

INFORMATIČKI KLUB

FUTURA

LIGA PROGRAMIRANJA



python

#5

**LIGA PROGRAMIRANJA U PYTHONU ZA
OSNOVNE ŠKOLE – 1. RADIONICA**

SVEUČILIŠTE
U DUBROVNIKU
UNIVERSITY
OF DUBROVNIK

Tomo Sjekavica, Informatički klub FUTURA
Dubrovnik, 07. studenog 2023.



Zaštićeno licencom <http://creativecommons.org/licenses/by-nc-sa/3.0/hr/>

Creative Commons



- **slobodno smijete:**

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



- **pod slijedećim uvjetima:**

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <http://creativecommons.org/>.

Raspored Lige programiranja

- 07.11.2023. – **1. radionica**
- 21.11.2023. – **1. kolo Lige programiranja**
- 05.12.2023. – 2. radionica
- 19.12.2023. – 2. kolo Lige programiranja
- termini u 2024. godini na web stranici
- Web stranica Lige programiranja:

<https://www.futura.com.hr/liga-programiranja-u-pythonu-2023-2024/>

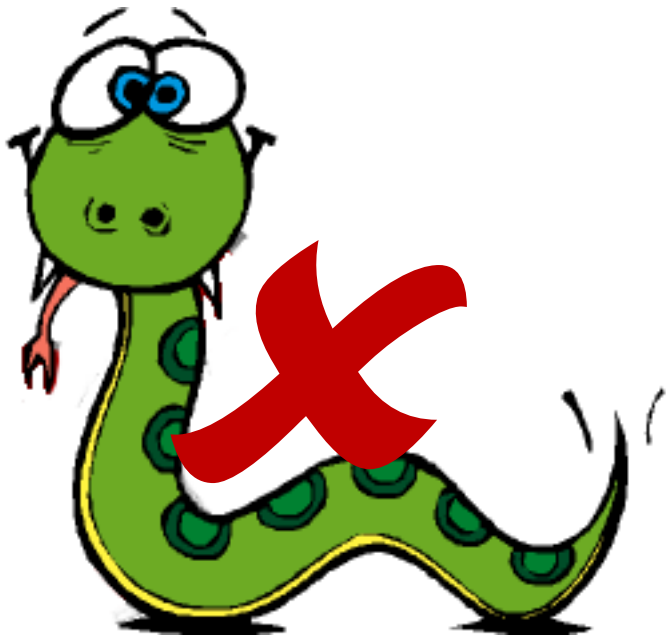
Pravila Lige programiranja

- ❑ Ekipno natjecanje škola **5./6. razredi**
- ❑ Ekipu škole **5./6. razredi** čine 2 ili 3 učenika iz 5. ili 6. razreda
- ❑ Za rezultate ekipa se uzimaju u obzir bodovi 2 najbolja učenika te ekipe za svako kolo lige
- ❑ Pojedinačno natjecanje **5./6. i 7./8.**
- ❑ Nagrade najboljim ekipama i najboljim pojedincima u kategorijama 5./6. i 7./8. razredi

