Základy programovacieho jazyka Python

IT ACADEMY

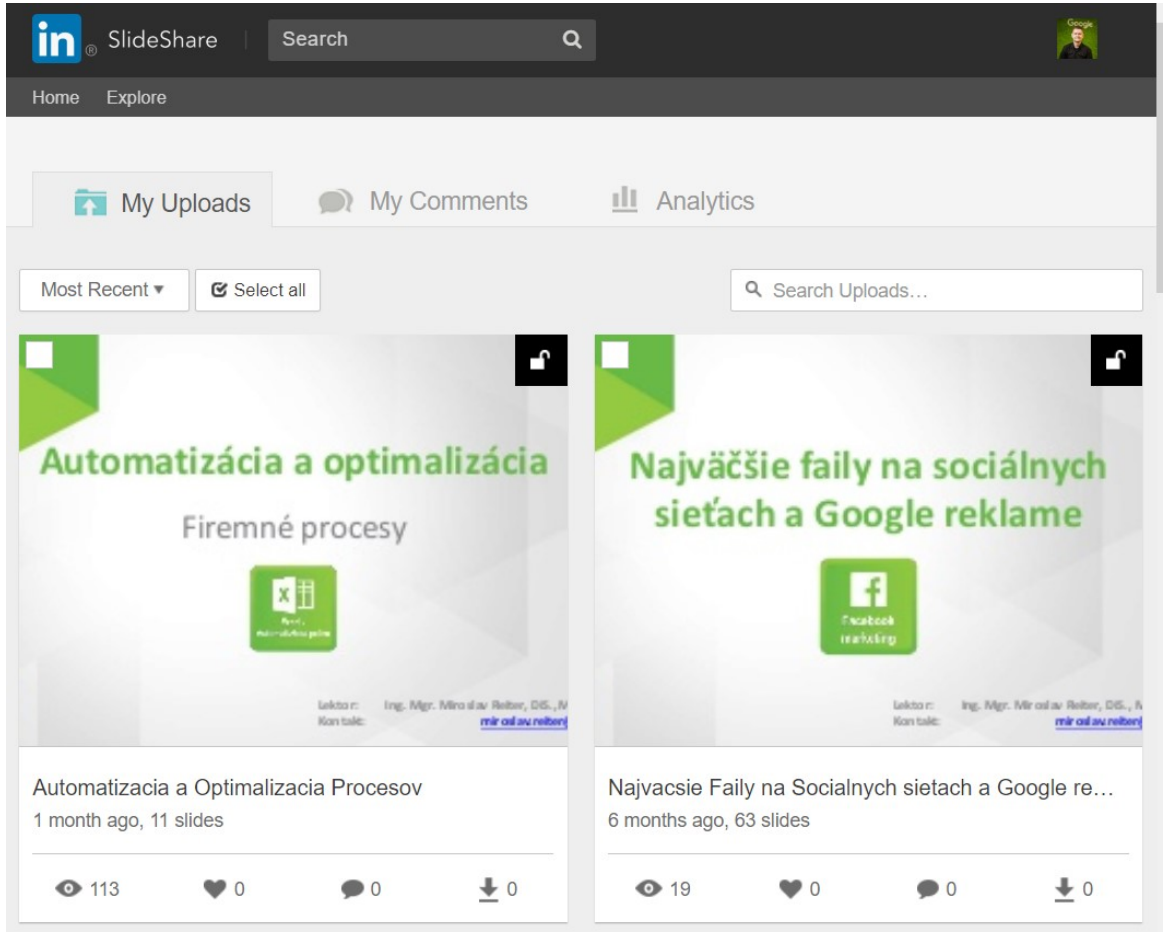
Lektor: PhDr. Ing. Mgr. Miroslav Reiter, MSc, DBA, MPA
Kontakt: miroslav.reiter@it-academy.sk

Ako začneme?

1. **Stiahni** si **Anaconda Navigator**
   - https://www.anaconda.com/products/

2. **Pridaj** si ma na **LinkedIn**
   - https://www.linkedin.com/in/miroslav-reiter/

3. **Prezentácia** a **materiály** po prednáške



IT ACADEMY

ITKurzy

IT kurzy CSČ SAV, v.v.i. - VS SAV

Základy programovacieho jazyku Python

Python je open-source, objektovo-orientovaný, vysoko úrovňový programovací jazyk. Python beží na mnohých variantách Unixu, na Macu a Windowse (súčasťou kurzu bude inštalácia na vašom systéme). Na kurz je potrebné prísť s vlastným notebookom (s ľubovolným operačným systémom podporujúcim Python).

