

Programiranje 1

Uvod u programski jezik Python

Univerzitet u Beogradu
Elektrotehnički fakultet
2020/2021.

Sadržaj

- Uvod i namena
- Osnovne osobine jezika
- Izvršavanje koda - Python interpreter
- Tipovi i objekti
- Izrazi, dodele vrednosti i konverzije
- Osnovni ulaz i izlaz
- Korišćenje pomoći i ugrađenih funkcija

Uvod (1)

- Python je osmislio holandski programer Guido van Rossum 80-tih godina XX veka
 - Ime dobio zbog fascinacije autora engleskom humorističkom serijom *Monty Python's Flying Circus*
- Python je interpretirani, programski jezik visokog nivoa
 - Podržava proceduralnu, imperativnu, objektno-orijentisanu i funkcionalnu paradigmu programiranja
 - Podržava određene mogućnosti neinterpretiranih jezika



Uvod (2)

- Jezik je dizajniran da bude:
 - Jednostavan za učenje
 - Visoko proširiv
 - Zabavan za upotrebu
- Filozofija jezika je sadržana u dokumentu The Zen of Python koji sadrži principe kao što su:
 - Beautiful is better than ugly.
 - Explicit is better than implicit.
 - Simple is better than complex.
 - Complex is better than complicated.
 - Readability counts.
- "There should be one — and preferably only one — obvious way to do it" filozofija jezika
- Dobija se sa *import this*

Verzije jezika (1)

- Dve važne grane jezika
 - 2.x i 3.x
 - Najavljeno gašenje podrške verziji 2.x u 2015.
 - Odloženo do 2020. godine, 1. januara 2020 "Python 2.x sunset"
 - Velika količina koda napisana u verziji 2.x
- Verzija 3.x donosi značajna poboljšanja na nivou jezika
 - Bolje definisani, čistiji konstrukti
 - Nešto drugačiji način upotrebe pojedinih operatora
 - Izmene u ponašanju pojedinih tipova podataka
- Na kursu se izučava verzija 3.x

Verzije jezika (2)

- Problemi kompatibilnosti koda
 - Programi pisani u okruženju 3.x se ne izvršavaju u starijim, 2.x okruženjima
 - Programi napisani u 2.x okruženjima mogu imati problema prilikom izvršavanja u 3.x okruženjima
 - Postoji alat za translaciju koda
- Verzija jezika ne igra bitnu ulogu kod učenja programiranja
 - Sintaksa u verziji 3.x je lakša za usvajanje

Namena

- Python je jezik opšte namene
- U njemu se mogu razvijati sve vrste aplikacija
 - Primene u inženjerskim, naučnim, poslovnim i web aplikacijama
 - Prototipovi, ali i produkcione aplikacije
 - Veliki izbor gotovih biblioteka i rešenja
 - Dobar interfejs ka drugim programskim jezicima i bibliotekama za poboljšanje performansi izvršavanja
- Postao veoma popularan tokom godina
 - Ušao u top-3 najpopularnijih jezika po različitim istraživanjima <https://youtu.be/Og847HVwRSI> (u novembru 2020 drugi iza C-a na TIOBE)
 - Koristi se u kompanijama poput Google, Yahoo, Amazon i institucijama poput CERN, NASA i sl.

Osnovne osobine

- Interpretirani programski jezik
- Razlikuje mala i velika slova prilikom pisanja koda
 - *Case-sensitive* jezik
- Promenljive se eksplicitno ne deklarišu
- Ne postoji terminator naredbi
 - Uobičajeno se piše jedna naredba u jednom redu
 - Ukoliko se piše više naredbi u jednom redu, koristi se separator naredbi tačka-zarez (;)
- Blokovi koda se zadaju identacijom (uvlačenjem)
 - Koristi se za grupisanje naredbi u blokove
 - Sve naredbe uvučene isti broj belih znakova čine jedan blok koda
- Dozvoljeno pisanje praznih linija
 - Ignorišu se

Rezervisane reči

- Relativno mali skup rezervisanih reči
- <https://www.programiz.com/python-programming/keyword-list#as>
 - Rezervisane reči ne mogu biti identifikatori