Programski jezik Python

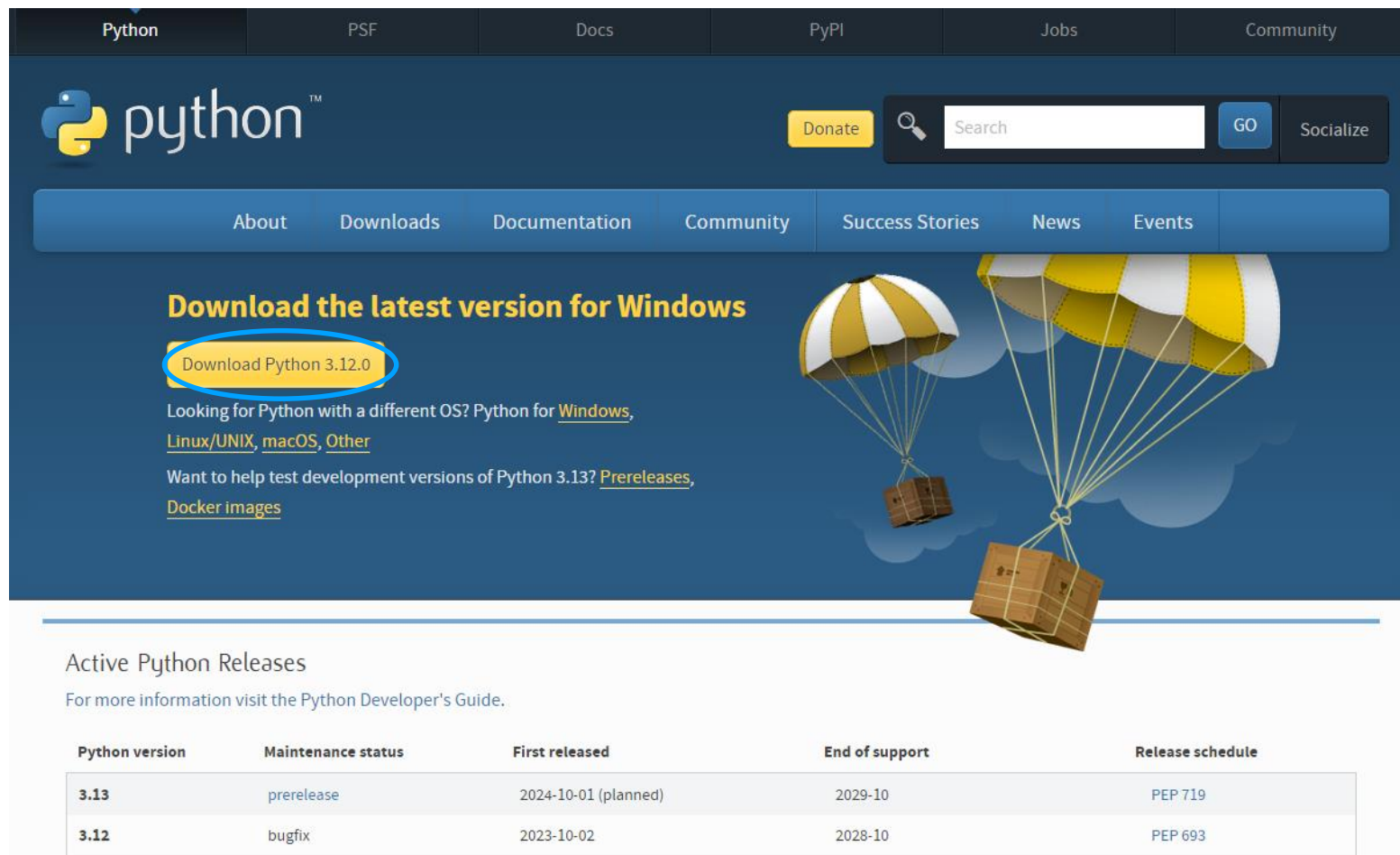


□ <https://www.python.org/>



Preuzimanje Pythona 3.12.0

□ <https://www.python.org/downloads/>



Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Download the latest version for Windows

[Download Python 3.12.0](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)

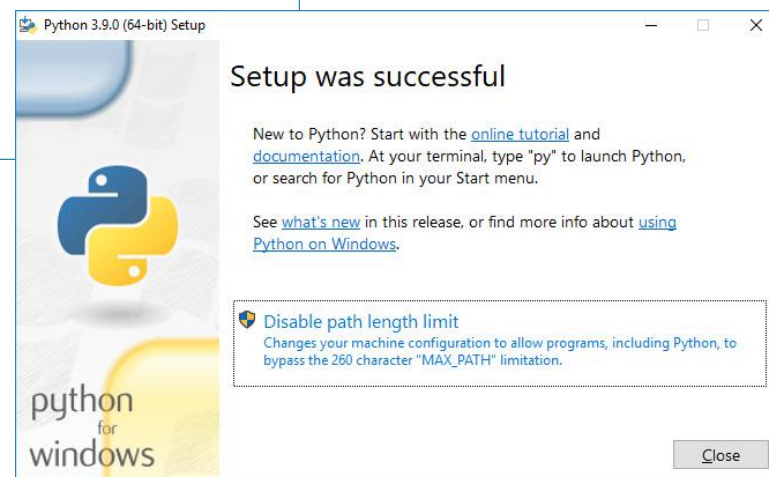
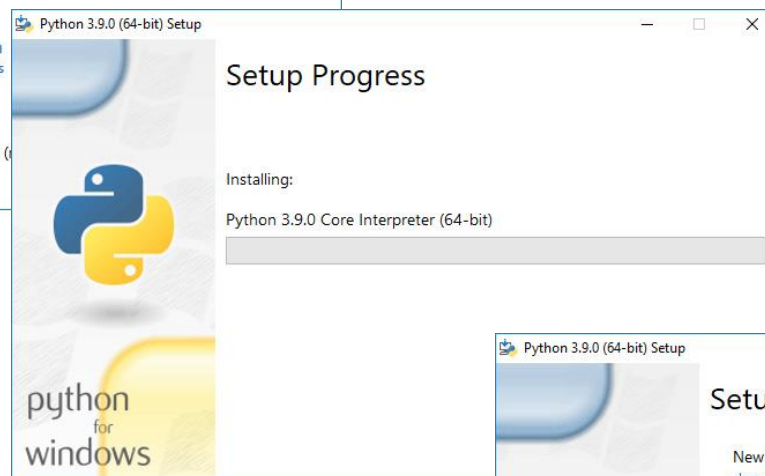
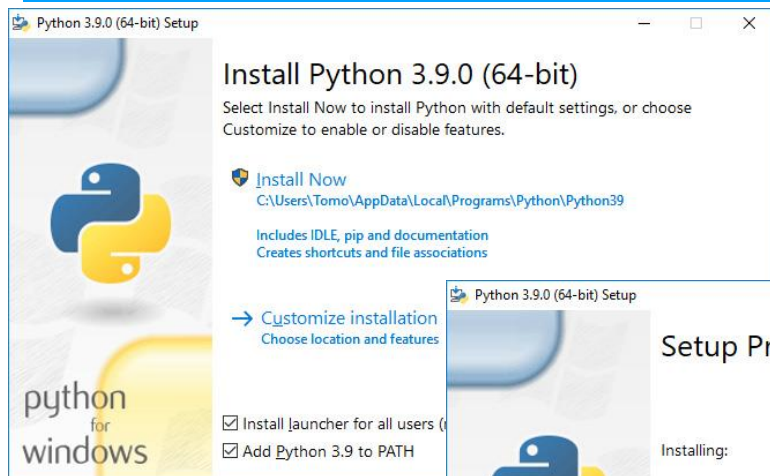
Want to help test development versions of Python 3.13? [Prereleases](#), [Docker images](#)

