



Osnove programiranja u Pythonu

Teme

- Uvod u programski jezik Python
- Varijable i tipovi podataka
- Upravljanje greškama u programskom kôdu
- Kontrola toka izvršavanja programskog kôda
- Funkcije u Pythonu



Uvod u programski jezik Python

Sadržaj

- Povijest programskog jezika Python
- Razvojno okruženje
- Početak programiranja u Pythonu
 - Tradicionalni „Hello World” program
 - Ključne riječi u Pythonu

Povijest Pythona

- Razvijen krajem 80-ih prošlog stoljeća
 - Autor nizozemski programer Guido von Rossum
- Prvo put objavljen 1991. (verzija 0.9.0)
 - Tek 1994. objavljena verzija Python 1.0
- 2000. godine objavljena verzija Python 2.0
 - 2020. godine prestaje razvoj, unapređenje, popravak grešaka ... na verziji Python 2.0
 - I dalje dostupna i funkcionalna, ali preporuka je preći na verziju 3
- 2008. godine objavljena verzija Python 3.0

Monty Python's



Slika preuzeta s: <https://www.imdb.com/title/tt0063929/>

Zen of Python

- 1999 Tim Peters na Python mailing listu za unapređenje Python programskog jezika, poslao skup od 19 načela kako napisati dobar softver.
- Python Enhancement Proposals (PEP ili PEP lista):
 - *PEP 20 – The Zen of Python*
Petersova načela pod nazivom „Zen of Python” su, 2004. godine, uključena u službenu PEP listu po oznakom PEP 20
(<https://www.python.org/dev/peps/pep-0020/>)
 - **PEP 8 – Style Guide for Python Code**
(<https://www.python.org/dev/peps/pep-0008/>)

Zen of Python

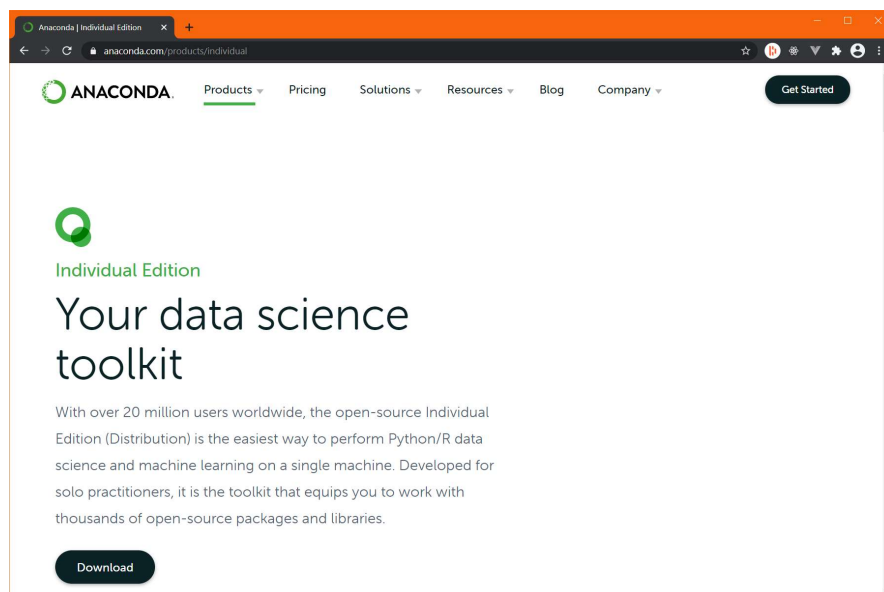
1. Beautiful is better than ugly.
2. Explicit is better than implicit.
3. Simple is better than complex.
4. Complex is better than complicated.
5. Flat is better than nested.
6. Sparse is better than dense.
7. Readability counts.
8. Special cases aren't special enough to break the rules.
9. Although practicality beats purity.
10. Errors should never pass silently.
11. Unless explicitly silenced.
12. In the face of ambiguity, refuse the temptation to guess.
13. There should be one, — and preferably only one — obvious way to do it.
14. Although that way may not be obvious at first unless you're Dutch.
15. Now is better than never.
16. Although never is often better than right now.
17. If the implementation is hard to explain, it's a bad idea.
18. If the implementation is easy to explain, it may be a good idea.
19. Namespaces are one honking great idea — let's do more of those!

Tko sve koristi Python?

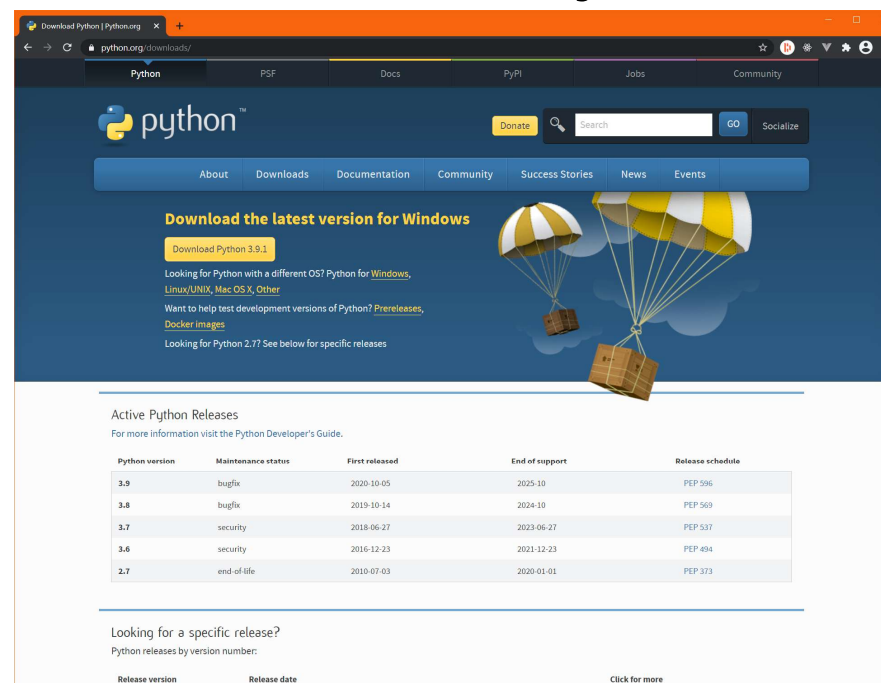
- Google tražilica, YouTube, Dropbox, Yahoo!, Walt Disney, Pixar, NASA, Red Hat, Nokia, IBM, Netflix, Yelp, Intel, Cisco, HP ...
- Igrice: Battlefield 2, Civilization IV, QuArK ...
- Python koriste za: sistemsko programiranje, web aplikacije, GUI aplikacije, igre, robotika, data science ...

Distribucije Pythona

Anaconda – Data Science

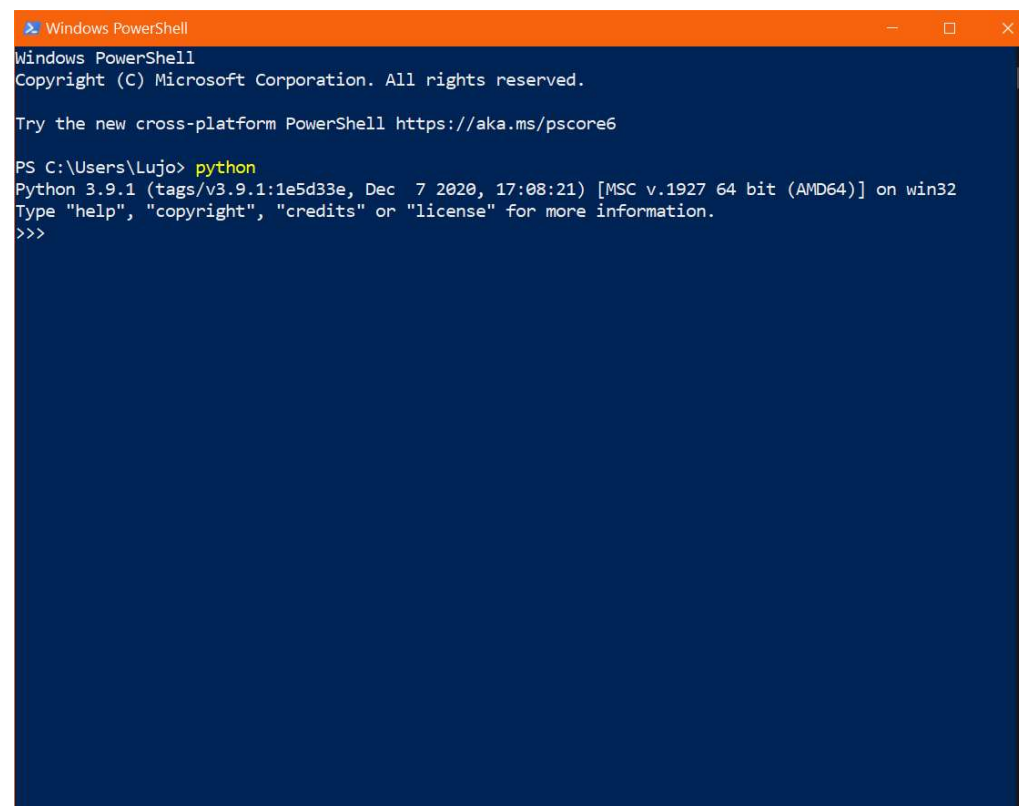


Samostalna distribucija



Python – Unutar PowerShell

- Pokrenuti PowerShell
- Upisati python i pritisnuti ENTER
- Dobit ćete tri znaka >>> na početku svakog reda. Tako znate da ste u Python načinu rada.
- Pokušajte upisati:
import this i pritisnuti enter
- Upišite help i pritisnite enter pa onda help(print)



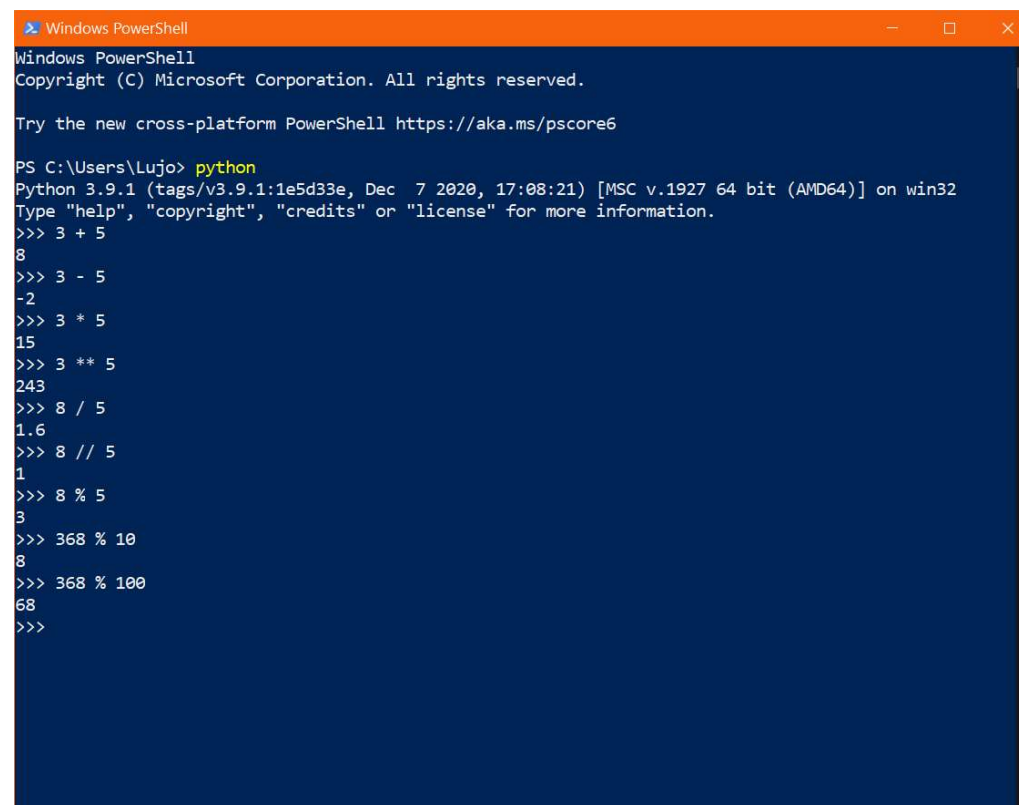
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Lujo> python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Python – Unutar PowerShell