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

Komentari u kodu

- Dva načina za zadavanje komentara u kodu
- Jednoredni komentari iza simbola `#` (*hash*)
`#primer jednorednog komentara`
- Višeredni komentari unutar
uzastopnih trostrukih znakova navoda `"""` ili `' '`
 - *heredocs / docstrings*
 - Upotrebljavaju se najčešće za dokumentovanje funkcija
`"""Ovo je
višeredni komentar.
Zadržava sve bele znakove i
prelaske u novi red"""`

Python interpreter (1)

- Program napisan u višem programskom jeziku predstavlja izvorni kod
- Izvorni kod se mora prevesti u mašinski jezik da bi bio izvršiv
 - Prevođenjem ili interpretiranjem
- Prevodilac direktno prevodi izvorni kod u mašinski, objektni kod
 - Može se pokretati i izvršavati na ciljnoj arhitekturi
- Interpreter je program koji izvršava naredbe programskog jezika korišćenjem svog skupa instrukcija
 - Program se izvršava naredbu po naredbu koje se prevode u trenutku izvršavanja
 - Dodatan nivo apstrakcije

Python interpreter (2)

- Prednost korišćenja interpretera su:
 - Brži razvoj i testiranje programa
 - Nema potrebe za stalnim prevođenjem programa u mašinski oblik
 - Lakše utvrđivanje grešaka
 - Čim se pojavi greška, interpreter staje sa radom
- Nedostatak interpretera su lošije performanse izvršavanja zbog postojanja dodatnog sloja
 - Ispravlja se prevođenjem produkcione verzije koda ili *Just-In-Time* (JIT) prevođenjem u modernim sistemima

Python interpreter (3)

- Više implementacija za Python
 - CPython – referentna, podrazumevana implementacija napisana u programskom jeziku C
 - Druge implementacije: Jython, PyPy, IronPython, itd.
- Python interpreter izvršava:
 - Direktne zadate naredbe u interaktivnom režimu rada
 - Napisane skripte i module sačuvane u datoteci
- Skripta je datoteka koja sadrži sekvencu naredbi namenjenih za direktno izvršavanje
- Modul je deo koda namenjen za uvoz u druge Python datoteke ili skripte

Python interpreter (4)

- Python interpreter izvršava skriptu u tri koraka:
 1. Redom prođe kroz skriptu kako bi utvrdio sintaksnu korektnost
 2. Prevede izvorni kod u međukod (*bytecode*)
 - Kako bi se omogućilo sprovođenje optimizacija
 - Ukoliko se program ne menja između dva izvršavanja, ovaj korak se preskače
 3. Pošalje kod na izvršavanje izvršnom okruženju
 - Python Virtual Machine (PVM)
 - Izvršava kod naredbu po naredbu

Python razvojna okruženja

- Integrirano razvojno okruženje čini skup alata koji se koriste prilikom razvoja programa
 - *Integrated Development Environment* (IDE)
 - Editor teksta, debugger, prevodilac ili interpreter i sl.
- Dobro podržana razvojna okruženja za Python
 - IDLE
 - Besplatno okruženje koje obično dolazi uz instalaciju jezika
 - Interaktivan rad u konzoli
 - PyCharm Community Edition (<https://www.jetbrains.com/pycharm/download/>)
 - PyScripter (<https://sourceforge.net/projects/pyscripter/>)
 - IntelliJ Idea
 - Potreban Python plugin

HelloWorld program (1)

- Pokrenuti interaktivni režim Python interpretera

- IDLE okruženje
- Komanda python ili python3 u konzoli (komandnoj liniji) operativnog sistema

```
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.
```

```
>>>
```

- Otkucati:

```
>>> print("HelloWorld!")  
HelloWorld!
```


HelloWorld program (2)

- Interaktivno okruženje omogućava da se odmah testira deo koda koji se napiše:

```
>>> 2 + 5
```

```
7
```

```
>>> print("Na ETF se uci Pajton")
```

```
Na ETF se uci Pajton
```

- Izlazak iz interaktivnog okruženja:
 - Ugrađene funkcije `quit()` ili `exit()`
 - Kombinacija tastera Ctrl-Z + Enter (*Windows*) ili Ctrl-D (*Linux*)

HelloWorld program (3)