Active Python Releases

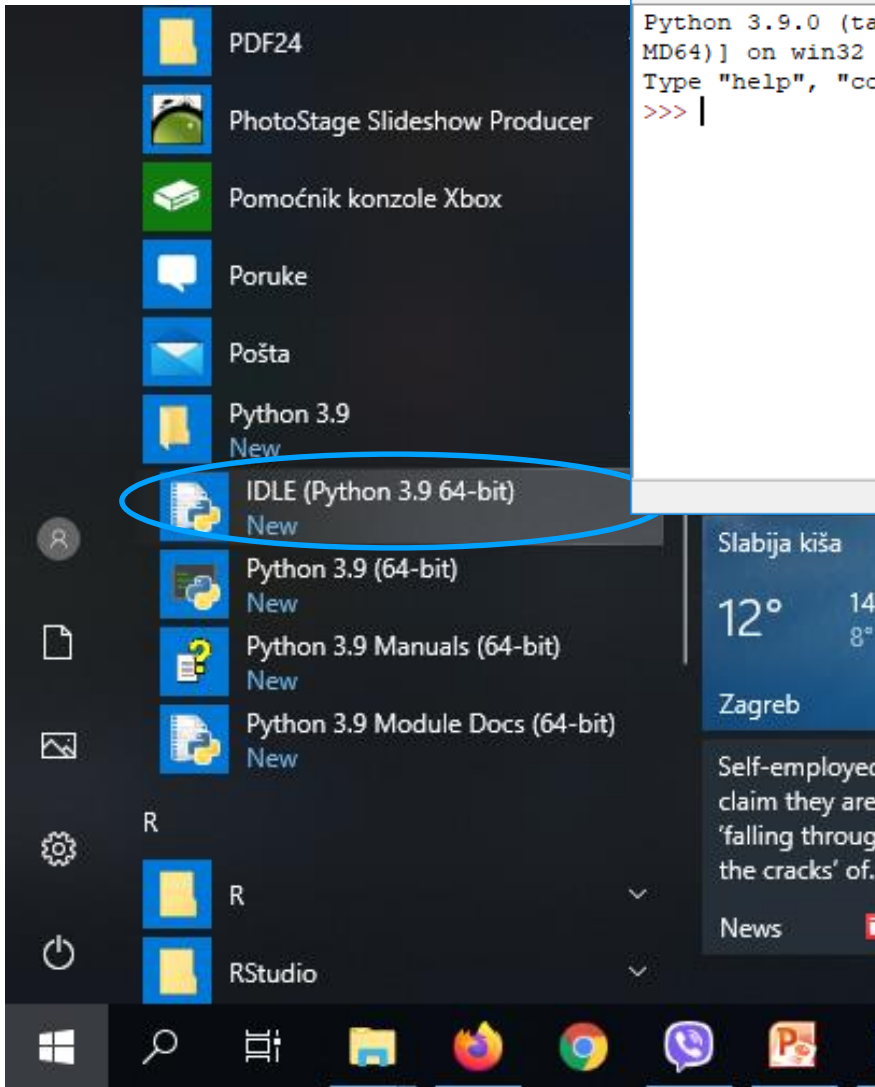
For more information visit the [Python Developer's Guide](#).

Python version	Maintenance status	First released	End of support	Release schedule
3.13	prerelease	2024-10-01 (planned)	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693

Instalacija Pythona 3.12.0



Pokretanje Python 3.12 IDLE



```
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.
>>> |
```

Ln: 3 Col: 4



□ Integrated Development Environment

Eric Idle
- komičar
iz Monty
Pythona



Osnovni tipovi podataka u Pythonu



- **int** – cijeli broj
- **float** – broj s pomičnom točkom
- **str** – niz znakova (string)
- **bool** – logički tip podatka

Cijeli brojevi



□ Primjeri cijelih brojeva

```
>>> 12
12
>>> 10000
10000
```

```
>>> -12
-12
```

```
>>> 0012
SyntaxError: invalid token
```

GREŠKA: kod Pythona ne možemo unositi vodeće nule, crvenom bojom je ispisana greška, a crvenom bojom pozadine je označen dio koda gdje se dogodila greška

Python unos ili rezultat neke naredbe ispisuje fontom plave boje, pa se lako može prepoznati što smo mi unijeli, a što je Python ispisao

□ Nije ograničen broj znamenaka cijelog broja

```
>>> 123456789012345678901234567890123456789012345678901234
123456789012345678901234567890123456789012345678901234
```

Brojevi s pomičnom točkom



□ Primjeri brojeva s pomičnom točkom

```
>>> 3.2          >>> -3.          >>> 0.32
3.2             -3.0         0.32
>>> 0.00032     >>> .000032    >>> 1e2
0.00032        3.2e-05     100.0
>>> 1e15        >>> 1e16
10000000000000000.0  1e+16
```

brojevi s pomičnom točkom imaju granicu

□ Donja granica brojeva s pomičnom točkom

```
>>> 1.2345678901234567e-323  >>> 1.2345678901234567e-324
1e-323                        0.0
```

□ Gornja granica brojeva s pomičnom točkom

```
>>> 1.23456789012345678e308  >>> 1.23456789012345678e308
1.2345678901234567e+308      inf
```