- Zbrajanje – $3 + 5 = 8$
- Oduzimanje – $3 - 5 = -2$
- Množenje – $3 * 5 = 15$
- Potenciranje – $3 ** 5 = 243$
- Dijeljenje – $8 / 5 = 1.6$
- Cjelobrojno dijeljenje
 $8 // 5 = 1.6$
- *Modulo* (ostatak dijeljenja)
 $8 \% 5 = 3$



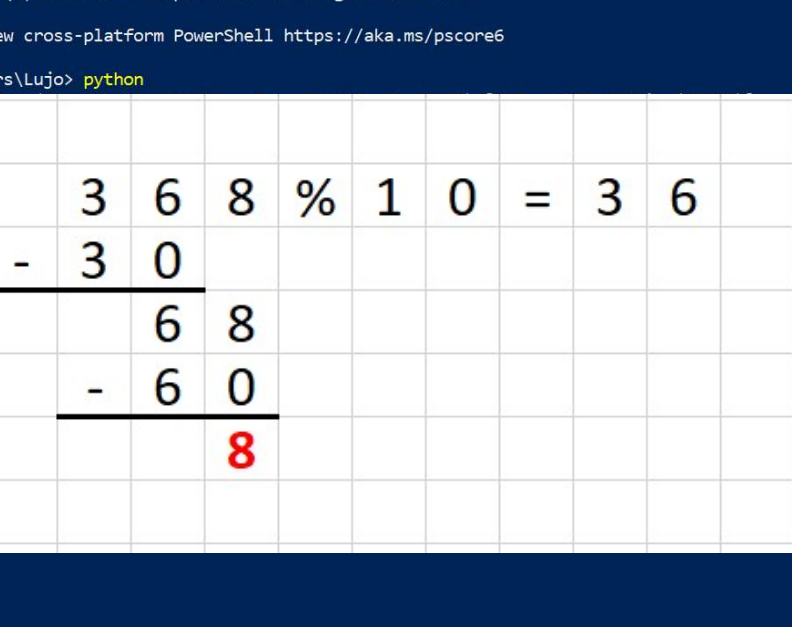
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Lujo> python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> 3 + 5
8
>>> 3 - 5
-2
>>> 3 * 5
15
>>> 3 ** 5
243
>>> 8 / 5
1.6
>>> 8 // 5
1
>>> 8 % 5
3
>>> 368 % 10
8
>>> 368 % 100
68
>>>
```

Modulo – ostatak od dijeljenja

- **$8 \% 5 = 3$** zato jer u 8 imamo samo jednu peticu i ostalo nam je 3 od one druge petice.
- **$368 \% 10 = 8$** zato jer u 368 imamo 36 desetica i ostalo je 8 od zadnje desetice.
Idealno za dobiti zadnju znamenku nekog broja.
- **$368 \% 100 = 68$** zato jer u 368 imamo 3 stotice i ostalo je 68 od one četvrte stotice.
Idealno za dobiti zadnje dvije znamenke nekog broja



Windows PowerShell

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS C:\Users\Lujo> python

```
368 % 10 = 36
- 30
-----
  68
- 60
-----
   8
```

in32

Aritmetički operatori – sažetak

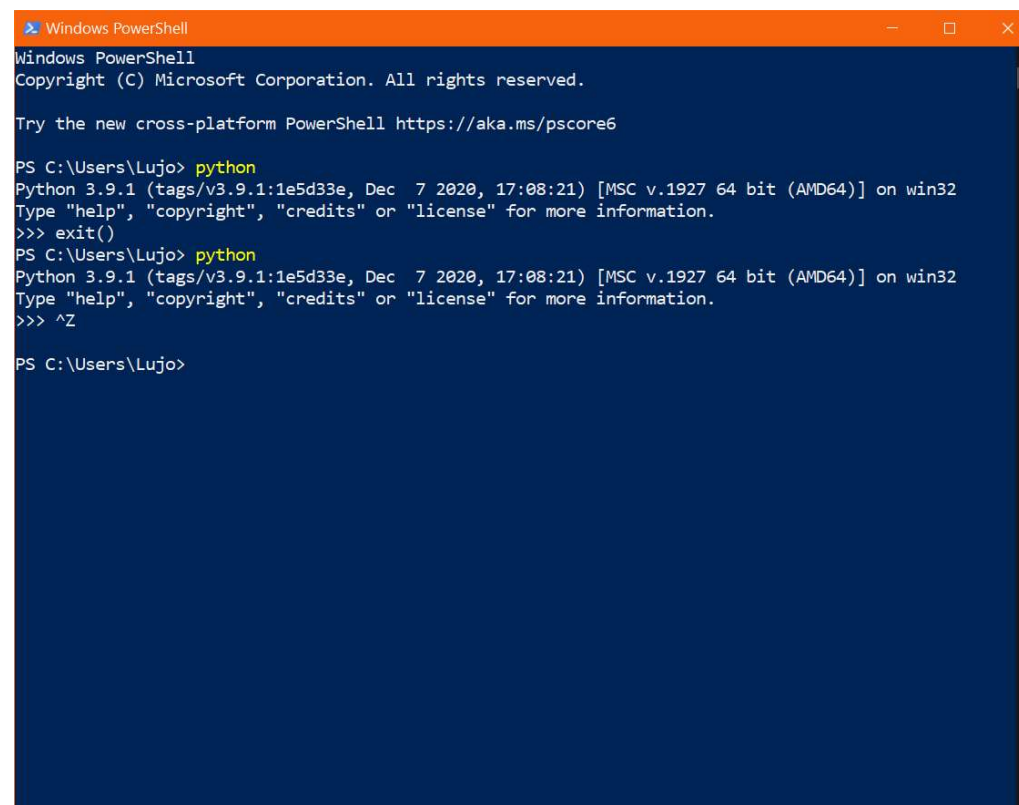
Operator	Namjena
+	Zbrajanje
-	Oduzimanje
*	Množenje
**	Potenciranje
/	Dijeljenje
//	Cjelobrojno dijeljenje
%	Modulo ostatak dijeljenja

Aritmetički operatori – skraćeni oblik

Operator	Primjer	Osnovni oblik
$+=$	$a += b$	$a = a + b$
$-=$	$a -= b$	$a = a - b$
$*=$	$a *= b$	$a = a * b$
$**=$	$a **= b$	$a = a ** b$
$/=$	$a /= b$	$a = a / b$
$//=$	$a //= b$	$a = a // b$
$\%=$	$a \% = b$	$a = a \% b$

Python – Izlaz iz PowerShell

- Izlazak iz Python command line načina rada:
 - `exit()`
 - `Ctrl + Z`
- Ili jednostavno zatvorite prozor PowerShella



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

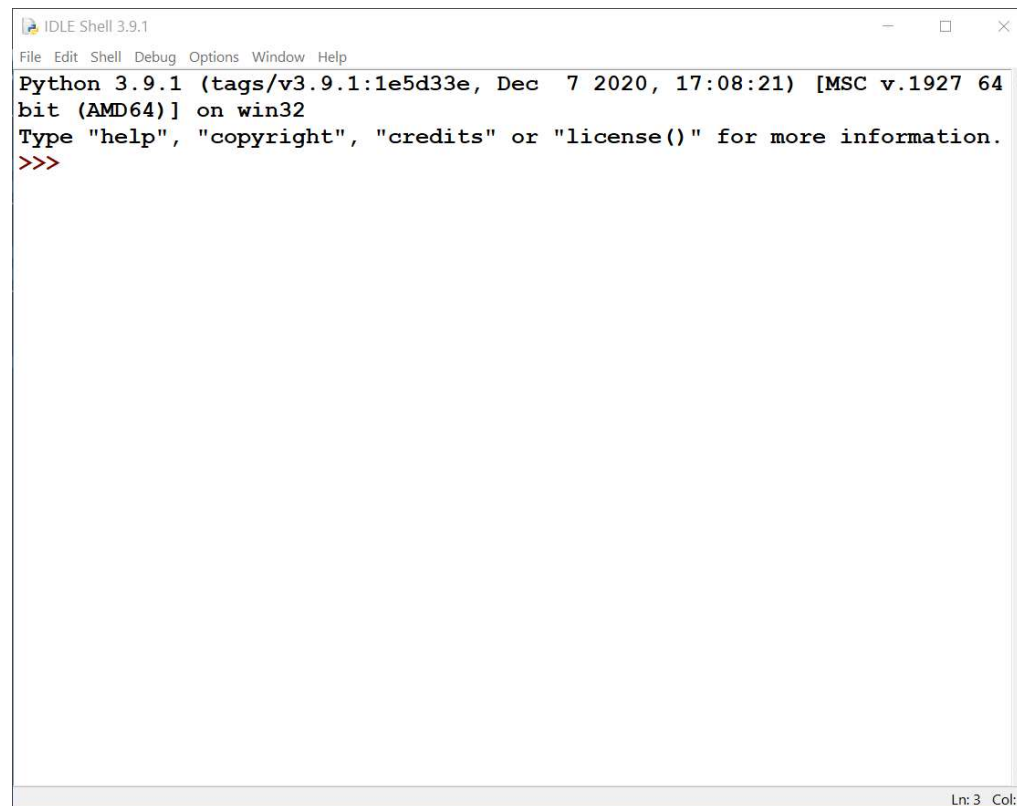
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Lujo> python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
PS C:\Users\Lujo> python
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> ^Z

PS C:\Users\Lujo>
```