- Program se može izvršiti i kao skripta
 - Snimiti program u datoteku `hello.py`
 - `.py` je uobičajena ekstenzija za Python kod

```
#!/usr/bin/env python3
print("HelloWorld!")
```
- Pokrenuti skriptu iz konzole (komandne linije)
 - Windows: `python hello.py`
 - Linux: `python3 hello.py`
 - Linija obeležena sivom bojom u skriptu predstavlja tzv. *shebang* upravljačku liniju i koristi se na Linux sistemima za lociranje interpretera

Tipovi i objekti (1)

- Podatak je objekat (entitet) kome se programskim putem pristupa putem imena (identifikatora)
 - Uglavnom ima lokaciju u memoriji računara
- Tip podatka predstavlja način tumačenja memorijskog sadržaja koji zauzima objekat
 - Precizira skup **vrednosti** i **operacija** nad podacima
 - Osnovni tipovi: `int`, `bool`, `float`
 - Složeni tipovi:
stringovi (nizovi znakova), liste, torke, rečnici, skupovi

Tipovi i objekti (2)

- Promenljiva je podatak kome je moguće menjati vrednost
 - Zauzima prostor u memoriji
 - Sadržaj memorije se tumači prema tipu promenljive
 - Pristupa se putem imena promenljive (identifikatora)
- Primer

```
>>> cena_kafe = 99.99
>>> broj_kafa = 3
>>> cena_porudzbine = cena_kafe * broj_kafa
>>> cena_porudzbine = 1.2 * cena_porudzbine
```

Tipovi i objekti (2)

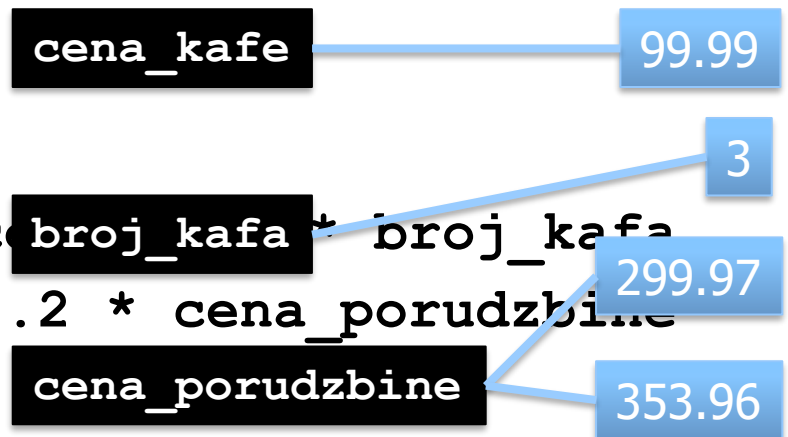
- Promenljiva je podatak kome je moguće menjati vrednost
 - Zauzima prostor u memoriji
 - Sadržaj memorije se tumači prema tipu primenljive
 - Pristupa se putem imena promenljive (identifikatora)
- Primer

```
>>> cena_kafe = 99.99
```

```
>>> broj_kafa = 3
```

```
>>> cena_porudzbine = cena_kafe * broj_kafa
```

```
>>> cena_porudzbine = 1.2 * cena_porudzbine
```



Tipovi i objekti (3)

- Promenljiva se može definisati
 - Ne definiše se eksplicitno, već dinamički nakon prve upotrebe
 - Ostatak programa je onda može koristiti
 - Može se koristiti specijalna vrednost **None** da označi odsustvo vrednosti
 - Međutim, promenljiva je definisana
 - Promenljivoj se može redefinisati tip
- Može biti lokalna ili globalna
 - Definiše opseg vidljivosti
 - Bitno kod upotrebe potprograma

Pravila imenovanja identifikatora (1)

- Ime promenljive (identifikator)
 - Može biti proizvoljne dužine
 - Može sadržati mala i VELIKA slova (UTF-8 kodni raspored), cifre 0-9 i znak *underscore* (`_`)
 - Razlikuju se mala i VELIKA slova
 - Ne sme početi cifrom, preporučuje se da počne malim slovom
 - Ne sme da bude rezervisana reč
 - Upotreba specijalnih karaktera poput `!`, `@`, `#`, `$`, `%` nije dozvoljena
- Pravila imenovanja važe i za druge tipove identifikatora
 - Imena potprograma i sl.