Aritmetički operatori



zbrajanje	+
oduzimanje	-
množenje	*
dijeljenje	/
cjelobrojno dijeljenje	//
modulo (ostatak od dijeljenja)	%
potenciranje	**

- Prvenstvo pri izvođenju ima potenciranje, pa nakon toga množenje, dijeljenje, cjelobrojno dijeljenje i modulo, te na kraju zbrajanje i oduzimanje

Aritmetički izrazi



□ Jednostavni aritmetički izrazi

```
>>> 3 + 2
5
>>> 2 / 3
0.6666666666666666
>>> 2 // 3
0
>>> 3 - 4
-1
>>> 2 % 3
2
>>> 3 * 2
6
>>> 2 ** 3
8
>>> 2 +
SyntaxError: invalid syntax
```

□ Složeni aritmetički izrazi

```
>>> 2 + 2 * 3
8
>>> (2 + 2) * 3
12
>>> (2 + 2) * 3 + 2 / 3
12.666666666666666
```

GREŠKA: ako ne napišemo dobro aritmetički izraz prikazati će nam se greška

Zadatak: Razlomak



- Napišite jednu naredbu koja će izračunati vrijednost sljedećeg razlomka:

$$\frac{2 + 1}{2} + 2(3 + 1) - 4$$
$$(2 + 3.1) \frac{2}{3} + 1$$

Vrijeme



- Rješenje:

```
>>> ((2+1)/2 + 2*(3+1) - 4) / ((2+3.1)*2/3 + 1)
1.25
```

brojnik razlomka
unutar zagrada

nazivnik razlomka
unutar zagrada



Nizovi znakova



□ Jednostruki ili dvostruki navodnici

```
>>> 'Python'  
'Python'
```

```
>>> "Python"  
'Python'
```

nizovi znakova su
označeni fontom
zelene boje

□ Ispis dvostrukih navodnika u nizu znakova

```
>>> 'Radionica "Python" za \"osnovne škole\"'  
'Radionica "Python" za "osnovne škole"'
```

□ Ispis jednostrukih navodnika u nizu znakova

```
>>> "Radionica 'Python' za \"osnovne škole\""  
"Radionica 'Python' za 'osnovne škole'"
```

Nizovi znakova – funkcija print



- Funkcija je definirani skup naredbi
- Opći oblik funkcije u Pythonu

```
naziv_funkcije(parametar1, parametar2, ... , parametarN)
```

- Funkcija može primiti 0, 1 ili više parametara
- Funkcija **print**

```
>>> print()
```

```
>>> print('Radionica', 'Python', 2023)  
Radionica Python 2023
```

```
>>> print('Python')  
Python
```

funkcije print kao
parametre može primiti
različite tipove podataka

standardne Python
funkcije su označene
fontom ljubičaste boje

Nizovi znakova – funkcija print



□ Aritmetički izrazi u ispisu

```
>>> print('Zbroj brojeva', 2, 'i', 3, 'je:', 2 + 3)
Zbroj brojeva 2 i 3 je: 5
```

□ Ispis lijevo nakošene crte \

```
>>> print('Nakošena crta - \\.')
Nakošena crta - \.
```

□ Tabulator - \t

```
>>> print('Korištenje\ttabulatora\tu\tPythonu.')
Korištenje      tabulatora      u      Pythonu.
```

□ Prelazak u novi red pri ispisu - \n

```
>>> print('Prelazak\nu novi red u Pythonu.')
Prelazak
u novi red u Pythonu.
```

Varijable

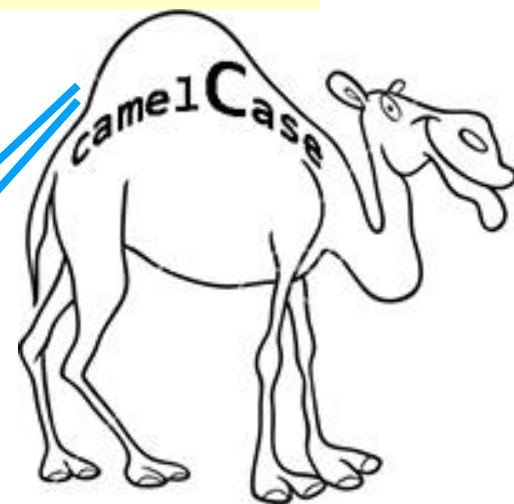


- Varijabla je memorijska lokacija kojoj pristupamo preko njenog naziva, a na njoj je zapisana vrijednost koja se može mijenjati
- Varijabla ima naziv i vrijednost

```
>>> varijabla = 10  
>>>
```

Diagram showing the code above with annotations: a blue oval around 'varijabla' is labeled 'naziv varijable' (variable name), and a blue oval around '10' is labeled 'vrijednost varijable' (variable value). Arrows point from the labels to their respective parts in the code.

loši nazivi	dobri nazivi
aaaaa	brojac
abcdefgh	ime_prezime
ahauifhasfuhsaiu	godinaRodjenja
hfjhds3u4444	imeNajPriatelja



Varijable



- Pravila za imenovanje varijabli:
 - Naziv varijable može sadržavati slova, brojeve i podvlake
 - Naziv varijable ne smije počinjati s brojem
 - Naziv varijable ne smiju biti ključne riječi za koje su rezervirani nazivi, kao što **bool**, **True**, **False**, ...
 - Naziv varijable smije sadržavati naše znakove (čćžšđČĆŽŠĐ), ali se to **nikako ne preporuča**
 - Python razlikuje velika i mala slova, pa su **x** i **X** dvije različite varijable

