

Prof. dr. sc. Marija Seder
Izv. prof. dr. sc. Jurica Babić
Doc. dr. sc. Đula Nađ
Prof. dr. sc. Vedran Podobnik
Ana Radović, univ. mag. ing. comp.
Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
Vojni studijski program: Vojno inženjerstvo
Vojno vođenje i upravljanje

Informatika

Uvod u Python I

Sadržaj

- ▶ Python
- ▶ Instalacija
- ▶ Python kao kalkulator
- ▶ Varijable
- ▶ Tipovi podataka
- ▶ Ispis na ekran
- ▶ Kornjačina grafika

Python

- ▶ Python osmislio Nizozemac Guido van Rossum
- ▶ Naziv po grupi i TV emisiji Monty Python
- ▶ koristit ćemo verziju Python 3.12.6 (zadnja verzija)
- ▶ <https://www.python.org/downloads/>
- ▶ razvojno okruženje za Python – IDLE (Integrated Development Environment, prezime komičara iz grupe Monty Python koji se zove Eric Idle)
- ▶ IDLE se instalira zajedno s Python

Odabрати Python verziju za instalaciju

Download the latest version for Windows

[Download Python 3.8.6](#)

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

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

Looking for Python 2.7? See below for specific releases



Active Python Releases

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

Python version	Maintenance status	First released	End of support	Release schedule
3.8	bugfix	2019-10		
3.7	security	2018-06		
3.6	security	2016-12		
3.5	security	2015-09		
2.7	end-of-life	2010-07		

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		ea132d6f449766623eee886966c7d41f	24377280	SIG
XZ compressed source tarball	Source release		69e73c49eeb1a853cefd26d18c9d069d	18233864	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	68170127a953e7f12465c1798f0965b8	30464376	SIG
Windows help file	Windows		4403f334f6c05175cc5edf03f9cde7b4	8531919	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	5f95c5a93e2d8a5b077f406bc4dd96e7	8177848	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	2acba3117582c5177cdd28b91bbe9ac9	28076528	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	c9d599d3880dfbc08f394e4b7526bb9b	1365864	SIG
Windows x86 embeddable zip file	Windows		7b287a90b33c2a9be55abc24a7febbb	7312114	SIG
Windows x86 executable installer	Windows		02cd63bd5b31e642fc3d5f07b3a4862a	26987416	SIG
Windows x86 web-based installer	Windows		acb0620aea46edc358dee0020078f228	1328200	SIG

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more
Python 3.8.6	Sept. 24, 2020	Download Release Notes
Python 3.5.10	Sept. 5, 2020	Download Release Notes

Python 3.8.6 (64-bit) Setup

Install Python 3.8.6 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Tomo\AppData\Local\Programs\Python\Python38

Includes IDLE, pip and documentation
Creates shortcuts and file associations

→ **Customize installation**
Choose location and features


Install launcher for all users (recommended)
 Add Python 3.8 to PATH

Cancel

Python 3.8.6 (64-bit) Setup

Setup Progress

Installing:
Python 3.8.6 Standard Library (64-bit)



Cancel

Python 3.8.6 (64-bit) Setup

Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

New to Python? Start with the [online tutorial](#) and [documentation](#).

See [what's new](#) in this release.

Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.

Close

Previous topic

Changelog

Next topic

1. Whetting Your Appetite

This Page

Report a Bug

Show Source

The Python Tutorial

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This tutorial introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.

For a description of standard objects and modules, see [The Python Standard Library](#). [The Python Language Reference](#) gives a more formal definition of the language. To write extensions in C or C++, read [Extending and Embedding the Python Interpreter](#) and [Python/C API Reference Manual](#). There are also several books covering Python in depth.

This tutorial does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python's most noteworthy features, and will give you a good idea of the language's flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [The Python Standard Library](#).

The [Glossary](#) is also worth going through.

- [1. Whetting Your Appetite](#)
- [2. Using the Python Interpreter](#)
 - [2.1. Invoking the Interpreter](#)
 - [2.1.1. Argument Passing](#)
 - [2.1.2. Interactive Mode](#)
 - [2.2. The Interpreter and Its Environment](#)
 - [2.2.1. Source Code Encoding](#)
- [3. An Informal Introduction to Python](#)
 - [3.1. Using Python as a Calculator](#)
 - [3.1.1. Numbers](#)
 - [3.1.2. Strings](#)

Download

Download these documents

Docs by version

- [Python 3.10 \(in development\)](#)
- [Python 3.9 \(pre-release\)](#)
- [Python 3.8 \(stable\)](#)
- [Python 3.7 \(security-fixes\)](#)
- [Python 3.6 \(security-fixes\)](#)
- [Python 3.5 \(security-fixes\)](#)
- [Python 2.7 \(EOL\)](#)
- [All versions](#)

Other resources

- [PEP Index](#)
- [Beginner's Guide](#)
- [Book List](#)
- [Audio/Visual Talks](#)
- [Python Developer's Guide](#)

Python 3.8.6 documentation

Welcome! This is the documentation for Python 3.8.6.