IDLE Python

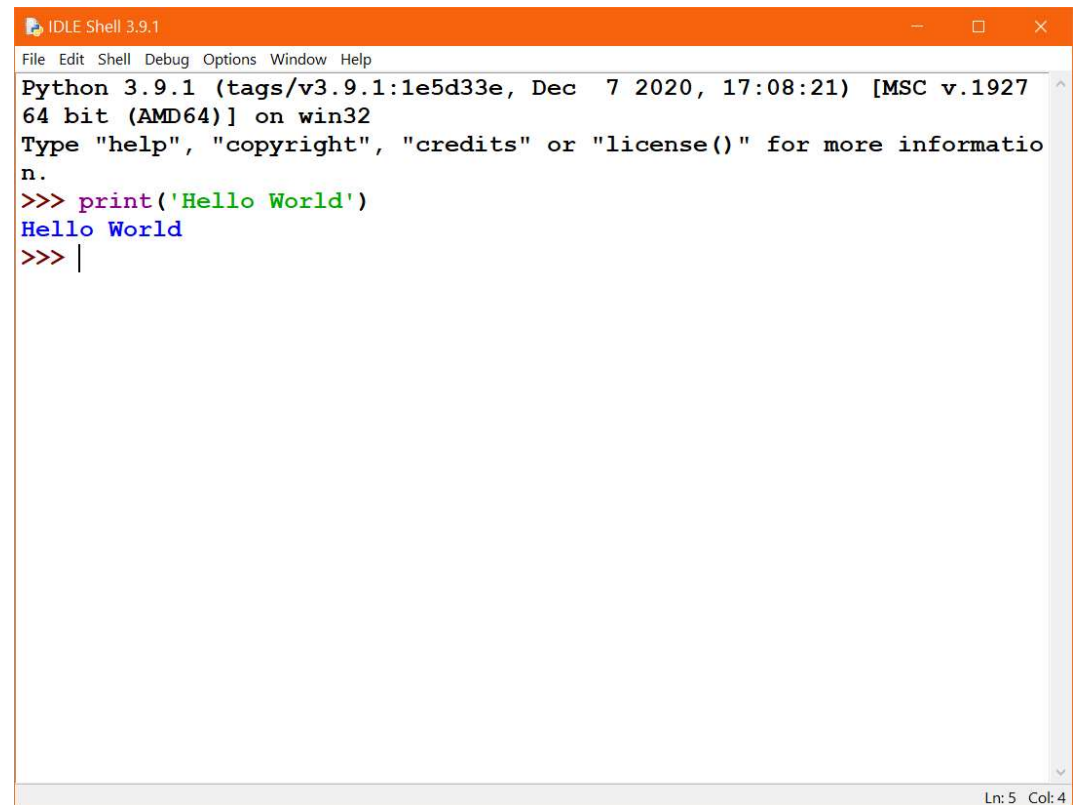
- IDE – Integrated Development Environment
- IDLE – Integrated Development Environment for Python
- Start Menu – IDLE
- Nešto naprednije sučelje za pisanje programskog kôda od PowerShell
- Dokumentacija:
<https://docs.python.org/3/library/idle.html>



The screenshot shows the IDLE Shell 3.9.1 window. The title bar reads "IDLE Shell 3.9.1". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following information: "Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32". Below this, it says "Type 'help', 'copyright', 'credits' or 'license()' for more information." followed by the prompt ">>>". The status bar at the bottom right indicates "Ln: 3 Col: 4".

“Hello World”

- Tradicionalno, prilikom učenja novog programskog jezika, prvi program ispisuje poruku „Hello World” na ekran.



```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
n.
>>> print('Hello World')
Hello World
>>> |
```

Ln: 5 Col: 4

„Hello World” – primjeri

C++

```
#include <iostream>
using namespace std;

int main(){
    cout << "Hello World" << endl;
    return 0;
}
```