Pridruživanje vrijednosti varijablama

□ Znak pridruživanja =

```
>>> x = 10
>>> print(x)
10
>>> x = x + 20
>>> print('x =', x)
x = 30
>>> y = -2.3
>>> print(y)
-2.3
>>> y = y * 2
>>> print('y =', y)
y = -4.6
```

□ U varijable se mogu spremiti i nizovi znakova

```
>>> godina = 2015
>>> radionica = 'Liga progamiranja u Pythonu'
>>> print('Radionice', radionica, godina, 'oš')
Radionice Liga programiranja u Pythonu 2015 oš
```


Unos s tipkovnice



- Funkcija **input**
- Pomoću funkcije **input** unesite vaše ime s tipkovnice, spremite ga u varijablu **ime**, te nakon toga ispišite vrijednost varijable **ime**.

```
>>> ime = input('Unesi svoje ime: ')
Unesite vaše ime: Tomo
>>> print('Uneseno ime je:', ime)
Uneseno ime je: Tomo
```

- Funkcija **input** sve što se unese s tipkovnice sprema kao niz znakova



Unos s tipkovnice

□ Primjer funkcije `input` s cijelim brojem

```
>>> broj = input('Unesi cijeli broj: ')
Unesi cijeli broj: 10
```

```
>>> broj + 10
```

```
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    broj+10
```

```
TypeError: Can't convert 'int' object to str implicitly
```

GREŠKA: broj 10 unesen s tipkovnice je spremljen kao niz znakova

□ Funkcija `int` – pretvara u cijeli broj

```
>>> broj = input('Unesi cijeli broj: ')
Unesi cijeli broj: 10
```

```
>>> broj = int(broj)
```

```
>>> broj + 10
```

```
20
```

skraćeno se može pisati:
`broj = int(input('Unesi cijeli broj: '))`

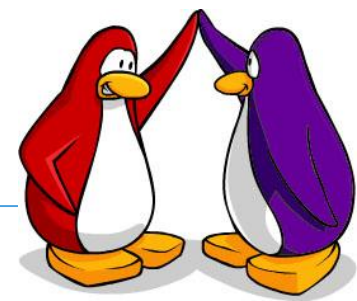
za pretvorbu u broj s pomičnom točkom koristi se funkcija `float`

Program



- ❑ Naredbe smo dosad unosili i odmah pokretali u Python IDLE-u
- ❑ Što će se dogoditi ako zatvorimo Python IDLE?
- ❑ Izgubili smo sve naredbe koje smo unosili
- ❑ Program je skup naredbi čijim se izvršenjem obavlja neki posao
- ❑ Naredbe možemo spremiti kao poseban program, pa taj program možemo naknadno ažurirati i pokretati

Prvi Python program



Python IDLE

Python IDLE editor

komentari

kod programa

ekstenzija .py

unos naziva programa

spremanje programa

```
#Jednolinijski komentari
#Ovo je moj prvi Python program.

"""
Komentar u više linija.
Moгу se koristiti i jednostruki navodnici.
Ovo je moj prvi Python progr.
"""

'Ovo je također jednolinijski komentar.'
'Ovo je moj prvi Python program.'
```

print('Hello World!')