Parts of the documentation:

[What's new in Python 3.8?](#)

or all "What's new" documents since 2.0

[Tutorial](#)

start here

[Library Reference](#)

keep this under your pillow

[Language Reference](#)

describes syntax and language elements

[Python Setup and Usage](#)

how to use Python on different platforms

[Python HOWTOs](#)

in-depth documents on specific topics

Indices and tables:

[Global Module Index](#)

quick access to all modules

[General Index](#)

all functions, classes, terms

[Glossary](#)

the most important terms explained

Meta information:

[Reporting bugs](#)

[Contributing to Docs](#)

[Installing Python Modules](#)

installing from the Python Package Index & other sources

[Distributing Python Modules](#)

publishing modules for installation by others

[Extending and Embedding](#)

tutorial for C/C++ programmers

[Python/C API](#)

reference for C/C++ programmers

[FAQs](#)

frequently asked questions (with answers!)

[Search page](#)

search this documentation

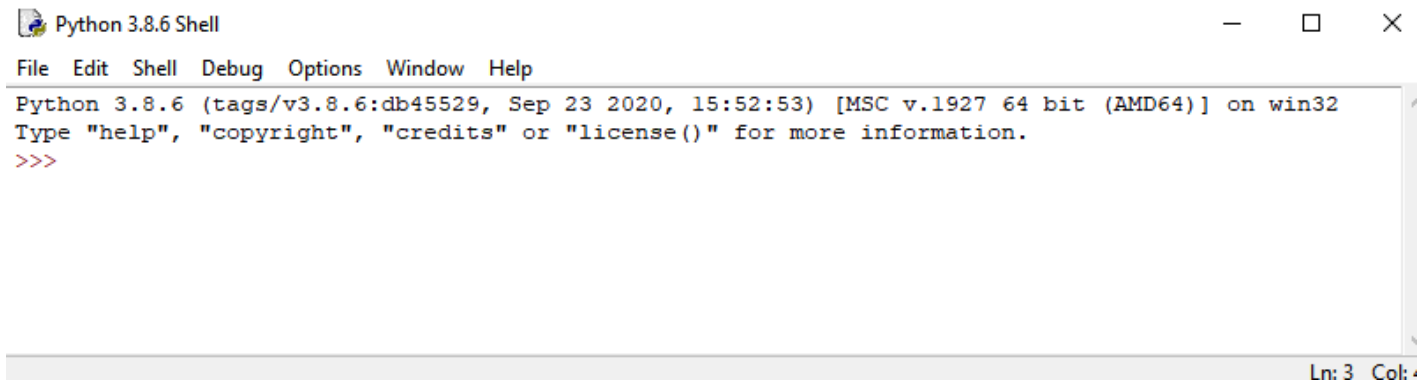
[Complete Table of Contents](#)

lists all sections and subsections

[History and License of Python](#)

IDLE

- ▶ Start → Python → IDLE



```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 3 Col: 1

- ▶ Python Shell je interaktivno sučelje s Pythonom

Python kao kalkulator

- ▶ Izračunaj: 5-7

```
>>> 5-7  
-2
```

- ▶ **Crvene** >>> čekaju da unesemo naredbu
- ▶ Naša naredba je crne boje
- ▶ Rezultat koji je odredio Python je **plave** boje

- ▶ Izračunaj: 35.7+21.4

```
>>> 35.7+21.4  
57.1
```

Ugrađene matematičke funkcije

+	zbrajanje	>>> 19+3 22
-	oduzimanje	>>> 19-3 16
*	množenje	>>> 19*3 57
**	potenciranje	>>> 19**3 6859
/	dijeljenje	>>> 19/3 6.333333333333333 >>> 19./3 6.333333333333333 >>> 19/3. 6.333333333333333
//	cjelobrojno dijeljenje	>>> 19//3 6 >>> 19.//3 6.0 >>> 19//3. 6.0
%	ostatak dijeljenja	>>> 19%3 1 >>> 19.%3 1.0 >>> 19%3.0 1.0

▶ Apsolutna vrijednost broja: `abs()`

```
>>> abs(.3)
0.3
>>> abs(-3.3)
3.3
```

▶ Količnik i ostatak pri dijeljenju: `divmod()`

▶ Izračunaj $7 : 3$

▶ $7 : 3 = 2$ i ostatak 1

```
>>> divmod(7,3)
(2, 1)
```

Tipovi podataka

- ▶ **Cijeli brojevi = integer = `int()`**
- ▶ Operacije s cijelim brojevima kao rezultat ne daju cijele brojeve ($19/3$) već decimalne (Python 2 vs Python 3)
- ▶ **Decimalni brojevi = `float()`**
- ▶ Decimalne znamenke se odvajaju točkom
- ▶ Operacije s decimalnim brojevima kao rezultat daju decimalne brojeve ($19.0/3.0$)
- ▶ Kombinirane operacije s cijelim i decimalnim brojevima kao rezultat daju decimalne brojeve ($19/3.$)

Red prvenstva matematičkih operacija

1. Potenciranje
2. Negacija
3. Množenje, dijeljenje, ostatak
4. Zbrajanje, oduzimanje

▶ Da ne bi došlo do nesporazuma – koristite **zagrade**

▶ Izračunajte: $3^2 + 16 : 4 - 3$