1. **Základné informácie**
   - História a vlastnosti Pythonu
   - Pyhton v2 vs v3
   - Inštalácia Pythonu
   - Python IDLE
2. **Python syntax**
3. **Premenné**
4. **Typy dát**
   - String
   - List
   - Dictionary, Tuple, Set
5. **Operátory**
6. **Podmienky**
7. **Cykly**
   - For
   - While, break, continue
8. **Python objekty**
   - Class
   - Module

Moodle Slovenskej akadémie vied

Dostupné kurzy

Základy programovacieho jazyku Python

Učiteľ: Miroslav Reiter

Python je open-source, objektovo-orientovaný, vysoko úrovňový programovací jazyk. Python beží na mnohých variantách Unixu, na Macu, aj na Windowse (súčasťou kurzu bude inštalácia na ľubovolnom systéme). Na kurz je potrebné mať vlastné PC (s ľubovolným operačným systémom podporujúcim Python).

Základy programovacieho jazyku Python 2

Učiteľ: Miroslav Reiter

Objektovo orientované programovanie v Pythone

Spracovanie a vizualizácia dát v Pythone

V kurze Spracovanie a vizualizácia dát v Pythone bude poslucháč oboznámený knižnicami NumPy, Pandas a Matplotlib.

Neurónové siete s knižnicou TensorFlow

Kurz je zameraný na výučbu hlbokého učenia v jazyku Python. Poslucháčov naučíme pracovať s knižnicou Tensorflow. Tensorflow je aktuálne najpopulárnejšia knižnica zameraná na vytváranie, trénovanie a testovanie programov hlbokého učenia.
Odporúčanou podmienkou pre absolvovanie tohto kurzu je účasť na kurzoch: Základy programovacieho jazyku Python, Objektovo orientované programovanie v Pythone, Spracovanie a vizualizácia dát v Pythone.

Štatistika v R

ZPP
Účastníci
Odznaky
Kompetencie
Známky
Všeobecné
Základné informácie
Premenné a typy dát
Operátory
Podmienky a cykly
Funkcie
Práca so súbormi
Objekty
Nástenka
Domovská stránka
Kalendár
Súkromné súbory
Content bank
Moje kurzy

Základy programovacieho jazyku Python

Nástenka / Moje kurzy / ZPP

Zapnúť upravovanie

Oznámenia
Informácie o kurze
Sylabus kurzu
Študijné materiály
Jupyter

login: student
heslo: Jupyter2021

Základné informácie

Prednáška
Hello World

Premenné a typy dát

Premenné a typy dát

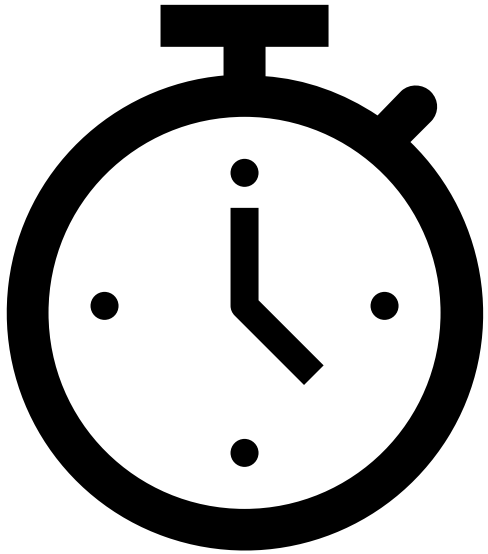
Operátory

Operátory

Úvodné informácie

- Časový rozvrh (9:00-13:30)
- Prestávky
- Mobilné telefóny a zariadenia

- Priprav si otázky a rovno sa pýtaj
- Interaktívna forma



O lektorovi - Miroslav Reiter

10000+ klientov a 500+ firiem

Programátor, Analytik, Manažér

Google, Microsoft ISTQB tréner

114 certifikácii

83 príručiek a publikácii

12 škôl

52 projektov

Vlastná firma

Miroslav Reiter

1. PhDr. VŠM (Podnikovný manažment)
2. **Ing. STU FEI (Aplikovaná informatika)**
3. Mgr. UK FM (Strategický manažment a marketing)
4. **Mgr. VŠM (Manažérstvo kvality)**
5. Mgr. VŠEMVŠ (Verejná správa)
6. Mgr. DTI (Učiteľstvo ekonomických predmetov)
7. DiS. AMOS (Cestovný ruch)
8. MBA LIGS (Executive management)
9. **DBA Humanum (IT manažment)**
10. MPA IES (Verejná správa a samospráva krajov)
11. **MSc. Humanum (Bezpečnosť informačných systémov)**
12. Ing. Paed. IGIP