Pravila imenovanja identifikatora (2)

- Ne preporučuje se upotreba znaka *underscore* na početku ili kraju imena identifikatora
 - Obično se koristi za razdvajanje reči u promenljivama čije ime se sastoji od više reči
 - Postoje posebna pravila upotrebe takvih identifikatora
 - Ponekad predstavljaju ugrađene identifikatore
- Primeri imenovanja identifikatora:
 - Korektno: `Programiranje`, `ETF`, `lab_60`, `_si`
 - Nekorektno: `60si`
 - Diskutabilno: `__start__`

Dodela vrednosti promenljivama (1)

- Deklarisanje i definisanje promenljive se vrši prilikom prve dodele vrednosti promenljivoj
 - Koristi se operator =

```
>>> osnova = 2
>>> pi = 3.14
>>> fakultet = "ETF"
>>> print(osnova, pi, fakultet)
2 3.14 ETF
```
- Python dozvoljava višestruke dodele vrednosti:

```
a = b = c = 1
osnova, pi, fakultet = 2, 3.14, "ETF"
```

Dodela vrednosti promenljivama (2)

- Promenljivoj se može redefinisati tip novom dodelom vrednosti
 - Ugrađena funkcija `type()` za dohvaćanje trenutnog tipa podatka
- Primer

```
>>> a = 5                # type(a) → int
>>> type(a)
<class 'int'>
>>> b = 5.5              # type(b) → float
>>> type(b)
<class 'float'>
>>> a = b                # type(a) → float
>>> type(a)
<class 'float'>
```

Uništavanje promenljivih

- Uglavnom se obavlja dinamički
 - Python poseduje sakupljač đubreta (*garabage collector*)
- Eksplicitnom naredbom **del**

```
>>> a = 100
```

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

```
100
```

```
>>> del(a)
```

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

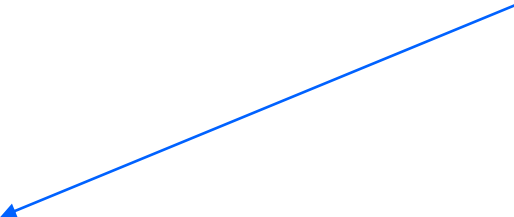
```
Traceback (most recent call last):
```

```
  File "<pyshell#21>", line 1, in <module>
```

```
    print(a)
```

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

Interpreter
odmah prekida
izvršavanje programa



Numerički tipovi podataka (1)

- Python podržava tri glavna numerička tipa
- **int**
 - Označeni celi brojevi
 - Opseg vrednosti: celi brojevi
 - *minint* ...-4, -3, -2, -1, 0, 1, 2, 3, 4,... *maxint*
- **float**
 - Realni brojevi u pokretnom zarezu
 - Opseg vrednosti: podskup skupa realnih brojeva
 - Problem tačnosti i zaokruživanja
- **complex**
 - Kompleksni brojevi
 - U obliku $x + yj$, gde x i y predstavljaju realne brojeve, a j označava imaginarni deo

Numerički tipovi podataka (2)

- Drugi oblici numeričkih konstanti

- **int**

- Binarne konstante prefiks 0b ili 0B
- Oktalne konstante prefiks 0o ili 0O
- Heksadecimalne konstante prefiks 0x ili 0X

- **float**

- Sufiks e ili E za eksponent

<code>int</code>	<code>float</code>	<code>complex</code>
<code>15</code>	<code>1.0</code>	<code>3 + 4j</code>
<code>-21</code>	<code>.9</code>	<code>4.3j</code>
<code>0b101</code> ili <code>0B101</code>	<code>-11.</code>	<code>-.3+0J</code>
<code>0o10</code> ili <code>0O10</code>	<code>3.4e3</code> ili <code>3.4E3</code>	<code>3e+2j</code>
<code>0x40</code> ili <code>0X40</code>	<code>-1.E-10</code>	<code>3.9e-4j</code>

Osnovni numerički operatori (1)

Operator	Opis	Primeri
- Negacija (unarno)	Menja znak operandu	-30 $b = -a$
+ Sabiranje	Sabira vrednosti dva operanda	$a + b = 31$
- Oduzimanje	Oduzima desni operand od levog operanda	$a - b = -11$
* Množenje	Množi vrednosti dva operanda	$a * b = 210$
/ Deljenje	Deli levi operand desnim operandom	$b / a = 2.1$
% Moduo	Određuje ostatak pri deljenju levog operanda desnim	$b \% a = 1$ $c \% a = 1.3$

U datim primerima, $a = 10$, $b = 21$, $c = 31.3$.