```
>>> 3 ** 2 + 16 / 4 - 3
10.0
>>> (3 ** 2 + 16) / 4 - 3
3.25
>>> 3 ** 2 + 16 / (4 - 3)
25.0
>>> 3 ** (2 + 16) / 4 - 3
96855119.25
```

Logički ili Booleov tip podataka

- ▶ Je li neka tvrdnja istinita ili lažna
- ▶ **Logički tip podataka = BOOL**
 - ▶ Ako je izraz istinit → `True`
 - ▶ Ako je izraz lažan → `False`

```
>>> 5 < 7
True
>>> 5 > 7
False
```



-
- ▶ U Pythonu su **bitna velika i mala slova**
 - ▶ Crvenom bojom su dani opisi grešaka koje se mogu pojaviti

```
>>> True
True
>>> False
False
>>> true

Traceback (most recent call last):
  File "<pyshell#28>", line 1, in <module>
    true
NameError: name 'true' is not defined
>>> false

Traceback (most recent call last):
  File "<pyshell#29>", line 1, in <module>
    false
NameError: name 'false' is not defined
```

- ▶ Vraćanje prošlog retka kako ne bismo morali sve pisati ponovno, a moramo ispraviti samo jedno slovo: **ALT + p**
- ▶ ili miš postaviti u redak koji želimo ponoviti i pritisnuti **ENTER**

Pretvaranje logičke u brojčanu vrijednost

- ▶ Često se umjesto `True` koristi broj 1
- ▶ A umjesto `False` broj 0
- ▶ Pretvaranje logičke u brojčanu vrijednost:

`int()`

`float()`

```
>>> int(True)
1
>>> int(False)
0
>>> float(True)
1.0
>>> float(False)
0.0
```

- ▶ Pretvaranje brojčane u logičku vrijednost:

`bool()`

```
>>> bool(0)
False
>>> bool(1)
True
>>> bool(-4.3)
True
```

0	False
bilo koji broj osim nule	True

Matematički trik

- ▶ Zamisli dva jednoznamenkasta broja. Prvi broj pomnoži s 5, umnošku dodaj 3, te zbroj pomnoži s 2. Broju koji si dobio dodaj drugi broj koji si zamislio. Reci mi rezultat i reći ću ti koje si brojeve zamislio.
- ▶ Dobio sam 60

```
>>> 5 * 5
25
>>> 25 + 3
28
>>> 28 * 2
56
>>> 56 + 4
60
```

```
>>> 5 * 5 + 3
28
>>> 28 * 2
56
>>> 56 + 4
60
```

- ▶ Zamislio si: prvi broj je 5, drugi broj je 4

Matematički trik

► Zašto? Matematika:

x – prvi broj, y – drugi broj

$$(x * 5 + 3) * 2 + y = 10x + y + 6$$

```
>>> 5 * 5
25
>>> 25 + 3
28
>>> 28 * 2
56
>>> 56 + 4
60
```

```
>>> 5 * 5 + 3
28
>>> 28 * 2
56
>>> 56 + 4
60
```

```
>>> 5 * 5 + 3
28
>>> _ * 2
56
>>> _ + 4
60
```

```
>>> x = 5
>>> y = 4
>>> (x * 5 + 3) * 2 + y
60
```

→ Varijable

↓
Posljednja izračunata vrijednost

Variable

- ▶ Variable su imena koja su pridružena pojedinim vrijednostima
- ▶ Ako istu vrijednost ponavljamo više puta – samo navedemo ime njene varijable
- ▶ Imenu varijable pridružujemo vrijednosti koristeći znak pridruživanja =
- ▶ Rezultat pridruživanja se ne ispisuje odmah na ekranu
- ▶ Vrijednost varijable je moguće ispisati naredbom `print()`

-
- ▶ Što će ispisati sljedeći program?

```
>>> a = 15
>>> b = a + 7
>>> c = b / 2
>>> print(c)
```

- ▶ Naredba `divmod()` kao rješenje daje dva broja → svaki od tih brojeva možemo spremiti u svoju varijablu

```
>>> a, b = divmod(17, 5)
>>> a, b
(3, 2)
```

- ▶ Odredi vrijednost varijable `k`

```
>>> c, d = divmod(26, 3)
>>> k = c / d
```

Znakovni niz – string - str

- ▶ Znakovni niz je tip podataka za pohranjivanje teksta
- ▶ Obilježava se jednostrukim ili dvostrukim navodnicima

```
>>> 'Primjer teksta'  
'Primjer teksta'  
>>> 'Ivan vikne: "Stoj!"'  
'Ivan vikne: "Stoj!"'  
>>> "Ivan vikne: 'Stoj!'"  
"Ivan vikne: 'Stoj!'"
```

- ▶ Ako ne želimo ispisati navodnike, koristimo naredbu `print()`