Java

```
public class HelloWorld{
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Integrated Development Environment

- Integrated Development Environment – IDE
- Integrirani set alata nužnih za programiranje u nekom programskom jeziku.
 - Text editor za pisanje kôda
 - Kontrola verzija kôda
 - Debugging – pronalazak i otklanjanje grešaka
 - Compiler – prevođenje izvornog kôda u izvršni kôd (izrada .exe datoteka)
 - Ovisno o namjeni aplikacije koja se razvija, IDE omogućava pokretanje aplikacije za potrebe testiranja funkcionalnosti
 - ...

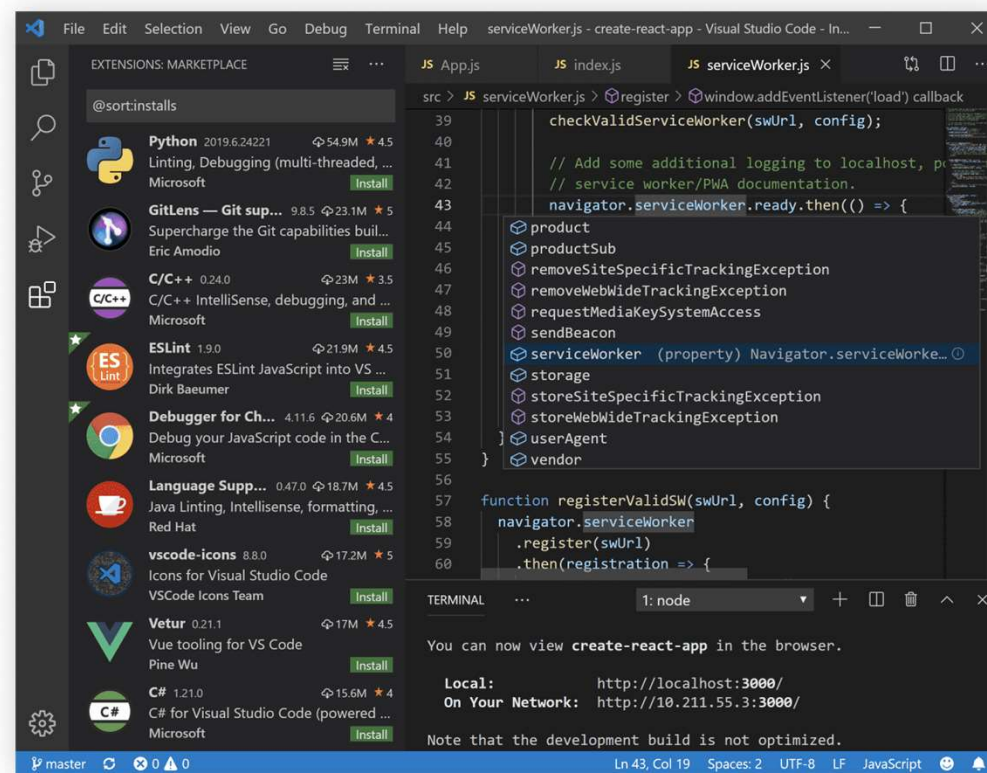
Besplatni IDE za Python

- Atom – (<https://atom.io/>)
- PyCharm Community – (<https://www.jetbrains.com/pycharm/>)
- Sublime Text 3 – (<https://www.sublimetext.com/>)
- Visual Studio Code – (<https://code.visualstudio.com/>)
 - Mi ćemo koristiti Visual Studio Code – VS Code
 - Nije isto kao i Microsoft Visual Studio

IDE alati su poredani abecednim redom

Visual Studio Code – VS Code

- Microsoft Visual Studio Code
- Dostupan na:
<https://code.visualstudio.com/>
- Dostupna verzija za sve verzije platformi
- Zahvaljujući ekstenzijama lako ga je prenamijeniti za gotovo bilo koji programski jezik
- Jako popularan za Python i JavaScript programske jezike



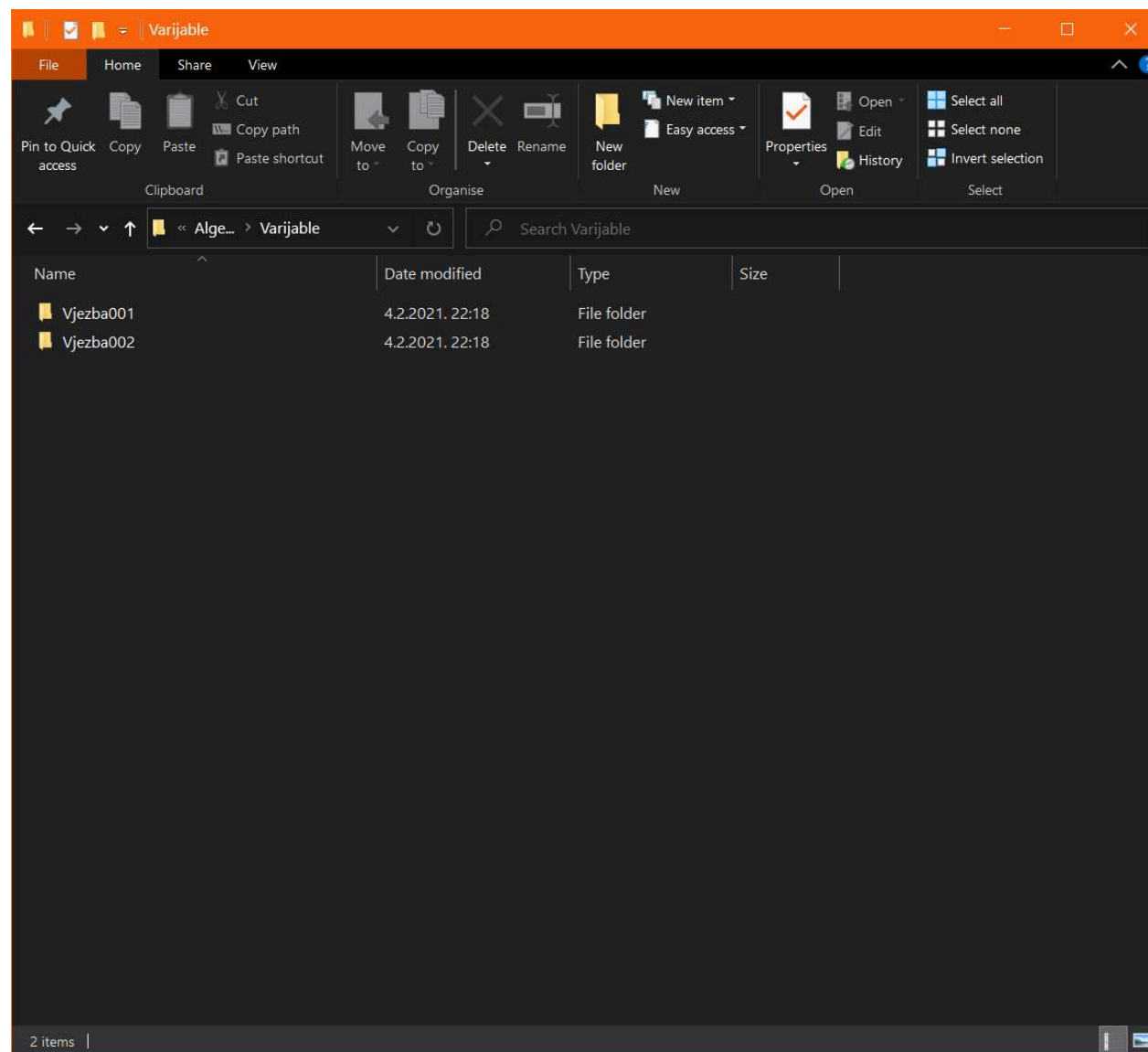
Visual Studio Code - Code Editing. Redefined

Kreiranje projekta u VS Code

- Projekt u VS Code predstavlja vršna mapa u kojoj su pohranjene sve datoteke bitne za pokretanje programa (uključujući i fotografije i sl.)

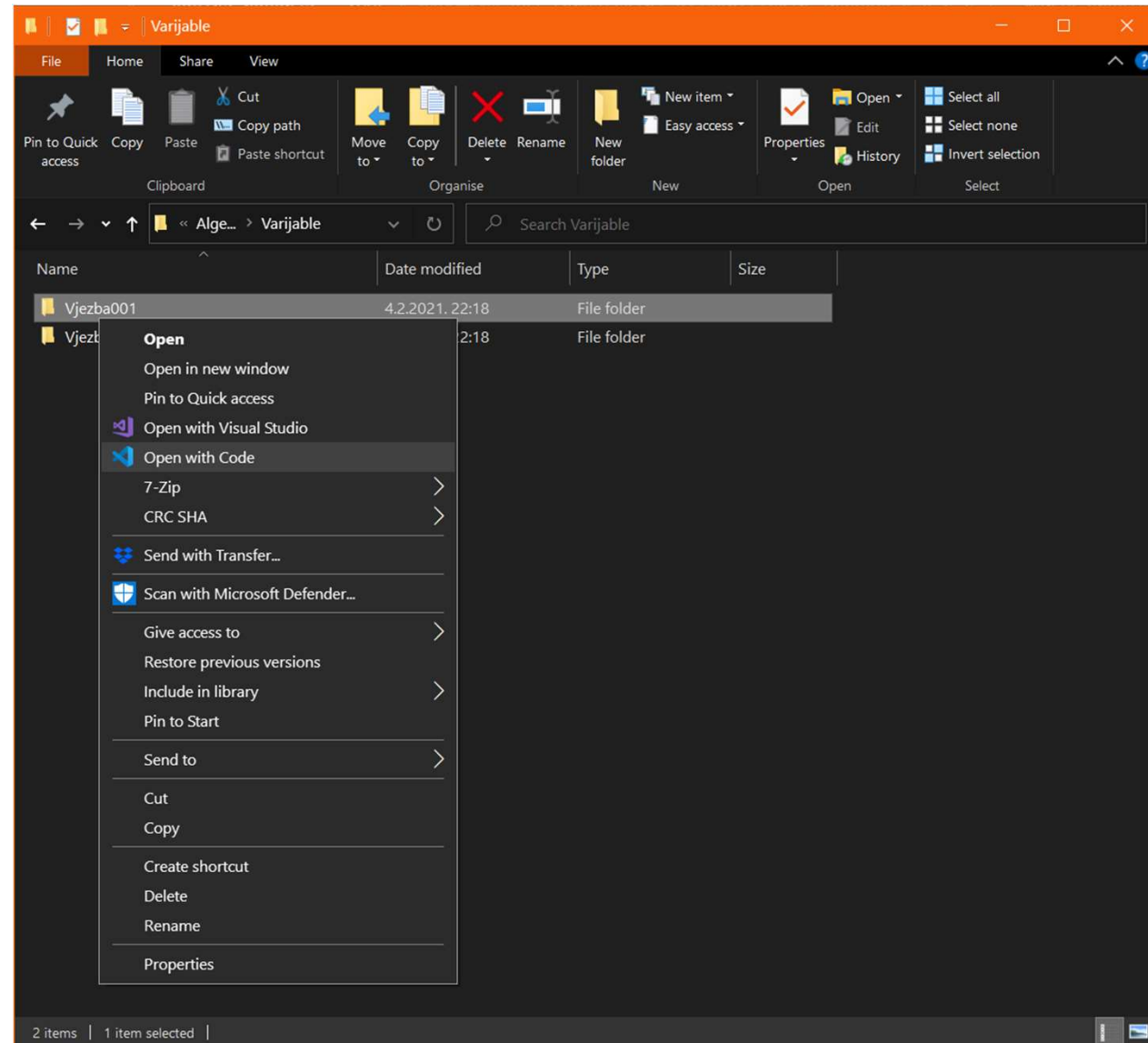
Kreiranje projekta u VS Code

- Kreirajte vršnu mapu u kojoj ćete imati sve projekte vezane uz program.
Primjer: D:\Algebra
- Unutar mape "Algebra" kreirajte jednu mapu koja će objedinjavati sve vježbe nekog poglavlja.
Primjer: D:\Algebra\Varijable
- I na kraju unutar mape poglavlja kreirajte mapu za svaku vježbu, zadatak ili primjer.
Primjer:
D:\Algebra\Varijable\Vjezba001



Kreiranje projekta u VS Code

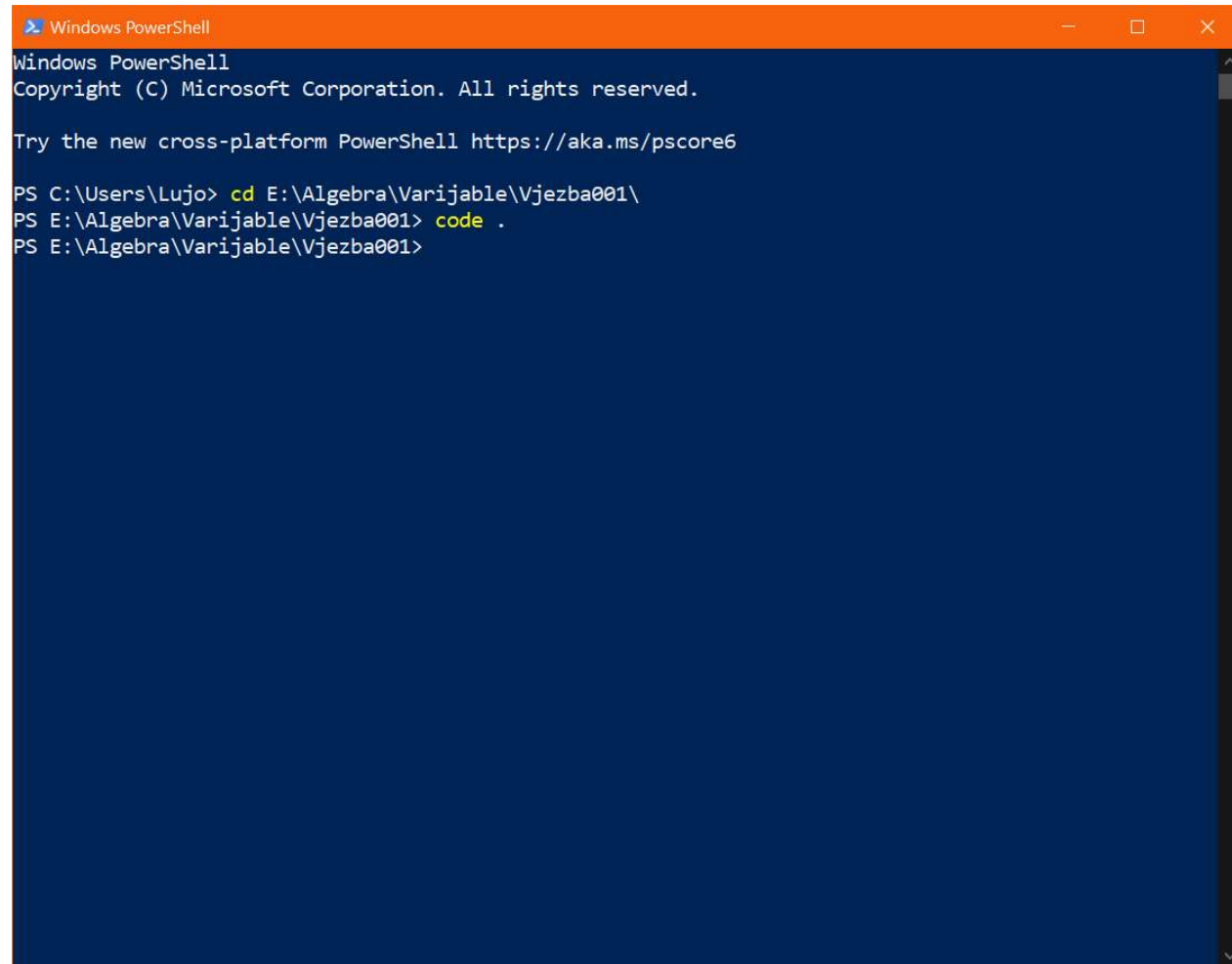
- Slijedeći korak je pokrenuti VS Code i dodijeliti
- VS Code možete pokrenuti tako da na mapu koju ste kreirali za čuvanje Python projekta, kliknete desnim gumbom i odaberete "*Open with Code*". Otvorit će se VS Code s već selektiranim folderom.



Kreiranje projekta u VS Code

- VS Code možete pokrenuti tako da se unutar PowerShell konzole pozicionirate na mapu koju ste kreirali za čuvanje Python projekta te upišete "code ." (code razmak točka).

Otvorit će se VS Code na pozicioniranom folderu.



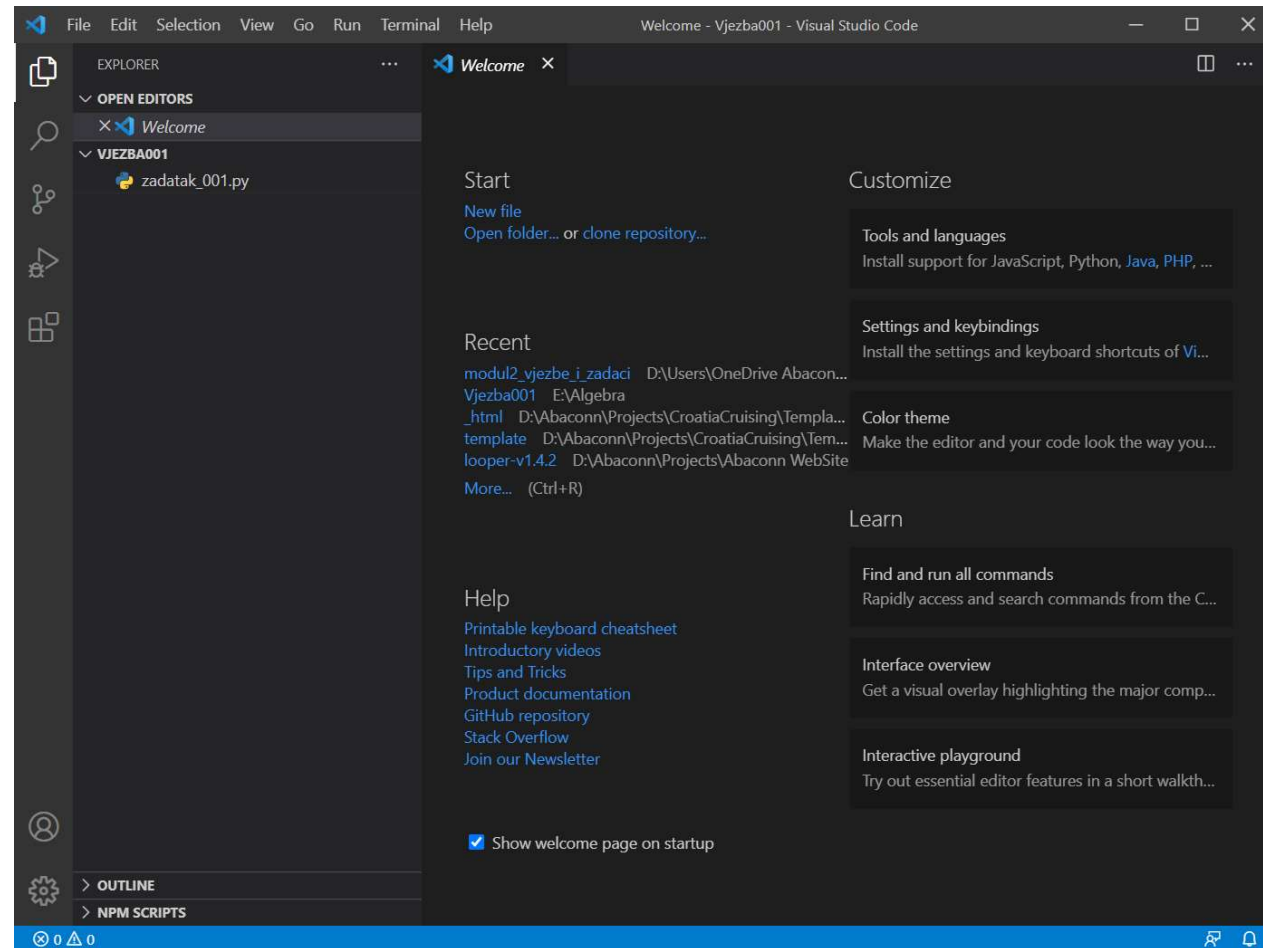
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Lujo> cd E:\Algebra\Variable\Vjezba001\
PS E:\Algebra\Variable\Vjezba001> code .
PS E:\Algebra\Variable\Vjezba001>
```