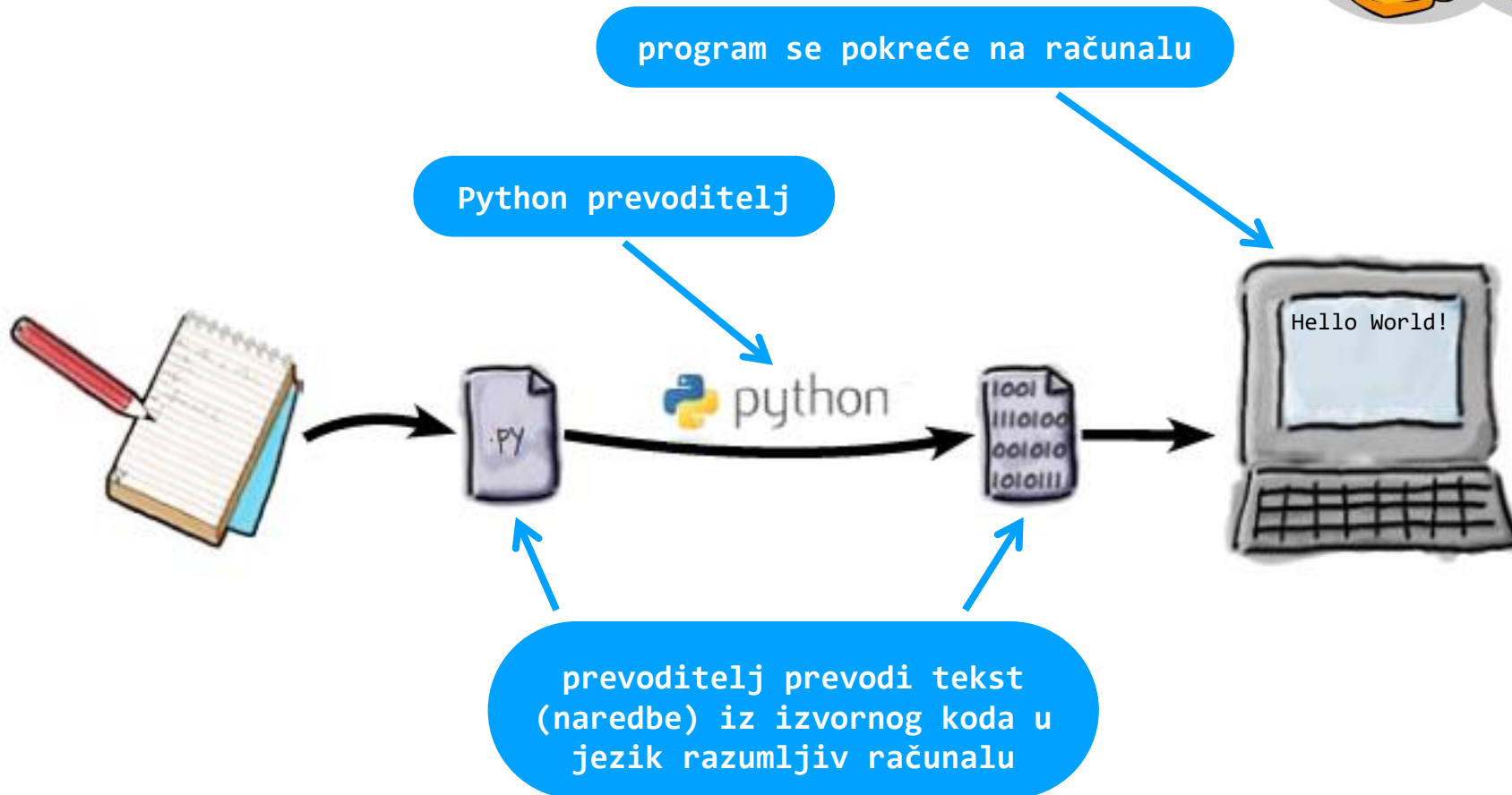
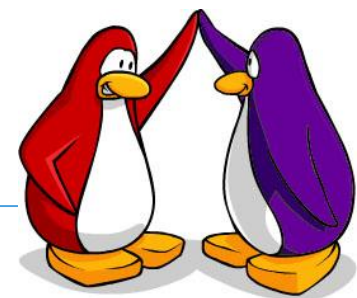
Save in: python_primeri

Name	Date modified	Type
No items match your search.		

File name: helloworld.py

Save as type: Python files (*.py;*.pyw)

Prevođenje programa



Prilagođeno iz: P. Barry & D. Griffiths, Head First Programming, O'Reilly, 2009

Pokretanje programa u Python IDLE-u

The image shows two windows from the Python IDLE environment. The left window, titled 'Python 3.4.1: helloworld.py - D:/python_primjeri/helloworld.py', contains a Python script with a multi-line comment and a print statement. The right window, titled 'Python 3.4.1 Shell', shows the output of the script: 'Hello World!'. A blue callout box points to the 'Run Module' option in the 'Run' menu of the script window, with the text 'pokretanje programa'. Another blue callout box points to the 'Run Module' option with the text 'prečac na tipkovnici: funkcijska tipka F5'. The shell window has a blue box around the output text.

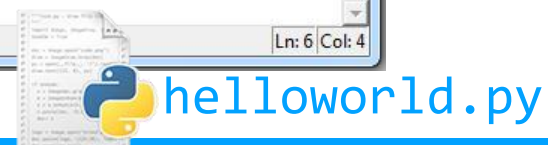
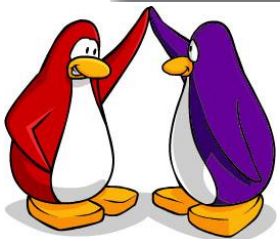
```
#Jednolinijski
#Ovo je moj pr

"""
Komentar u više linija
Mogu se koristiti i jednobraki narodnici.
Ovo je moj prvi Python progr.
"""

'Ovo je također jednolinijski komentar.'
'Ovo je moj prvi Python program.'

print('Hello World!')
```

```
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Hello World!
>>> |
```



Program: Unos imena



- Naredbe za unos imena i ispis imena na ekran iz primjera spremite u program `ime.py`, te pokrenite program u Python IDLE-u.

```
ime = input('Unesite vaše ime: ')\nprint('Uneseno ime je: ', ime)
```

unosi se ime
s tipkovnice

```
File Edit Shell Debug Options Windows Help\nPython 3.4.1 (vs.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32\nType "copyright", "credits" or "license()" for more information.\n>>> ===== RESTART =====\n>>>\nUnesite vaše ime: Tomo\nUneseno ime je: Tomo\n>>>
```

ispis unesenog imena



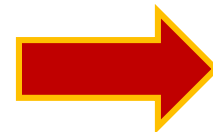
Zadatak: Prosjek



- Napišite program koji računa prosječnu ocjenu iz četiri predmeta. Ocjene se unose s tipkovnice. Ispišite dobivenu prosječnu ocjenu ispišite. Pokrenite i testirajte vaš program.

$$\frac{ocjena1 + ocjena2 + ocjena3 + ocjena4}{ukupan_broj_ocjena}$$

Vrijeme





Zadatak: Prosjek - rješenje

```
ocjena1 = int(input('Unesi prvu ocjenu: '))
ocjena2 = int(input('Unesi drugu ocjenu: '))
ocjena3 = int(input('Unesi treću ocjenu: '))
ocjena4 = int(input('Unesi četvrtu ocjenu: '))
prosjek = (ocjena1 + ocjena2 + ocjena3 + ocjena4)/4
print('Prosjek ocjena:', prosjek)
```