```
>>> print('Primjer teksta')  
Primjer teksta  
>>> print('Ivan vikne: "Stoj!"')  
Ivan vikne: "Stoj!"  
>>> print("Ivan vikne: 'Stoj!'")  
Ivan vikne: 'Stoj!'
```

Specijalni znakovi pri ispisu

Znak	Opis
<code>\n</code>	Prijelaz u novi redak
<code>\t</code>	Tabulator – razmak između riječi
<code>\\</code>	Ispis lijevo ukošene crte
<code>\'</code>	Ispis jednostrukog navodnika
<code>\"</code>	Ispis dvostrukog navodnika

- ▶ Napiši program koji će na ekran pregledno ispisati:

Ime	Godine
Marko	19
Ivan	20
Maja	19

-
- Napiši program koji će na ekran pregledno ispisati:

Ime	Godine
Marko	19
Ivan	20
Maja	19

```
>>> print('Ime\tGodine\nMarko\t19\nIvan\t20\nMaja\t19')  
Ime      Godine  
Marko    19  
Ivan     20  
Maja     19
```

Znak	Opis
\n	Prijelaz u novi redak
\t	Tabulator – razmak između riječi

Tri navodnika

- ▶ Koristimo kada želimo pisati točno onako kako želimo vidjeti ispis na ekranu

```
>>> print('Ime      Godine
Marko    19
Ivan     20
Maja     19')
Ime      Godine
Marko    19
Ivan     20
Maja     19
```

- ▶ Bez naredbe `print()` rezultat je suprotni

```
>>> '''Ime      Godine
Marko    19
Ivan     20
Maja     19'''
'Ime\tGodine\nMarko\t19\nIvan\t20\nMaja\t19'
```


ASCII

- ▶ Pri pohrani znaka u računalo – svaki znak se prvo pretvara u broj – kod
- ▶ American Standard Code for Information Interchange – ASCII
- ▶ omogućuje pohranu 128 znaka (1 znak = 7 bitova)
- ▶ ALT + 3 broja

Regular ASCII Chart (character codes 0 - 127)															
000	(nul)	016	▶ (dle)	032	sp	048	0	064	@	080	P	096	'	112	p
001	☉ (soh)	017	◀ (dc1)	033	!	049	1	065	A	081	Q	097	a	113	q
002	☉ (stx)	018	↓ (dc2)	034	"	050	2	066	B	082	R	098	b	114	r
003	♥ (etx)	019	!! (dc3)	035	#	051	3	067	C	083	S	099	c	115	s
004	♦ (eot)	020	¶ (dc4)	036	\$	052	4	068	D	084	T	100	d	116	t
005	⚡ (enq)	021	§ (nak)	037	%	053	5	069	E	085	U	101	e	117	u
006	♣ (ack)	022	⎓ (syn)	038	&	054	6	070	F	086	V	102	f	118	v
007	• (bel)	023	␣ (etb)	039	'	055	7	071	G	087	W	103	g	119	w
008	■ (bs)	024	↑ (can)	040	(056	8	072	H	088	X	104	h	120	x
009	(tab)	025	↓ (em)	041)	057	9	073	I	089	Y	105	i	121	y
010	(lf)	026	(eof)	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂ (vt)	027	← (esc)	043	+	059	;	075	K	091	[107	k	123	{
012	♀ (np)	028	- (fs)	044	,	060	<	076	L	092	\	108	l	124	
013	(cr)	029	⊕ (gs)	045	-	061	=	077	M	093]	109	m	125	}
014	♂ (so)	030	▲ (rs)	046	.	062	>	078	N	094	^	110	n	126	~
015	* (si)	031	▼ (us)	047	/	063	?	079	O	095	_	111	o	127	␣

8-bitna ASCII tablica

- ▶ Prošireni ASCII kod
- ▶ Pohrana 256 znakova
- ▶ 1 znak = 8 bitova = 1 bajt
- ▶ Razni jezici – razne varijante
- ▶ Npr. isti kodni broj u SAD predstavljao znak za dolar, a u Velikoj Britaniji znak za funtu

0	32	64	@	96	'	128	160	192	À	224	à
1	33	65	A	97	a	129	161	193	Á	225	á
2	34	66	B	98	b	130	162	194	Â	226	â
3	35	67	C	99	c	131	163	195	Ã	227	ã
4	36	68	D	100	d	132	164	196	Ä	228	ä
5	37	69	E	101	e	133	165	197	Å	229	å
6	38	70	F	102	f	134	166	198	Æ	230	æ
7	39	71	G	103	g	135	167	199	Ç	231	ç
8	40	72	H	104	h	136	168	200	È	232	è
9	41	73	I	105	i	137	169	201	É	233	é
10	42	74	J	106	j	138	170	202	Ê	234	ê
11	43	75	K	107	k	139	171	203	Ë	235	ë
12	44	76	L	108	l	140	172	204	Ì	236	ì
13	45	77	M	109	m	141	173	205	Í	237	í
14	46	78	N	110	n	142	174	206	Î	238	î
15	47	79	O	111	o	143	175	207	Ï	239	ï
16	48	80	P	112	p	144	176	208	Ð	240	ð
17	49	81	Q	113	q	145	177	209	Ñ	241	ñ
18	50	82	R	114	r	146	178	210	Ò	242	ò
19	51	83	S	115	s	147	179	211	Ó	243	ó
20	52	84	T	116	t	148	180	212	Ô	244	ô
21	53	85	U	117	u	149	181	213	Õ	245	õ
22	54	86	V	118	v	150	182	214	Ö	246	ö
23	55	87	W	119	w	151	183	215	×	247	÷
24	56	88	X	120	x	152	184	216	Ø	248	ø
25	57	89	Y	121	y	153	185	217	Ù	249	ù
26	58	90	Z	122	z	154	186	218	Ú	250	ú
27	59	91	[123	{	155	187	219	Û	251	û
28	60	92	\	124		156	188	220	Ü	252	ü
29	61	93]	125	}	157	189	221	Ý	253	ý
30	62	94	^	126	~	158	190	222	ƒ	254	ƒ
31	63	95	_	127		159	191	223	ß	255	ÿ

UNICODE

- ▶ Preko milijun znakova svih svjetskih jezika
- ▶ Razlikuje vrste slova → u latinci razlikuje velika i mala slova
- ▶ Više načina kodiranja – najčešća varijanta UTF-8
- ▶ Ako neko slovo ne vidite na ekranu (prikazano je kao pravokutnik s nekih čudnim znakovima) → znači da niti jedan od fontova instaliranih na vašem računalu ne sadrži opis tog slova → ipak, podaci na računalu se pravilno pohranjuju i razmjenjuju kroz mrežu

UTF-8

- ▶ Unicode Transformation Format 8
- ▶ Kompatibilan sa 7-bitnim ASCII
- ▶ Većina nama zanimljivih znakova – kodirana sa 16 bita
- ▶ Većina svjetskih znakova – kodirana do 48 bitova

-
- ▶ ASCII vrijednost znaka dobiva se naredbom `ord()`