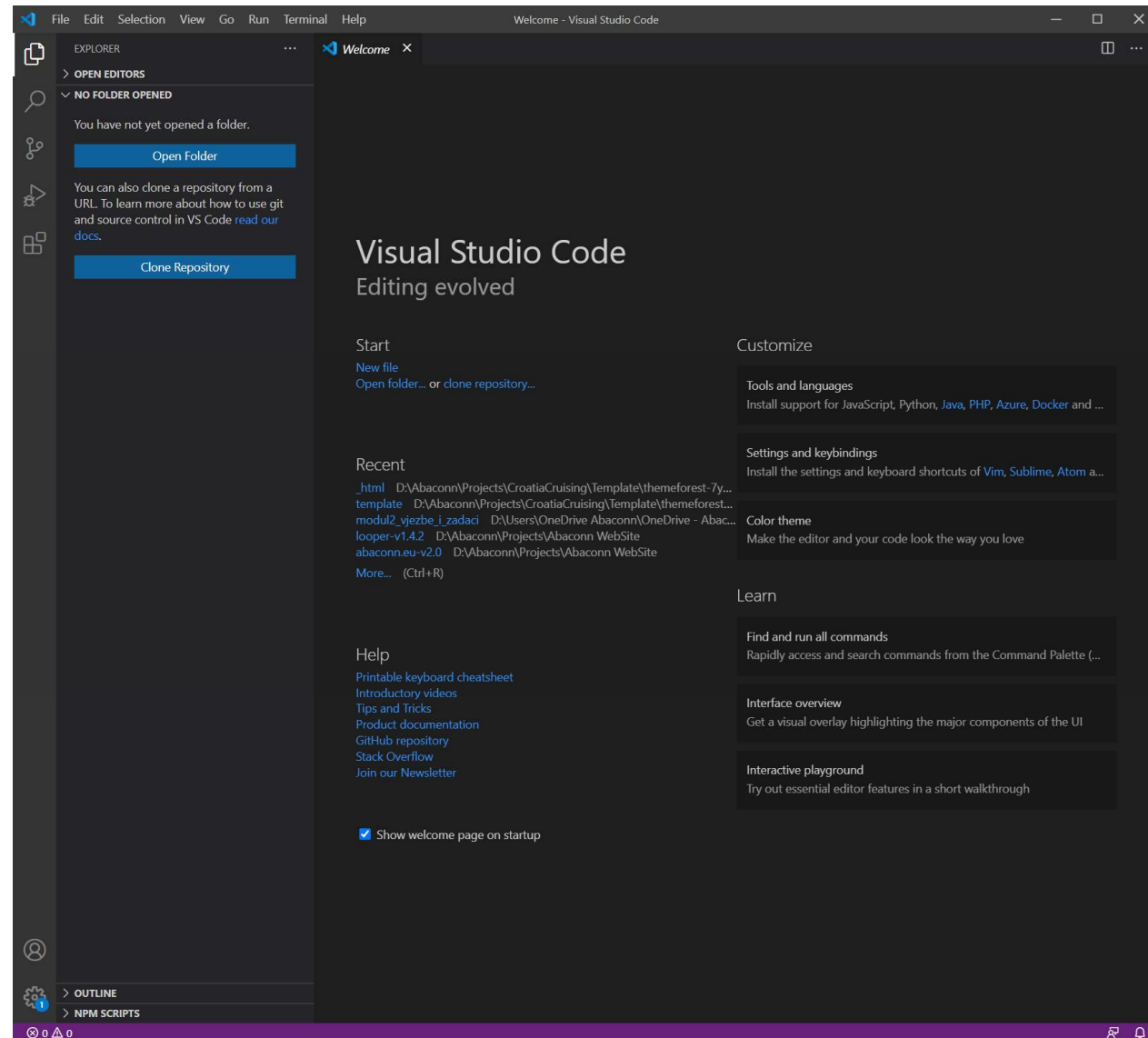
Kreiranje projekta u VS Code

- Lijevi dio VS Code prozora bit će otvoren na alatu za rad s folderima i datotekama. Ako već imate datoteke unutar foldera one će biti prikazane.
- VS Code prepoznaje ekstenzije datoteka pa će za .py datoteke pridružiti ikonu Pythona.
- Desni dio VS Code prozora prikazuje Welcome ekran koji možete zatvoriti klikom na X na vrhu kartice tog prozora ili ga možete zanemariti.
- Ako već imate datoteke unutar foldera, dvostruki klik na datoteku otvorit će datoteku u novoj kartici na desnoj strani prozora.



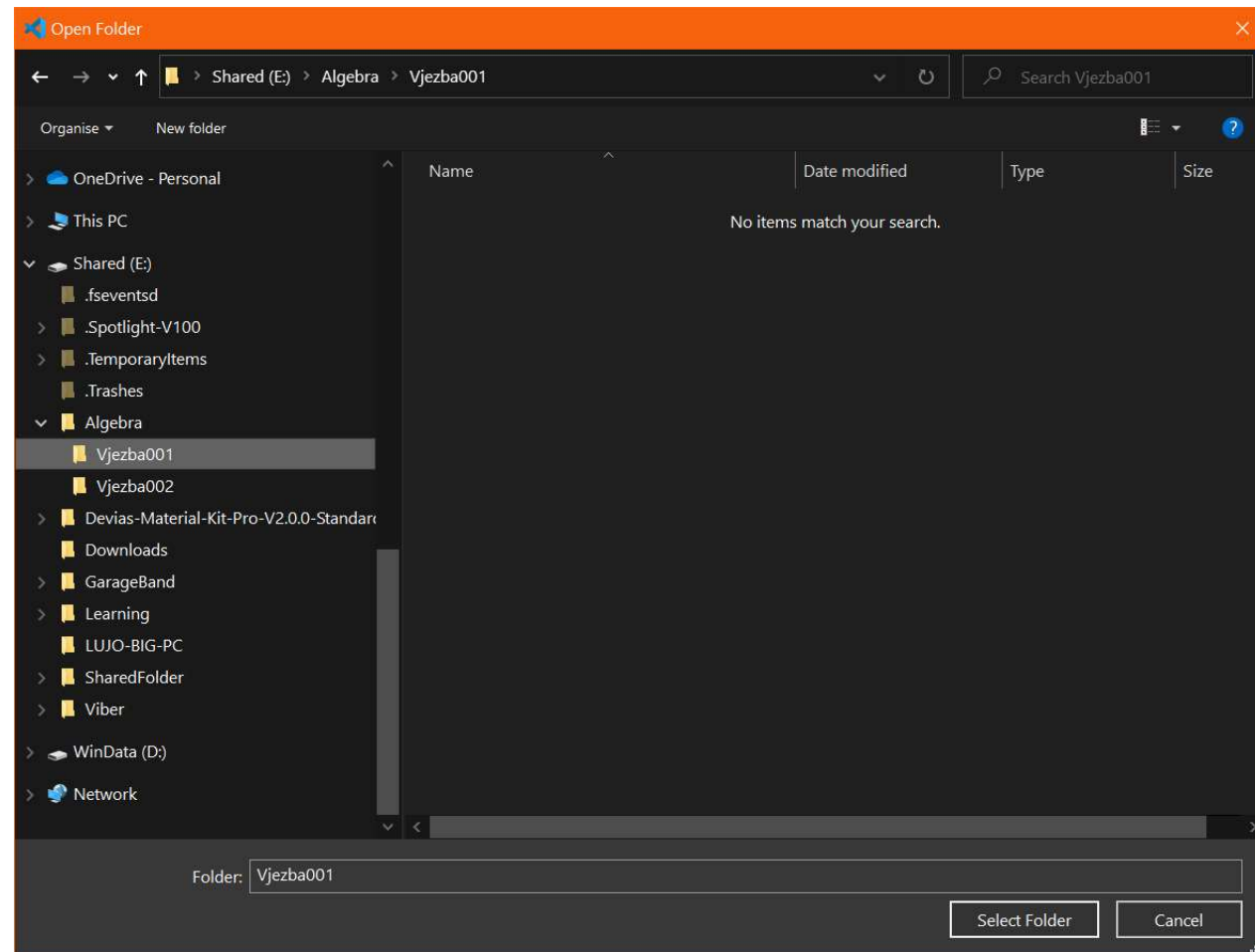
Kreiranje projekta u VS Code

- Projekt unutar VS Code možete kreirati i tako da pokrenete VS Code te odete u alat za rad s folderima i datotekama (prvi stupac lijevo, prva ikona).
- Unutar drugog stupca lijevo, kliknuti na plavi gumb "Open Folder"