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Unesi prvu ocjenu: 3
Unesi drugu ocjenu: 5
Unesi treću ocjenu: 4
Unesi četvrtu ocjenu: 5
Prosjek ocjena je: 4.25
>>> |
```



prosjek.py

Relacijski operatori



veće od	>
manje od	<
veće od ili jednako	>=
manje od ili jednako	<=
jednako	==
nije jednako	!=

- Relacijski operatori uspoređuju dva operanda. Rezultat usporedbe ima vrijednosti **True** ili **False** (Istina ili Laž).

Relacijski operatori

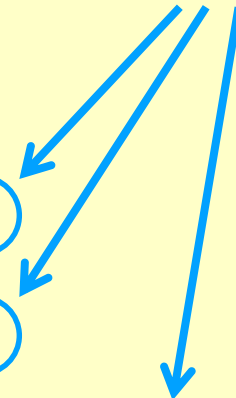


□ Provjeriti kako djeluju operatori:

```
>>> 3 > 2
True
>>> 3 < 2
False
>>> 3 >= 2
True
>>> 3 <= 2
False
>>> 3 == 2
False
>>> 3 != 2
True
```

```
>>> a = 2
>>> b = 7
>>> b > a
True
>>> b + 1 == a * 4
True
>>> b / a != b // a
True
>>> (a + b) ** 2 <= 10 * a
False
```

prvo se izračunaju aritmetički izrazi s lijeve i desne strane, pa se izvršava usporedba



Donošenje odluka u programima

- Izvođenje jedne ili više naredbi na temelju ispitivanja nekog uvjeta:

```
...
ako je uvjet onda
    naredba1_1
    ...
    naredba1_n
...

```

- Odabir jedne od dvije mogućnosti (izvršava se samo jedan blok naredbi):

```
...
ako je uvjet onda
    naredba1_1
    ...
    naredba1_n
inače
    naredba2_1
    ...
    naredba2_m
...

```

- Višestruki izbor izvršava se samo jedan od blokova naredbi):

```
...
ako je uvjet_1 onda
    blok_naredbi_1
inače ako je
uvjet_2 onda
    blok_naredbi_2
...
inače ako je
uvjet_n onda
    blok_naredbi_n
inače
    blok_naredbi
...

```


Donošenje odluka u Pythonu

- Izvođenje jedne ili više naredbi na temelju ispitivanja nekog uvjeta:

```
...
if uvjet:
    naredba1_1
    ...
    naredba1_n
...
```

- Odabir jedne od dvije mogućnosti (izvršava se samo jedan blok naredbi):

```
...
if uvjet:
    naredba1_1
    ...
    naredba1_n
else:
    naredba2_1
    ...
    naredba2_m
...
```

- Višestruki izbor izvršava se samo jedan od blokova naredbi):

```
...
if uvjet_1:
    blok_naredbi_1
elif uvjet_2:
    blok_naredbi_2
...
elif uvjet_n:
    blok_naredbi_n
else:
    blok_naredbi
...
```

Primjer: Veći broj v1



...

if uvjet:

→ naredba1_1

→ ...

→ naredba1_n

...

izvršit će se ako je *uvjet* zadovoljen (logički izraz je istinit)

svaki redak koji će se izvesti ako je *uvjet* zadovoljen mora biti uvučen. Najbolje je koristiti tipku TAB

```
A = int(input('Unesi broj A: '))
```

```
B = int(input('Unesi broj B: '))
```

```
if A > B:
```

```
    print('A je veće od B')
```

```
print('Kraj programa!')
```



veci_broj_v1.py

Primjer: Veći broj v2

izvršit će se samo JEDAN od ova dva bloka naredbi!



...

if uvjet:

→ naredba1_1

→ ...

→ naredba1_n

else:

→ naredba2_1

→ ...

→ naredba2_m

...

izvršit će se ako je *uvjet* zadovoljen (logički izraz je istinit)

izvršit će se ako *uvjet* nije zadovoljen (logički izraz je lažan)

```
A = int(input('Unesi broj A: '))  
B = int(input('Unesi broj B: '))
```

```
if A > B:  
    print('A je veće od B')  
else:  
    print('A nije veće od B')  
print('Kraj programa!')
```

Najbolje je koristiti tipku TAB



veci_broj_v2.py

Primjer: Veći broj v3



izvršit će se
samo JEDAN od
blokova
naredbi!

```
...  
if uvjet_1:  
    blok naredbi_1  
elif uvjet_2:  
    blok naredbi_2  
...  
elif uvjet_n:  
    blok naredbi_n  
else:  
    blok naredbi  
...
```

```
A = int(input('Unesi broj A: '))  
B = int(input('Unesi broj B: '))  
  
if A > B:  
    print('A je veće od B')  
elif A == B:  
    print('A i B su jednaki')  
else:  
    print('A je manje od B')  
print('Kraj programa!')
```



veci_broj_v3.py

Zadatak: tko je viši?



- Dube i Ivo se ne mogu dogovoriti tko je viši, pa to treba riješiti programom. Potrebno je unijeti Dubinu i Ivovu visinu (u cm), i onda ispisati tko je viši i za koliko cm, ili da su ipak jednake visine.



Vrijeme



Zadatak: tko je viši? - rješenje