Osnovni numerički operatori (2)

Operator	Opis	Primeri
** Stepenovanje	Stepenovanje levog operanda desnim operandom	$a^{**}b = 10^{21}$
// Deljenje sa zaokruživanjem	Deljenje sa zaokruživanjem na ceo deo Prilikom deljenja, rezultat se zaokružuje na najbliži ceo na dole Ako je jedan od operanada negativan, zaokruživanje se vrši prema negativnoj beskonačnosti	$9//2 = 4$ $9.0//2.0 = 4.0$ $-11//3 = -4$ $-11.0//3 = -4.0$

Kombinovani operatori dodele

- Kombinuju operaciju sa dodelom vrednosti
 - Operacija se sprovedi nad levim i desnim operandom
 - Rezultat se smesti u levi operand

Operator	Ponašanje
<code>+=</code>	$c += a \leftrightarrow c = c + a$
<code>-=</code>	$c -= a \leftrightarrow c = c - a$
<code>*=</code>	$c *= a \leftrightarrow c = c * a$
<code>/=</code>	$c /= a \leftrightarrow c = c / a$
<code>%=</code>	$c \% = a \leftrightarrow c = c \% a$
<code>**=</code>	$c ** = a \leftrightarrow c = c ** a$
<code>//=</code>	$c //= a \leftrightarrow c = c // a$

Poredak izračunavanja izraza (1)

- Poredak izračunavanja izraza prati pravila prioriteta, konverzija i smer grupisanja operatora
- Mnemonik: **PEMDAS**
Parenthesis **()**, **E**xponentiation ******,
Multiplication *****, **D**ivision **/**,
Addition **+**, **S**ubtraction **-**

Tip operanda	Tip rezultata
int op int	int
float op int int op float	float
float op float	float

Operator	Smer grupisanja
**	←
* / %	→
+ -	→

Poredak izračunavanja izraza (2)

- Primeri

```
>>> 3**2*4*(2+3*8**2**3/2**3%4)+7*5/3
```

```
83.66666666666667
```

```
>>> 100.0/2/2
```

```
25.0
```

```
>>> 100/2*2
```

```
100.0
```

- U poslednjem primeru, obratiti pažnju da su $*$ i $/$ na istom nivou prioriteta
 - Izvršavaju (grupišu) se sleva na desno

EksPLICITNA konverzija tipa

- Potrebna da bi programer **eksplicitno** naznačio sa kojim tipom podataka želi da radi
- Sintaksa:
 - **tip**(**izraz**)
 - Konvertuje tip **izraza** u navedeni **tip**
- Primer konverzije realnog broja u ceo broj
 - **int(a)** vraća ceo deo argumenta

```
>>> real = 3.141
>>> ceo = int(real)
>>> print(real, ceo)
3.141 3
```

Nizovi znakova – tip `string`

- Python ima ugrađenu podršku za manipulaciju nizovima znakova (stringovima)
 - Veliki broj ugrađenih funkcija
 - Operator `+` za nadovezivanje dva stringa
 - Može se pristupati pojedinačnim karakterima
- Stringovi su nepromenljivi objekti
- Navode se između jednostrukih ili dvostrukih znakova navoda

```
>>> fakultet = "ETF"
>>> odsek = 'SI'
>>> print(fakultet, odsek)
ETF SI
>>> print(odsek + fakultet)
SIETF
```

Korišćenje ugrađene pomoći (1)

- Python sadrži ugrađenu funkciju `help()` za traženje pomoći o drugim funkcijama
 - Ispisuje zaglavlje funkcije za koju se pomoć traži
 - Daje uvid u argumente funkcije
 - Ispisuje kratak opis rada funkcije
- Korisno kada god korisnik ne zna kako neka funkcija radi
 - Naročito za specifične opcije

Korišćenje ugrađene pomoći (2)

- Primer traženja pomoći

```
>>> help(round)
```

```
Help on built-in function round in module  
builtins:
```

```
round(number, ndigits=None)
```

```
Round a number to a given precision in  
decimal digits.
```

```
The return value is an integer if ndigits is  
omitted or None. Otherwise the return  
value has the same type as the number.  
ndigits may be negative.
```

