8\%3+2**3\%4$ .



Programiranje 1 2013./2014.  
Vježba 1

## Zadaci za rješavanje

Budin, L., Brođanac, P., Markučić, Z., Perić, S. (2012)  
Rješavanje problema programiranjem u Pythonu,  
Element, Zagreb



**3.6. Zadaci za ponavljanje i vježbu, strana 34-36**