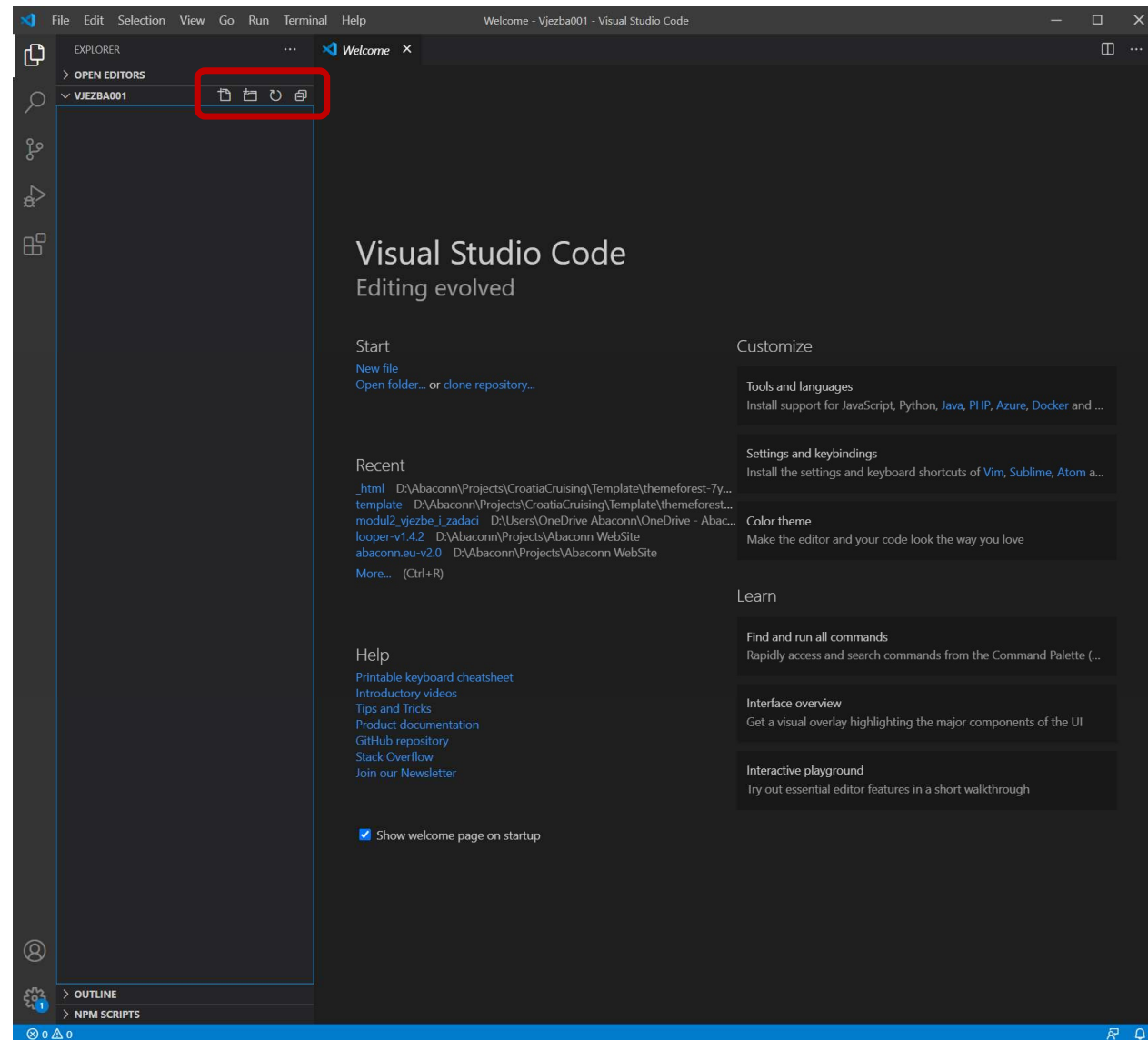
Kreiranje projekta u VS Code

- Odabrati folder koji želimo da bude vršni folder našeg projekta.
- Kliknuti na gumb "Select Folder"
- Rezultat će biti isti kao i kod prethodna dva načina kreiranja projekta unutar VS Code IDE programa.



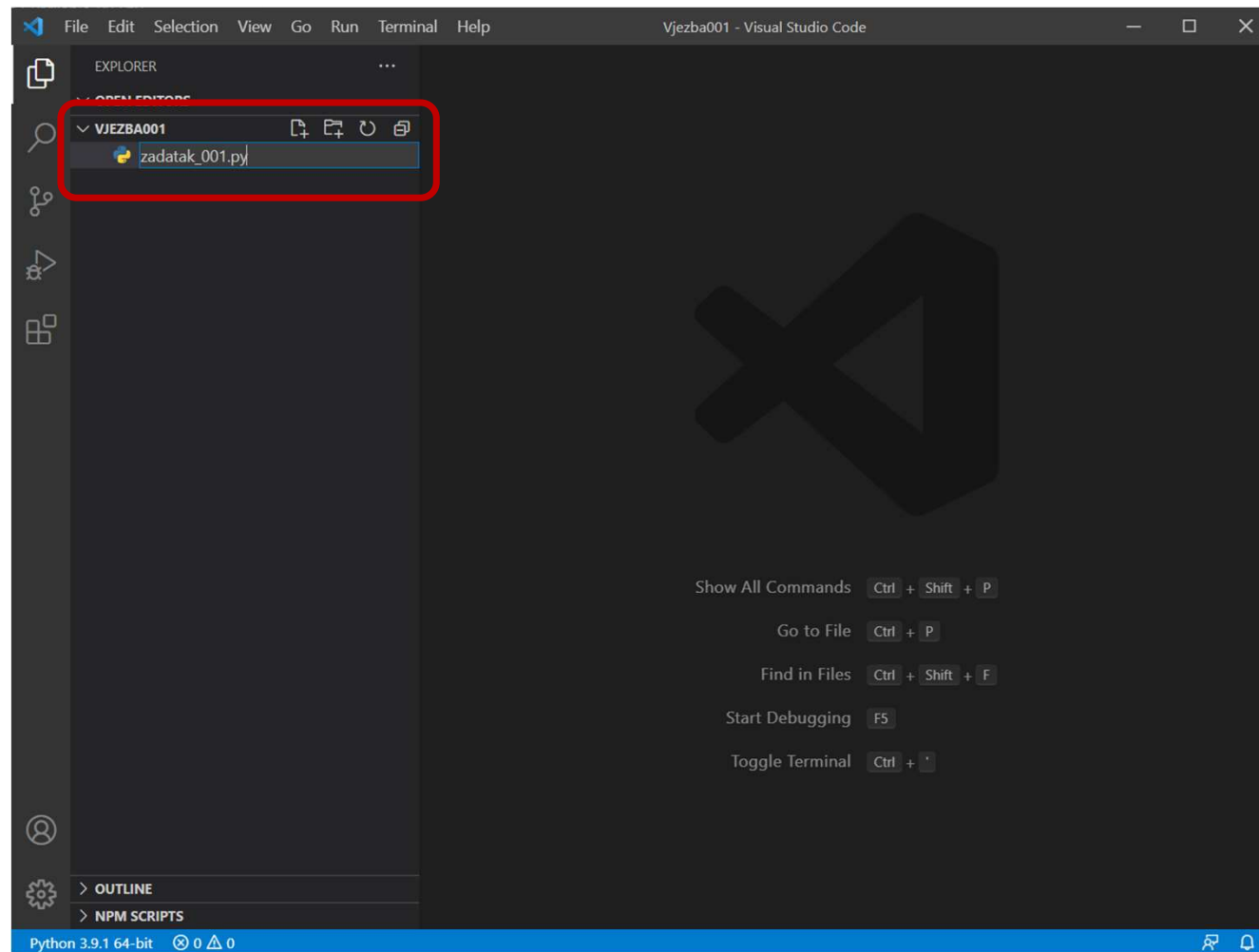
Kreiranje datoteke

- Za pisanje programskog kôda potrebno je kreirati odgovarajuću datoteku koja će sadržavati naš programski kôd.
- Unutar crvenog pravokutnika prikazane su ikone koje NISU uvijek vidljive. Ako pomaknete strelicu miša unutar ovog drugog lijevo stupca te ikone će se pojaviti.
- Prva ikona služi za kreiranje datoteka
- Druga za kreiranje pod-foldera (kasnije)
- Treća služi za osvježavanje prikaza. Ako ne vidimo neku datoteku koja se nalazi u folderu pa želimo osvježiti prikaz



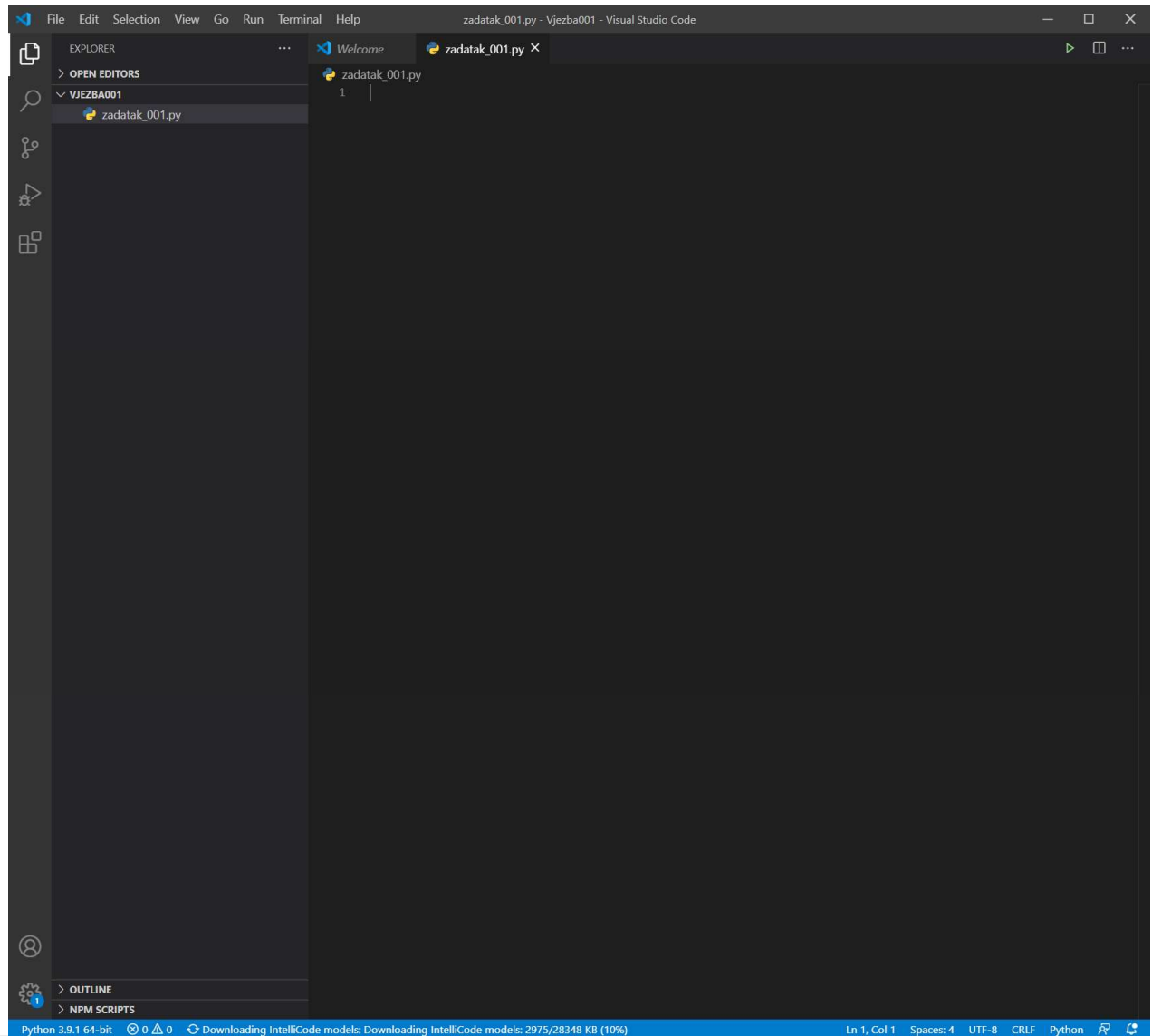
Kreiranje datoteke

- Klikom na prvu ikonu, pojavit će se okvir za unos naziva datoteke.
- Upišite naziv datoteke i NE ZABORAVITE dodati ekstenziju **.py**
- Ako zaboravite dodati ekstenziju VS Code će datoteku tretirati kao običnu tekstualnu datoteku i program se neće moći pokretati.
- Ako ste pogriješili prilikom unosa, jednostavno desnim gumbom kliknite na naziv datoteke i odaberite "Rename" te ponovite unos. Preimenovanje datoteke možete napraviti i ako selektirate datoteku i pritisnete tipku F2 na vašoj tikovnici