```
>>> print('A', ord('A'), 'a', ord('a'))
A 65 a 97
>>> print('+', ord('+'), ' ', ord(' '))
+ 43  32
>>> print('\n', ord('\n'), '\t', ord('\t'))

10      9
```

- ▶ Ispis više podataka unutar `print()` odvaja se zarezom

Formatirani ispis

▶ Ispis varijable (string) i teksta

```
>>> print(ime, ', kako si?')
Marko , kako si?
>>> print(ime+', kako si?')
Marko, kako si?
>>> print('{0}, kako si?'.format(ime))
Marko, kako si?
```

▶ Ispis varijable (broj) i teksta

```
>>> broj = 5
>>> print(broj, ' je najbolja ocjena.')
5 je najbolja ocjena.
>>> print(broj+ ' je najbolja ocjena.')
Traceback (most recent call last):
  File "<pyshell#186>", line 1, in <module>
    print(broj+ ' je najbolja ocjena.')
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> print(str(broj)+ ' je najbolja ocjena.')
5 je najbolja ocjena.
>>> print('{0} je najbolja ocjena.'.format(broj))
5 je najbolja ocjena.
>>>
```

-
- ▶ `str()` pretvara broj u string
 - ▶ `print('dio_teksta1 {0} dio_teksta2 {1}')`
`.format(varijabla1, varijabla2)`
 - ▶ Prva varijabla ima redni broj 0 i navodi se kao `{0}`,
druga varijabla ima redni broj 1 i navodi se kao `{1}`...

▶ Ispis više varijabli (brojeva, stringova) i teksta