Osnovni izlaz (1)

- Funkcija `print()` ispisuje na izlaz vrednosti koje joj se zadaju kao argumenti
 - Ispisuje vrednosti razdvojene blanko znakom ' '
 - Na kraju se ispisuje znak za novi red '`\n`'

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

```
Help on built-in function print in module builtins:
```

```
print(...)
```

```
    print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

```
    Prints the values to a stream, or to sys.stdout by default.
```

```
    Optional keyword arguments:
```

```
    file:  a file-like object (stream); defaults to the current
           sys.stdout.
```

```
    sep:   string inserted between values, default a space.
```

```
    end:   string appended after the last value, default a newline.
```

```
    flush: whether to forcibly flush the stream.
```

Osnovni izlaz (2)

- Ponašanje se može promeniti putem argumenata
 - Polje **sep** definiše sadržaj (string) koji se umeće između argumenata za ispis kao separator
 - Polje **end** definiše sadržaj (string) koji se umeće nakon poslednjeg argumenata za ispis
- Primeri:

```
>>> osnova, pi, fakultet = 2, 3.14, "ETF"
>>> print(osnova, pi, fakultet)
2 3.14 ETF
>>> print(osnova, pi, fakultet, sep='##')
2##3.14##ETF
>>> print(osnova, pi, fakultet, sep='##', end='!')
2##3.14##ETF!
```


Osnovni izlaz (3)

- Polje (string) za ispis se može formatirati funkcijom **format()**
 - Omogućava umetanje vrednosti promenljivih u stringu (tekstu ispisa) na mestu vitičastih zagrada
- ```
print("Racun za {} kafa = {}"
.format(broj_soljica, cena_kafe))
```
- Vrednosti predate funkciji **format()** se uzimaju redom i umetaju umesto vitičastih zagrada
  - Može se i eksplicitno navesti koja se po redu navedena vrednost ispisuje
- ```
print("Cena jedne kafe = {1}.  
Ukupni iznos = {0} x {1} = {2}"  
.format(broj_soljica, cena_kafe,  
(broj_soljica*cena_kafe)))
```

Osnovni ulaz (1)

- Funkcija `input()` čita tekst sa standardnog ulaza i sačuva kao string
 - Tekst sa ulaza se mora konvertovati eksplicitnom konverzijom
 - Uz moguće greške prilikom konverzije tipova
 - Opcioni parametar definiše poruku koja se može ispisati korisniku prilikom učitavanja
 - Ne ispisuje novi red
- Primeri:

```
>>> str = input("Ucitaj broj: ")
Ucitaj broj: 4
>>> print(str)
4
```

Osnovni ulaz (2)

○ Primeri – nastavak:

```
>>> a = str + 7
```

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

```
File "<pyshell#122>", line 1, in <module>
```

```
a = str + 7
```

```
TypeError: can only concatenate str (not "int") to str
```

```
>>> a = int(str) + 7
```

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

```
11
```

```
>>> str = input()
```

```
3.14
```

```
>>> a = int(str) + 7
```

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

```
File "<pyshell#126>", line 1, in <module>
```

```
a = int(str) + 7
```

```
ValueError: invalid literal for int() with base 10: '3.14'
```

Neophodna
konverzija

Pogrešna konverzija

Literatura – knjige

- M. Kovačević, Osnove programiranja u Pajtonu, Akademska misao, Beograd, 2017.
- M. Lutz, Learning python: Powerful object-oriented programming, 5th edition, O'Reilly Media, Inc., 2013.
- J. Zelle, Python Programming: An Introduction to Computer Science, 3rd Ed., Franklin, Beedle & Associates, 2016.
- D. Beazley, B. K. Jones, Python Cookbook, 3rd edition, O'Reilly Media, 2013.
- A. Downey, J. Elkner, C. Meyers, How To Think Like A Computer Scientist: Learning With Python, free e-book

Literatura – *online* izvori

- Python 3.8.0 documentation,
<https://docs.python.org/3/index.html>
- Colin Morris, 7-day Python course,
<https://www.kaggle.com/learn/python>
- Learn Python, Basic tutorial,
<https://www.learnpython.org/>
- TutorialsPoint, Python tutorial
<https://www.tutorialspoint.com/python/index.htm>