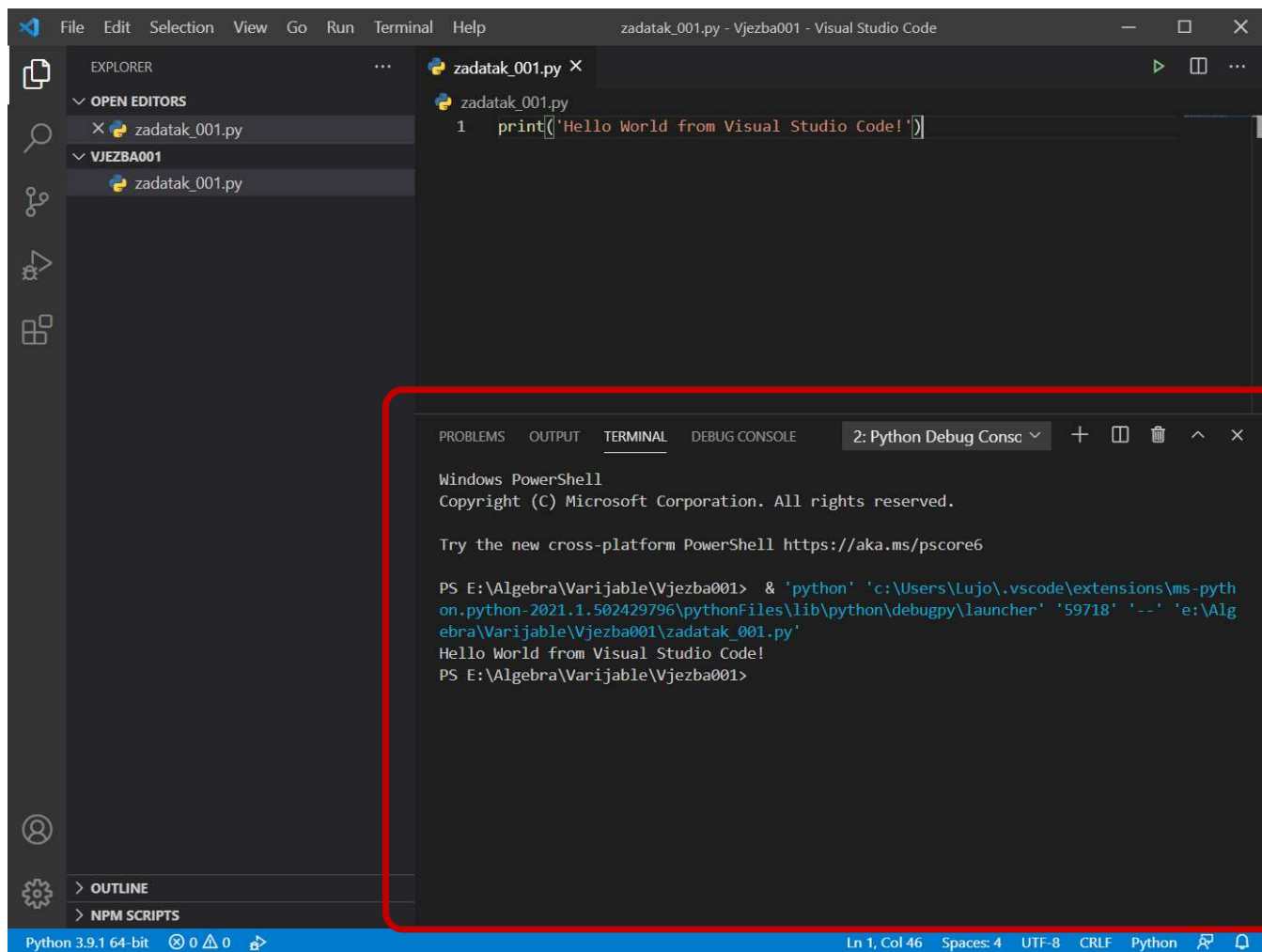
Kreiranje datoteke

- Nakon što ste unijeli naziv i ekstenziju datoteke te unos potvrdili ENTER tipkom, otvorit će se novi prozor, odnosno kartica sa sadržajem datoteke.
- Naša datoteka je prazna te je spremna za unos programskog kôda.
- Ovo možete ponoviti za svaku novu datoteku koju želite dodati unutar projekta. Jedan zadatak, neka bude jedna datoteka.



Pokretanje programa

- U prozor datoteke upišite: `print('Hello World from Visual Studio Code!')`
- Program pokrenite kombinacijom tipki CTRL+F5
- Ili klikom na izbornik Run -> Start Without Debugging
- Rezultat je prikazan u donjem dijelu prozora unutar crvenog pravokutnika



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'VJEZBA001' containing a file 'zadatak_001.py'. The main editor window shows the content of 'zadatak_001.py', which contains a single line of Python code: `1 print('Hello World from Visual Studio Code!')`. Below the editor, the 'TERMINAL' panel is active, showing the output of running the program. The terminal text includes the Windows PowerShell prompt, the command to run the Python file using the debugpy launcher, and the resulting output: 'Hello World from Visual Studio Code!'. A red rectangle highlights the terminal output area.

```
zadatak_001.py - Vjezba001 - Visual Studio Code

EXPLORER
  OPEN EDITORS
    zadatak_001.py
  VJEZBA001
    zadatak_001.py

zadatak_001.py
1 print('Hello World from Visual Studio Code!')
```

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 2: Python Debug Console
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\Algebra\Variable\Vjezba001> & 'python' 'c:\Users\Lujo\.vscode\extensions\ms-python.python-2021.1.502429796\pythonFiles\lib\python\debugpy\launcher' '59718' '--' 'e:\Algebra\Variable\Vjezba001\zadatak_001.py'
Hello World from Visual Studio Code!
PS E:\Algebra\Variable\Vjezba001>
```

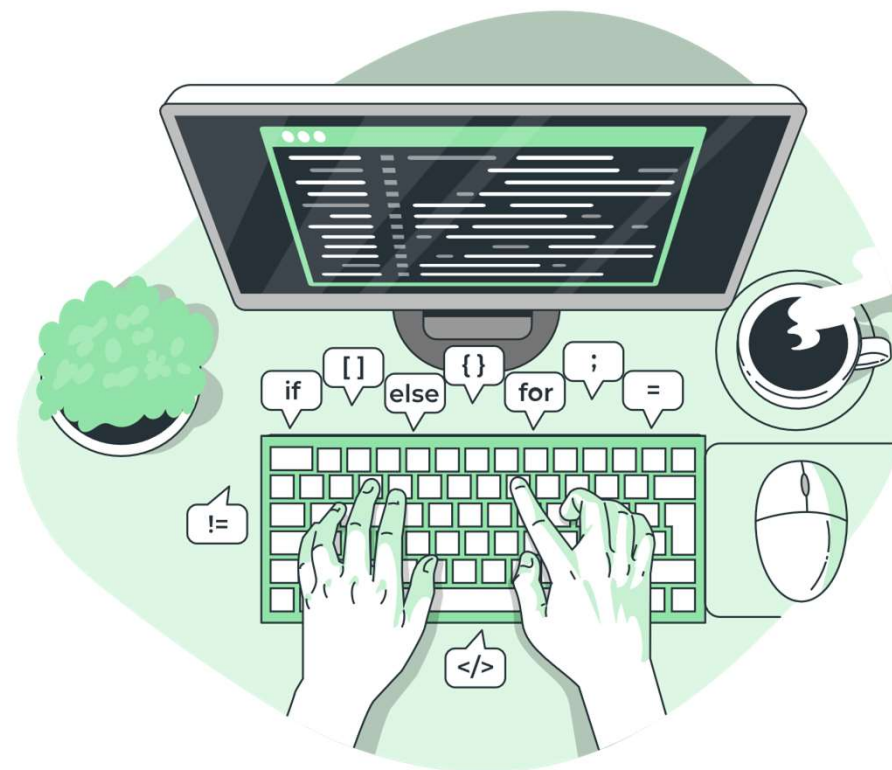
Python 3.9.1 64-bit 0 0 Python Ln 1, Col 46 Spaces: 4 UTF-8 CRLF

Varijable i tipovi podataka



Dobro je ponoviti

- **Programiranje** (engl. Computer Programming) je pisanje instrukcija koje računalo „razumije” kako bi slijedom tih instrukcija izvršilo željene aktivnosti.
- **Instrukcija** (engl. Instruction) predstavlja skup naredbi poredanih točno definiranim redoslijedom koje računalo „slijepo” slijedi. Ako je programer pogriješio u redoslijedu, računalo nije inteligentno da to prepozna i prilagodi redoslijed. Kao što je rečenica skup riječi poredanih određenim redoslijedom.
- **Naredba** (engl. Statement) je najmanji dio programskog jezika koja predstavlja određenu radnju koju treba izvršiti.
 - Primjer: `print(); input(); if ... else; while ...`



Varijable u programiranju

- Svaki program treba imati mogućnost „zapamtiti” jedan ili više podataka koji se koriste tijekom izvršavanja programa.
- Za pohranu tih podataka koriste se varijable.
- Varijabla predstavlja NAZIV lokacije unutar memorije računala u koju je pohranjen određeni podatak.
- Svaku varijablu čine minimalno:
 - Naziv varijable
 - Tip podatka pohranjenog u varijabli
 - Vrijednost pohranjena u varijabli
- Varijable mogu imati i konstantnu vrijednost (PI, g, c ...). Njih nazivamo konstantama.

Naziv varijable	Vrijednost varijable	Memorijska adresa
broj	15	0x0041F3
		0x0041F4
ime	Petar	0x0041F5
		0x0041F6
cijena	89.99	0x0041F7
valuta	kn	0x0041F8
		0x0041F9
		...

Varijable u Pythonu – naziv varijable

- Svaka varijabla mora imati naziv
- Pravila imenovanja varijabli ovise o programskom jeziku
- Neka pravila vrijede za sve programske jezike.
- Naziv varijable:
 - NE smije biti isti kao neka od ključnih riječi
 - NE smije početi brojkom
 - NE smije imati razmak
 - NE smije imati posebne znakove: !, ?, @, #, \$, %, ...

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue	global	raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Varijable u Pythonu – naziv varijable

- Preporuke za imenovanje varijabli.
- Naziv varijable bi trebao:
 - Opisivati podatak koji je pohranjen u varijabli
 - Imati znak „_” (podvlaka ili underscore) umjesto razmaka.
*Opcionalno bez razmaka tako da svaka riječ počinje velikim slovom *camelCase*
 - Imati sve znakove napisane malim slovima
 - Bez (hrvatskih) diakritičkih znakova
 - Zadržati dosljednost pa kako je varijabla dio programskog kôda, bolje je za naziv varijable koristiti engleski jezik, ali nikako NE miješati malo engleski, malo hrvatski ili neki drugi jezik.

Naziv varijable	Podatak koji čuva
\$x	'Patar Perić' – NE
ime_prezime	'Petar Perić' – DA
A	23,2° C – NE
unutarnja_temperatura	28,9° C – DA
vanjska_temperatura	23,2° C – DA
polumjer_kružnice	24.45 – NE
polumjerKruznice*	52 – DA

Varijable u Pythonu – naziv varijable

- Nazivi varijabli u Pythonu su Case Sensitive. Python razlikuje velika i mala slova u nazivu varijable
- Ime \neq ime \neq iMe
- PEP 8 – Style Guide for Python Code
(<https://www.python.org/dev/peps/pep-0008/#naming-conventions>)
- Dodatno: Google preporuke za Python pravila imenovanja:
<https://google.github.io/styleguide/pyguide.html#316-naming>

Varijable u Pythonu – Primjeri

TOČNO