```
>>> print('{0} {2} {0} {3} {4}'.format(5, 6, '+', '=', 5+5))
5 + 5 = 10
>>> print('Rezultat {2:s} brojeva {0:d} i {1:f} je {3:f}'.format(2, 3.5,
'množenja', 2*3.5))
Rezultat množenja brojeva 2 i 3.500000 je 7.000000
>>> print('Rezultat {2:s} brojeva {0:5d} i {1:5.3f} je {3:5.1f}'.format(2, 3.5,
'množenja', 2*3.5))
Rezultat množenja brojeva      2 i 3.500 je      7.0
>>> print('Rezultat {2:s} brojeva {0:5d} i {1:.3f} je {3:.1f}'.format(2, 3.5,
'množenja', 2*3.5))
Rezultat množenja brojeva      2 i 3.500 je 7.0
```

- ▶ **s** → ako ispisujemo string
- ▶ **d** → ako ispisujemo cijeli broj
- ▶ **f** → ako ispisujemo realni broj
- ▶ **{1:4.3f}** → drugi broj po redu, napisan na četiri mjesta, sa tri decimale. Broj koji promatramo je realan broj.

Kornjačina grafika

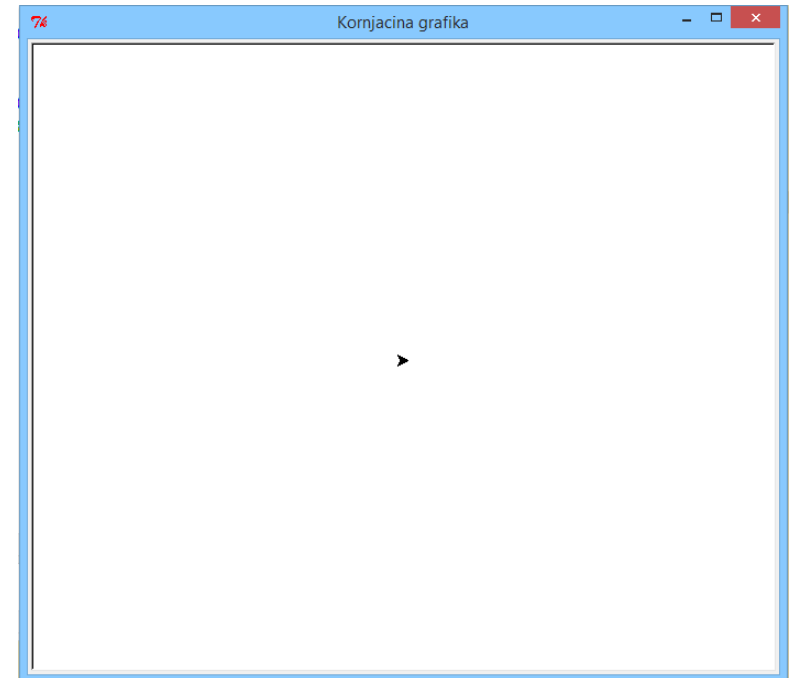
- ▶ Kornjača se kreće po ekranu i ako ima spušten rep ostavlja trag, a ako joj je rep dignut ne ostavlja trag. Glava kornjače (vrh strelice) predstavlja smjer kretanja.
- ▶ Osnovni atributi kornjače:
 - ▶ Položaj
 - ▶ Orijehtacija
- ▶ Dodatni atributi kornjače:
 - ▶ Boja
 - ▶ Širina
 - ▶ Gornji položaj (kada ne ostavlja trag)
 - ▶ Donji položaj (kada ostavlja trag)
- ▶ Što će kornjača raditi određuje se naredbama

Pokretanje kornjačine grafike

```
>>> from turtle import *  
>>> title('Kornjacina grafika')  
>>> reset()
```

I. omogućavanje funkcija povezanih s kornjačinom grafikom

2. otvara prozor veličine 600x600 piksela s naslovom “Kornjacina grafika”
3. svaki piksel predstavlja koordinatu u koordinatnom sustavu sa ishodištem u sredini prozora



4. pojavljuje se kornjača čija glava gleda u smjeru

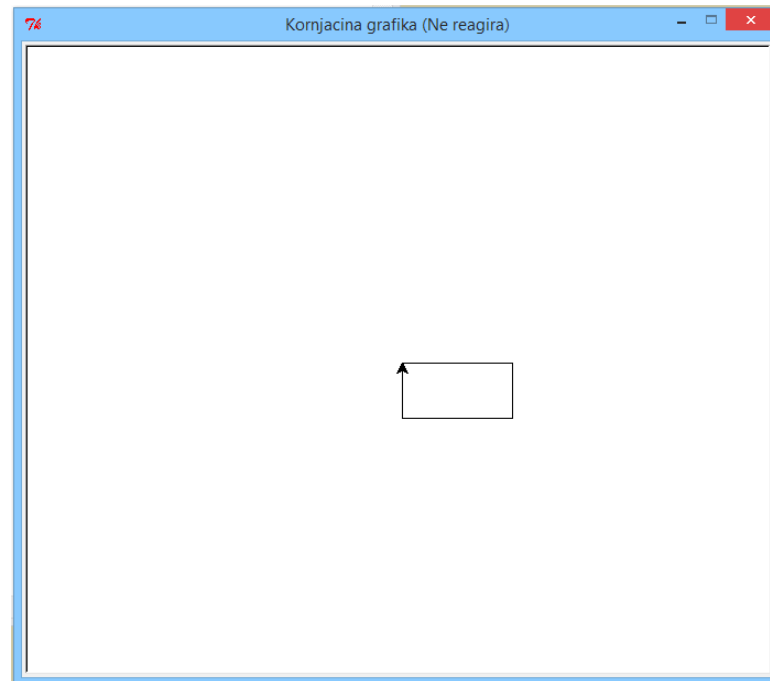
5. kornjačina grafika se prekida zatvaranjem prozora “Kornjacina grafika”

pozitivne osi x

Primjer: pravokutnik

- ▶ Napišite program koji će nacrtati pravokutnik stranica duljine 100 i 50.