NÁRODNÁ RADA SLOVENSKEJ REPUBLIKY

SAAIC
Slovenská akademická asociácia pre medzinárodnú spoluprácu

IT ACADEMY

DIGITÁLNA UNIVERZITA

SIEA
SLOVENSKÁ INOVAČNÁ A ENERGETICKÁ AGENTÚRA

VITA

SPŠD TRNAVA

FAKULTA MANAGEMENTU
Univerzita Komenského
v Bratislave

STU FIIT

Kde učím a vzdelávam?

Miroslav Reiter

získava status

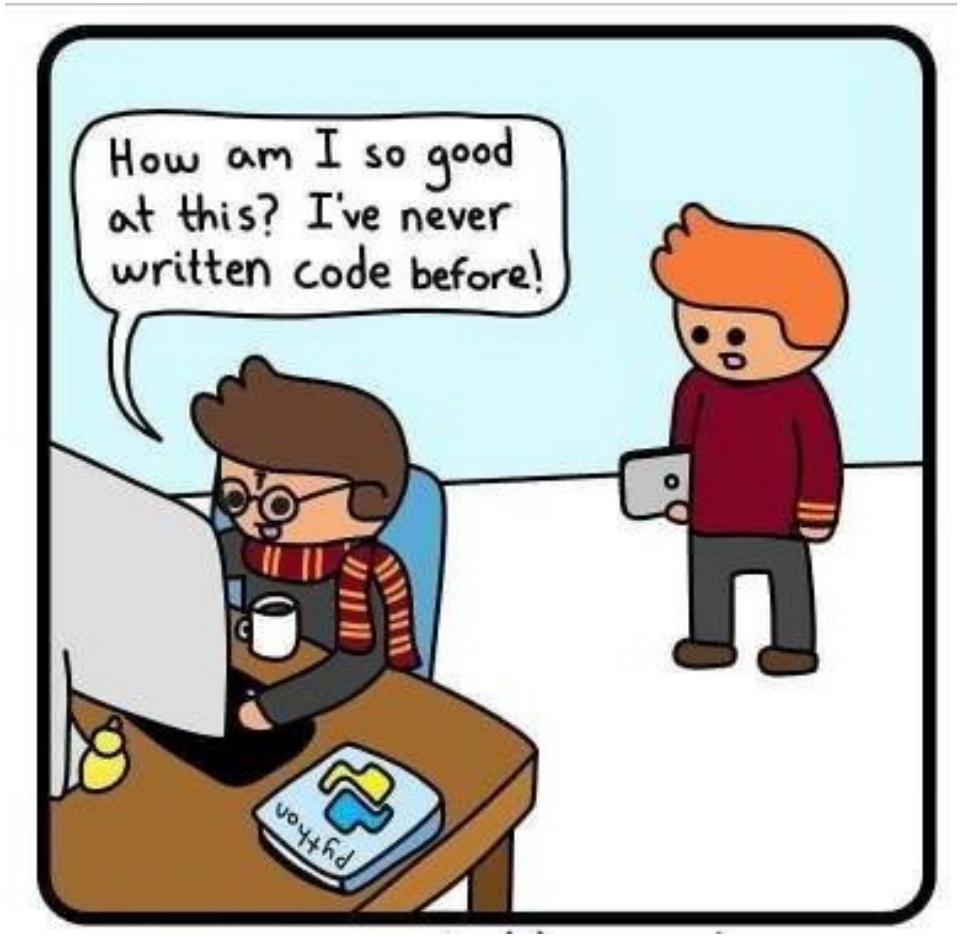
Google Certified Trainer

Automation

Google

Čo robíte?