```
visDube = int(input('Dubina visina: '))
visIvo = int(input('Ivova visina: '))
if visDube > visIvo:
    print ('Dube je viša za', visDube-visIvo, 'cm' )
elif visDube < visIvo:
    print ('Ivo je viši za', visIvo-visDube, 'cm' )
else:
    print ('Ivo i Dube su iste visine!')
```



visina.py

Three screenshots of a Python 3.4.1 Shell window showing the execution of the program. The first screenshot shows the program running with inputs 155 and 149, resulting in the output 'Dube je viša za 6 cm'. The second screenshot shows the program running with inputs 139 and 151, resulting in the output 'Ivo je viši za 12 cm'. The third screenshot shows the program running with inputs 162 and 162, resulting in the output 'Ivo i Dube su iste visine!'. Each screenshot also shows the program's header information and a prompt to type 'copyright', 'credits', or 'license()'.

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014 [Intel]) on win32
Type "copyright", "credits" or "license()" for more
>>> ===== RESTART =====
>>>
Dubina visina: 155
Ivova visina: 149
Dube je viša za 6 cm
>>> |

Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014 [Intel]) on win32
Type "copyright", "credits" or "license()" for more
>>> ===== RESTART =====
>>>
Dubina visina: 139
Ivova visina: 151
Ivo je viši za 12 cm
>>> |

Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014 [Intel]) on win32
Type "copyright", "credits" or "license()" for more
>>> ===== RESTART =====
>>>
Dubina visina: 162
Ivova visina: 162
Ivo i Dube su iste visine!
>>>
```

Logički operatori i logički izrazi



- ❑ Što ako je uvjet na temelju kojeg treba donijeti odluku složen?
- ❑ Logički operatori:

logička I operacija	and
logička Ili operacija	or
NE operacija (negacija)	not

- ❑ Redoslijed izvođenja logičkih operacija:

1.	not
2.	and
3.	or

Logički operatori i logički izrazi



□ Primjeri logičkih operacija:

```
>>> a = 2
>>> b = 3
>>> c = 10
>>> a > b
False
>>>
>>> c > b
True
>>>
>>> a > b and c > b
False
>>>
>>> a > b or c > b
True
```

za logičku **AND** operaciju rezultat će biti **True** (istina) samo ako su oba izraza True (istinita)

za logičku **OR** operaciju rezultat će biti **True** (istina) već ako je jedan od izraza True (istinit)

Redosljed izvođenja operacija



1.	aritmetički
2.	relacijski
3.	logički

Ako imamo kombinirane aritmetičke, relacijske i logičke operatore, onda je ovo redosljed izvođenja operacija.

```
>>> a=2
>>> b=3
>>> c=10
>>> a+2*3 >= c or not(a > b) and a*b-2 == c%6
True
>>>
>>> (a+2*3 >= c) or (not(a > b) and (a*b-2 == c%6))
True
>>>
```

Ako ipak nismo posve sigurni u redosljed operacija onda je najbolje koristiti zagrade!

Zadatak: Skok u dalj



- Na natjecanju u skokovima u dalj organizatori su odlučili podijeliti više zlatnih, srebrnih i brončanih medalja prema sljedećim kriterijima:
 - Brončana medalja za sve koji preskoče između 5,5 i 6,5 metara, uključujući skokove od 5,5 i 6,5 metara
 - Srebrna medalja za sve koji preskoče preko 6,5 metara i manje od 7 metara
 - Zlatna medalja za sve koji preskoče 7 metara i preko
- Ispisati koju je medalju osvojio natjecatelj, te poruku ako nije osvojio medalju.





Zadatak: Skok u dalj - rješenje

```
skok = float(input('Unesi duljinu skoka: '))
```

```
if skok >= 5.5 and skok <= 6.5:  
    print('Brončana medalja')  
elif skok > 6.5 and skok < 7:  
    print('Srebrna medalja')  
elif skok >= 7:  
    print('Zlatna medalja')  
else:  
    print('Nije osvojena medalja')
```

```
Python 3.5.0 Shell  
File Edit Shell Debug Options Window Help  
mjeri_zadaci/skok_u_dalj.py  
Unesi duljinu skoka: 4  
Nije osvojena medalja  
>>>  
RESTART: D:/Google disk/futura/radi  
onice/Liga_programiranja_2015/01-pri  
mjeri_zadaci/skok_u_dalj.py  
Unesi duljinu skoka: 5.6  
Brončana medalja  
>>>  
RESTART: D:/Google disk/futura/radi  
onice/Liga_programiranja_2015/01-pri  
mjeri_zadaci/skok_u_dalj.py  
Unesi duljinu skoka: 6.7  
Srebrna medalja  
>>>  
RESTART: D:/Google disk/futura/radi  
onice/Liga_programiranja_2015/01-pri  
mjeri_zadaci/skok_u_dalj.py  
Unesi duljinu skoka: 8  
Zlatna medalja  
>>> |  
Ln: 60 Col: 4
```



skok_u_dalj.py

Ne zaboravite!

- Za 15 dana – u utorak 21.11.2023. –
1. kolo Lige programiranja
- **5./6. razredi** ekipe od **3** učenika (ili barem 2)
početak **17:00**
- **7./8. razredi** pojedinačno
početak **17:00**
- **3** zadatka rješavate **75** minuta
- nemojte kasniti!