```
>>> fd(100)
>>> rt(90)
>>> fd(50)
>>> rt(90)
>>> fd(100)
>>> rt(90)
>>> fd(50)
```



Relativno gibanje kornjače

Funkcija	Skraćeni naziv	Opis
<code>forward (duljina)</code>	<code>fd (duljina)</code>	Kornjača se pomiče za <code>duljina</code> unaprijed
<code>backward (duljina)</code>	<code>back (duljina)</code> <code>bk (duljina)</code>	Kornjača se pomiče za <code>duljina</code> unazad
<code>right (kut)</code>	<code>rt (kut)</code>	Kornjača se okreće za <code>kut</code> udesno
<code>left (kut)</code>	<code>lt (kut)</code>	Kornjača se okreće za <code>kut</code> ulijevo

Apsolutno gibanje kornjače

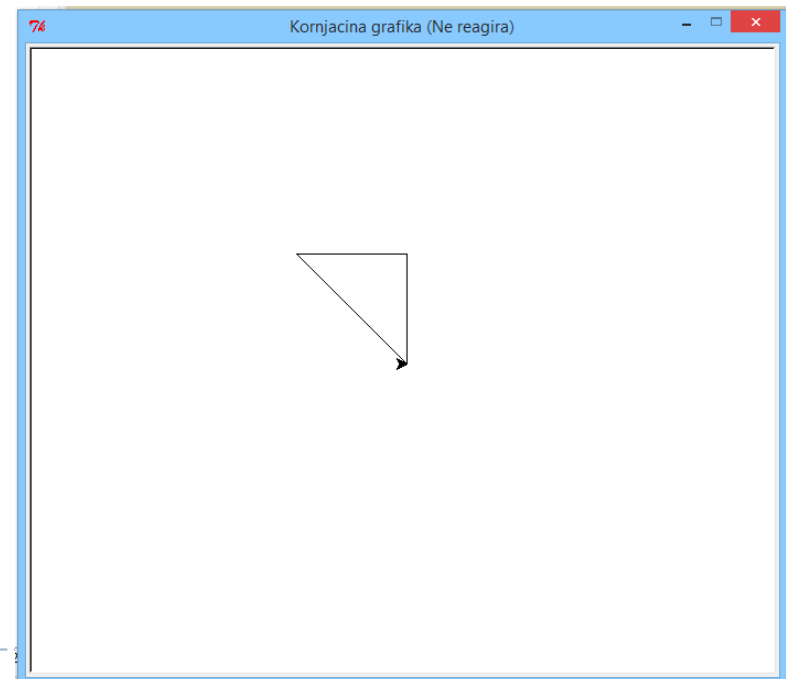
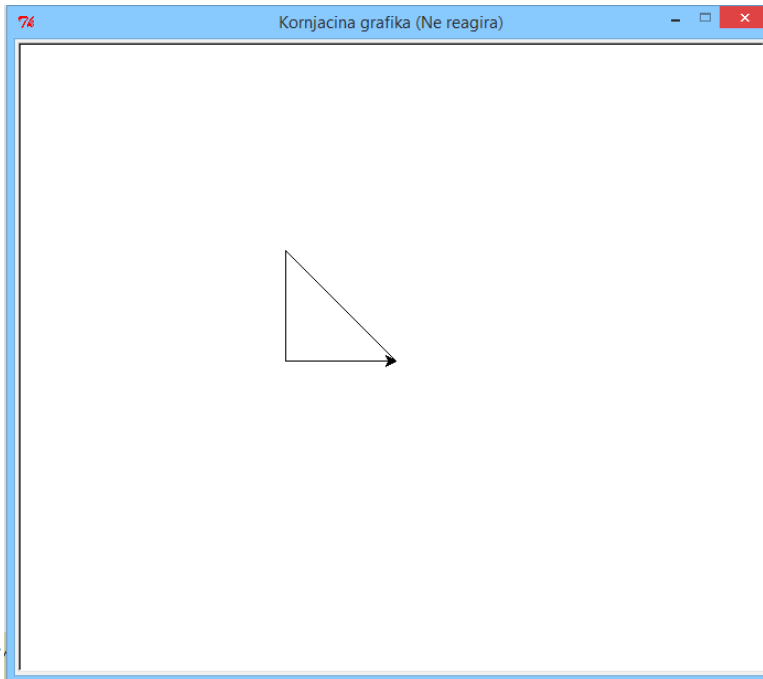
Funkcija	Drugi naziv	Opis
<code>goto(x, y)</code>	<code>setpos(x, y)</code>	Kornjača se pomiče na točku s koordinatama <code>(x, y)</code>
<code>setx(x)</code>		Prva koordinata kornjače se postavlja na <code>x</code> , druga koordinata se ne mijenja
<code>sety(y)</code>		Druga koordinata kornjače se postavlja na <code>y</code> , prva koordinata se ne mijenja
<code>setheading(kut)</code>	<code>seth(kut)</code>	Kornjača se okreće tako da pokazuje u smjer kuta <code>kut</code>

Pravokutni trokut

- ▶ Nacrtajte pravokutni trokut lijevo od početne pozicije kornjače.