```
ime = 'Petar'  
prezime = "Perić"  
godina_rođenja = 1995  
zaposlen = True  
tezina = 86.7  
spol = 'M'
```

POGREŠNO

```
1broj = 6  
ime i prezime = 'Petar Pe  
rić'  
break = 'Gablec'  
naziv_racuna_$ = 'Devizna  
štednja u dolarima'
```


Vježba – varijable

- Kreirajte varijable (imenujte ih i dodijelite im odgovarajuću vrijednost) za:
 - Ime, prezime, godinu rođenja, državu rođenja, status radnog odnosa, težinu te spol
 - Stranice a i b , četverokuta te za površinu tog četverokuta.

Komentari u kôdu

- Dio programskog kôda koji se NE izvršava
- Namjena za kratke opise unutar kôda
- Komentari u jednoj liniji počinju znakom "#"
- Komentari koji se protežu na više linija počinju i završavaju trostrukim navodnicima
""" Komentar ... """

Komentar unutar jedne linije koda
Komentar u drugoj liniji koda

"""

Ovo je komentar koji može biti u jednoj liniji koda, ali i nastaviti se u drugoj ili trećoj liniji koda

"""

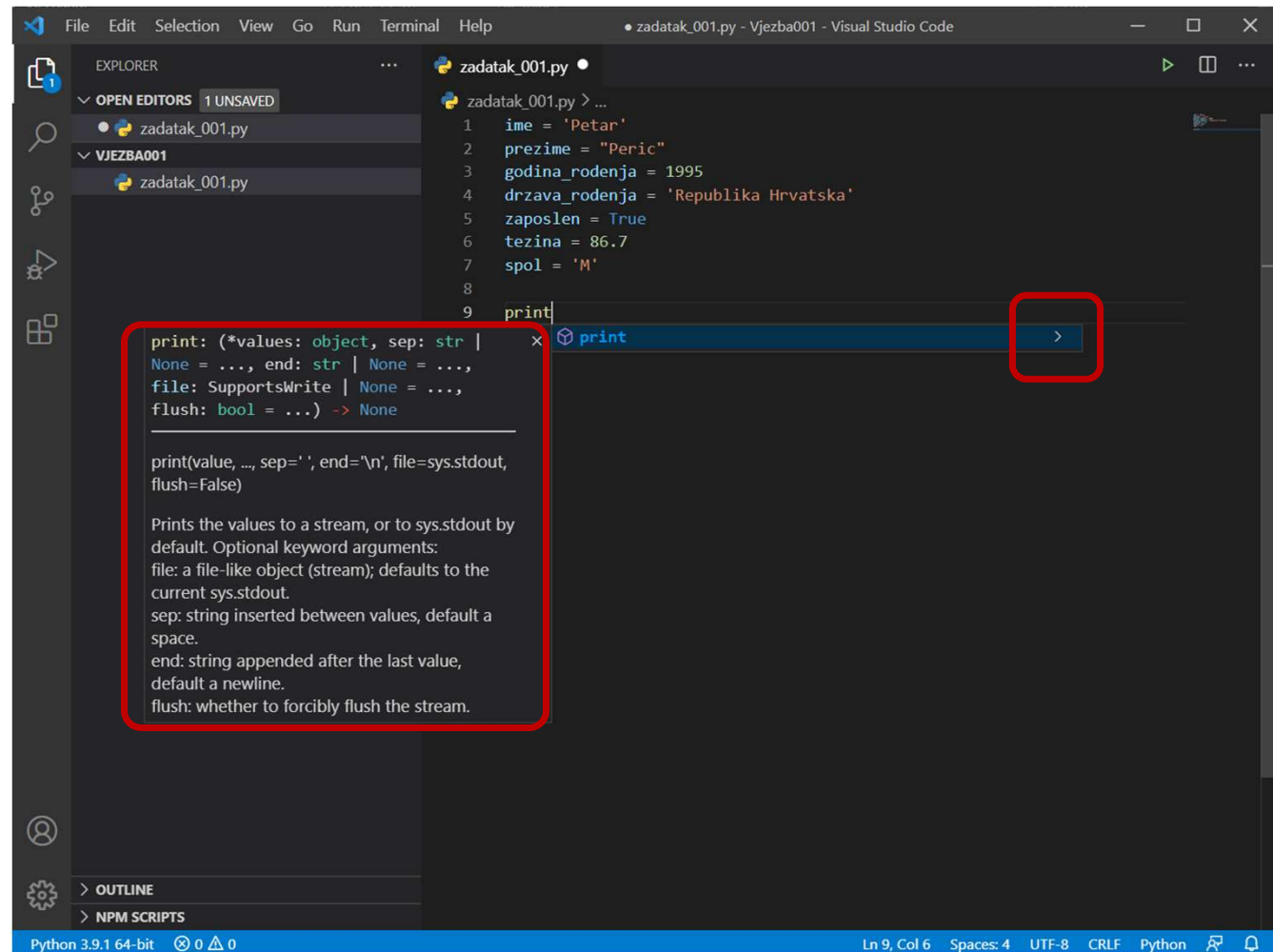
Komentari u kôdu – preporuka

- NE pretjerivati s komentarima
 - Ako je nešto očito u kôdu, nema potrebe za dodavanjem komentara
- Pišite *razumljiv* kôd tako da nema potrebe za komentarima.
- Lakše je napisati malo duži naziv varijable nego dodati komentar koji će opisati koja je namjena varijable.
- Koristiti ih za izradu mjesta u kôdu koje treba još završiti – TODO komentari
- *PEP 8 -- Style Guide for Python Code* (<https://www.python.org/dev/peps/pep-0008/#comments>)
- "Docstring" ili "Documentation Strgins" su komentari koji se koriste za izradu dokumentacije kôda. To su komentari koji počinju i završavaju s tri navodnika. *PEP 257 -- Docstring Conventiones* (<https://www.python.org/dev/peps/pep-0257/>)
- Odlična knjiga o programiranju općenito: Robert C. "Uncle Bob" Martin. 2008 *Clean Code: A Handbook of Agile Software Craftsmanship*, Pearson

Docstring help() *Intellisense*

Znak ">" kada kliknete otvorit će okvir s više detalja.

Ovaj okvir možete zatvoriti klikom na X u gornjem desnom dijelu okvira.



Tipovi podataka u programiranju

- Već smo spomenuli da varijable trebaju sadržavati tri informacije:
 - Naziv varijable ✓
 - Tip podatka pohranjenih u varijabli
 - Vrijednost
- Tipovi podataka
 - Osnovni ili primitivni tipovi podataka
 - Brojevi – cijeli i decimalni brojevi
 - Znakovi (čak i znakovi za brojke)
 - Boolean – točno/netočno ili True/False ili 1/0
 - Kompleksni
 - Tekst ili String – kolekcija znakova. Ponekad se i *string* ubraja u primitivne tipove podatka
 - Kolekcije podataka – liste, rječnici ... Ovdje može biti kolekcija znakova, ali i stringova i bilo kojih drugih tipova podataka
 - Korisnički definirani tipovi podataka – klase
- Postoji još i podjela na ugrađene i korisnički definirane tipove podatka

Ugrađeni tipovi podataka

- Primitivni tipovi podataka
 - Brojevi
 - Integer – Cijeli brojevi
 - Float – Decimalni brojevi
 - Znakovi (čak i brojke kao znak)
 - Boolean – točno ili netočno (True ili False)
- Složeni tipovi podataka
 - Tekst ili string – kolekcija znakova
 - Kolekcije podataka
 - lista, rječnik, n-terac ili tuple, set ...

Pixar filmovi			
Naslov (string)	Ocjena (float)	Broj pregleda (integer)	Favorit (Boolean)
Toy Story	9.78	159432	False
A Bug's Life	9.87	75365	False
Cars	9.79	45645	True
WALL-E	9.97	789654	True

Ispis vrijednosti varijable – print()

- Vrijednost varijabli često trebamo prikazati korisniku kako bi korisnik imao koristi od našeg programa
- Za prikaz vrijednosti varijable na ekran koristi se naredba **print()**

```
print('Hello World!')
```

```
>>> 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.

Vježba – varijable i ispis na ekran

- Kreirajte varijable (imenujte ih i dodijelite im odgovarajuću vrijednost) te ispišite na ekran odgovarajuće vrijednosti, za:
 - Ime, prezime, godinu rođenja, državu rođenja, status radnog odnosa, težinu te spol
 - Stranice a i b, četverokuta te za površinu tog četverokuta.
 - Izračun mjesečne potrošnje el. struje te cijene el. struje koju potroši mikrovalna pećnica snage 1,3 kW ako se koristi 2 sata dnevno?
 - Stranice trokuta, površinu trokuta ($P = \frac{a * v_a}{2}$, v_a je visina na stranicu a) te opseg trokuta.