1. Študent/učiteľ
2. Zamestnanec
3. Podnikateľ
4. Nezamestnaný/materská
5. Dievča pre všetko



National competence centre for high performance computing SLOVAKIA

EURO

EuroHPC
Joint Undertaking

SLOVENSKÁ AKADÉMIA VIED
SLOVAK ACADEMIA SCIENTIARUM

Národné kompetenčné centrum pre vysokovýkonné počítanie

Slovenská akadémia vied

Projekt EuroCC

Vzdelávanie

Kurzy:
itkurzy.sav.sk

Propagácia

Prednášky:
https://eurocc.nscc.sk/news/prednasky/

HPC služby

Prístup k výpočtovým prostriedkom

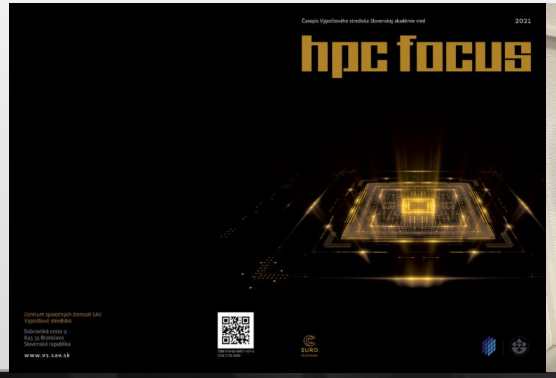
Mapovanie HPC prostredia

Prieskum:
https://eurocc.nscc.sk/mapping-survey/

Spolupráca

Pilotné projekty

Dlhodobá spolupráca



Qubit Conference

robíme it

Slovenská obchodná a priemyselná komora

S kým spolupracujeme:

- Akademické inštitúcie, univerzity, ústavy SAV,...
- Verejná správa
- Súkromné firmy, tretí sektor

Naučte sa pracovať v prostredí HPC systémov:

Najbližší kurz:

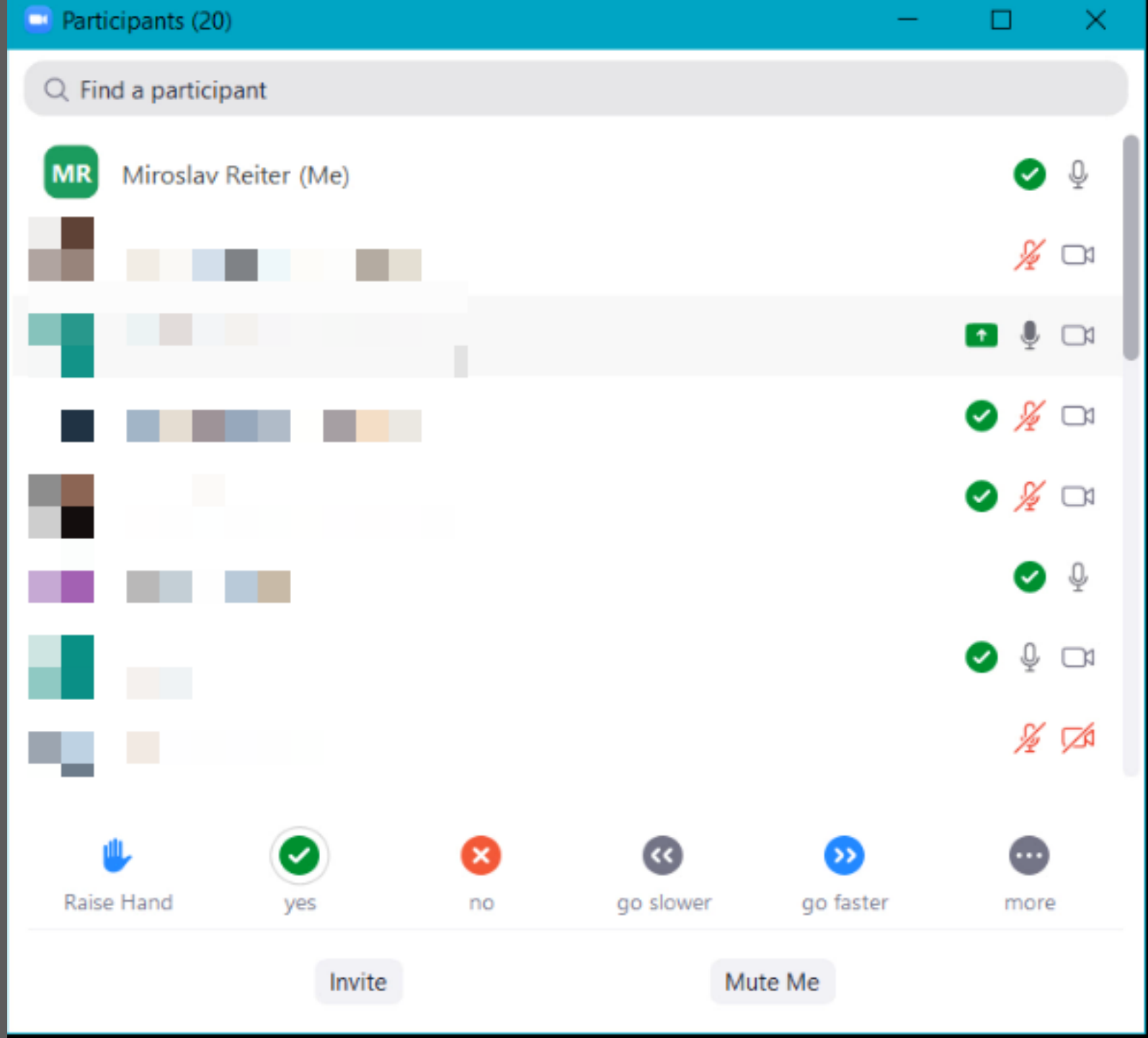
HPC infraštruktúra / 18. máj 2022

Hľadáme nových kolegov do tímu!
https://eurocc.nscc.sk/career/

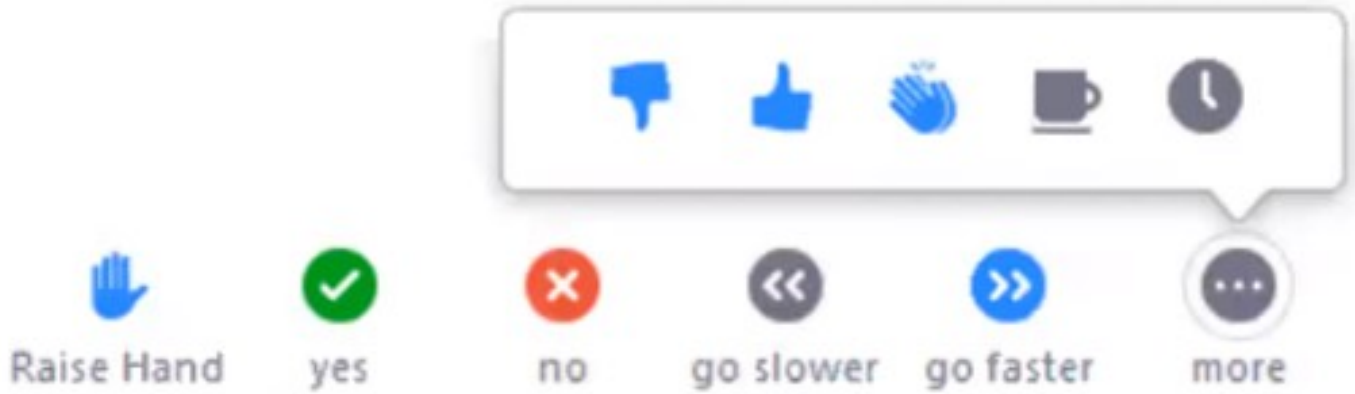
Sledujte nás na sociálnych sieťach:

Interaktívna prednáška

Aktívne používanie a zapájanie sa



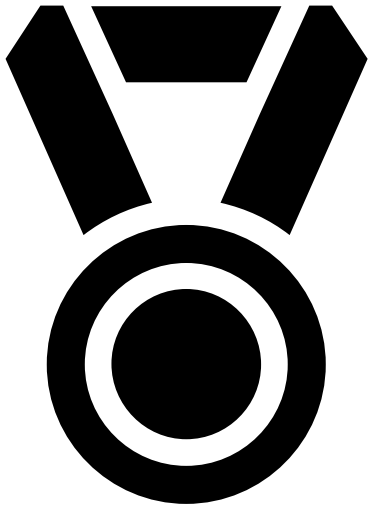
Používame Zoom



Vaše ciele a očakávania

1. Základy jazyka Python
2. Základy analytického myslenia
3. Základy programovania
4. Základy s PIP a knižnicami
5. Základy práce s platformou Anaconda Navigator