```
>>> goto (-100, 100)  
>>> sety(0)  
>>> setx(0)
```

```
>>> goto (-100, 100)  
>>> setx(0)  
>>> sety(0)
```



Upravljanje kornjačom

Funkcija	Drugi naziv	Opis
<code>pendown()</code>	<code>pd()</code> <code>down()</code>	Kornjača ostavlja trag
<code>penup()</code>	<code>pu()</code> <code>up()</code>	Kornjača ne ostavlja trag
<code>pensize(d)</code>	<code>width(d)</code>	Trag kornjače ima debljinu <code>d</code> jedinica
<code>circle(r)</code>		Kornjača crta krug radijusa <code>r</code>
<code>color('ime_boje')</code>		Kornjača će ostavljati trag u boji <code>'ime_boje'</code> (na engleskom)
<code>showturtle()</code>	<code>st()</code>	Kornjača postaje vidljiva
<code>hideturtle()</code>	<code>ht()</code>	Kornjača postaje nevidljiva

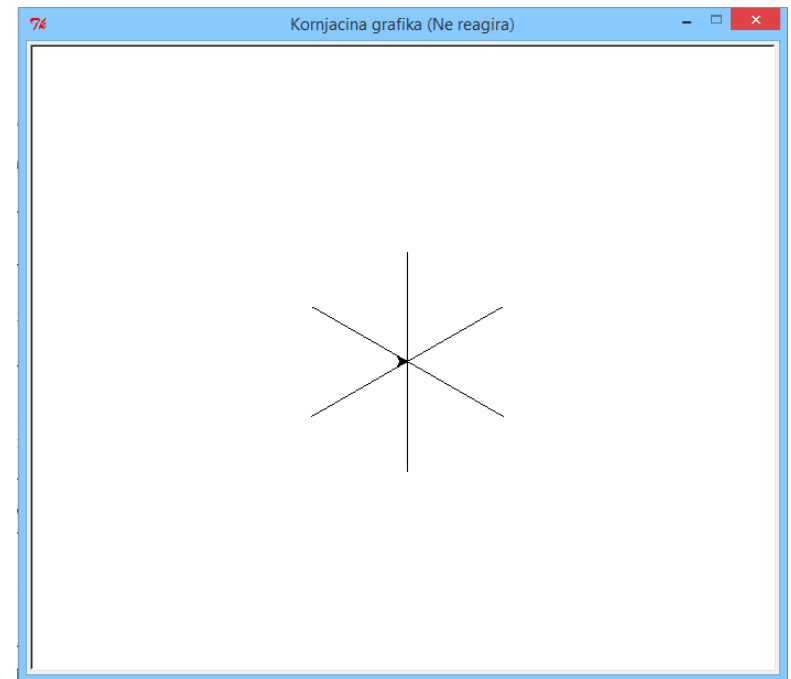
Upravljanje crtežom

Funkcija	Drugi naziv	Opis
<code>home ()</code>		Kornjača se postavlja u početni položaj
<code>undo ()</code>		Poništava se zadnja akcija kornjače
<code>reset ()</code>		Briše sve crteže u grafičkom prozoru i stavlja kornjaču u početni položaj
<code>clear ()</code>		Briše sve crteže u grafičkom prozoru, ali kornjača ostaje u zadnjem položaju

Zrakasta zvijezda

- ▶ Nacrtajte zrakastu zvijezdu sa šest krakova duljine 200.

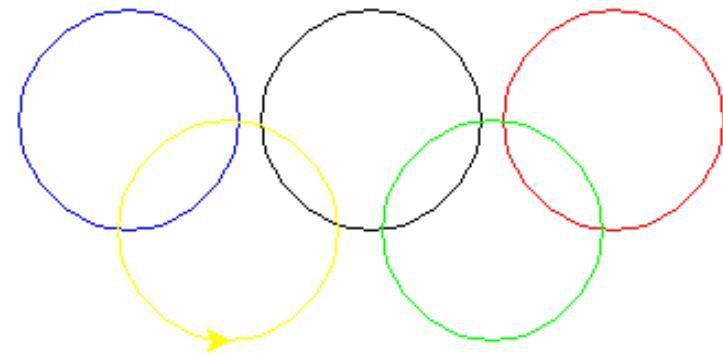
```
>>> reset ()
>>> lt (90)
>>> pu ()
>>> bk (100)
>>> pd ()
>>> fd (200)
>>> pu ()
>>> bk (100)
>>> lt (60)
>>> bk (100)
>>> pd ()
>>> fd (200)
>>> pu ()
>>> bk (100)
>>> lt (240)
>>> bk (100)
>>> pd ()
>>> fd (200)
>>> pu ()
>>> home ()
```



Olimpijski krugovi

- ▶ Nacrtajte znak Olimpijskih igara.

```
>>> reset()
>>> pu(), goto(-100,100); pd()
(None, None)
>>> color('blue'); circle(50)
>>> pu(), goto(10,100); pd()
(None, None)
>>> color('black'); circle(50)
>>> pu(), goto(120,100); pd()
(None, None)
>>> color('red'); circle(50)
>>>
>>> pu(), goto(65,50); pd()
(None, None)
>>> color('green'); circle(50)
>>> pu(), goto(-55,50); pd()
(None, None)
>>> color('yellow'); circle(50)
```



Detaljnije...

- ▶ <http://usingpython.com/python-variables/>
- ▶ <http://www.snee.com/logo/logo4kids.pdf>
- ▶ <http://hsin.hr/natjecanja.html>
- ▶ <https://code.org/learn>