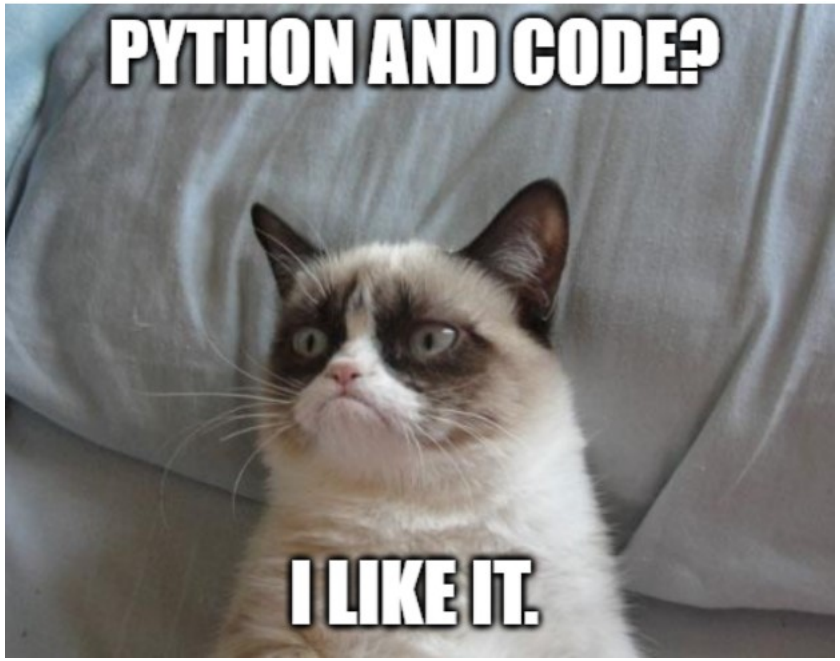
Zábava je v zaručená v každom bode :-)



Voľné miesta, platy, firmy

- Accenture
- Deloitte
- ČSOB
- Swiss Re
- DELL
- Slovenské elektrárne
- IBM (stáž)

- Profesia: ~153 pozícii
- Priemerný plat: 3 239 Eur



Python programátor/dátový analytik/data engineer

Google

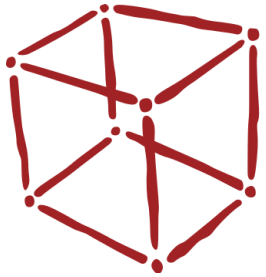
Seznam.cz

EXPONEA

IBM

NASA

Aké IDE mám použiť?



NetBeans

wxPython

Chceme úplne všetko!

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

Download

For Windows

Python 3.9 • 64-Bit Graphical Installer • 510 MB

Get Additional Installers

Open Source

Access the open-source software you need for projects in any field, from data visualization to robotics.

User-friendly

With our intuitive platform, you can easily search and install packages and create, load, and switch between environments.

Trusted

Our securely hosted packages and artifacts are methodically tested and regularly updated.

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Sign in to Anaconda Cloud

Home

Environments

Learning

Community

Documentation

Developer Blog

Applications on base (root) Channels Refresh

CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch

JupyterLab
1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch

Notebook
6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch

Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch

Qt Console
↗ 4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch

Spyder
↗ 4.0.1
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch

VS Code
1.52.1
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch

Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install

Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install

Microsoft Store

python

Aplikácie

Hranie

Knižnica

Pomocník

“python”

Filtre

Všetky oddelenia | Aplikácie | Hry

Python 3.9
Aplikácie • Developer tools
★★★★★ 4
Nainštalované

Python 3.8
Aplikácie • Developer tools
Bezplatné

Python 3.10
Aplikácie • Developer tools
Bezplatné

Python 3.7
Aplikácie • Developer tools
★★★★★ 1
Bezplatné

Learn Django and Python by GoLearningBus
Aplikácie • Books & reference
Bezplatné

WiBit.Net :: Programming in Python
Aplikácie • Education
0,99 €

计算机二级 Python 考试题库
Aplikácie • Education
Bezplatné

Data Science with Python
Aplikácie • Education

Python Programs
Aplikácie • Education

Learn Python
Aplikácie • Books & reference

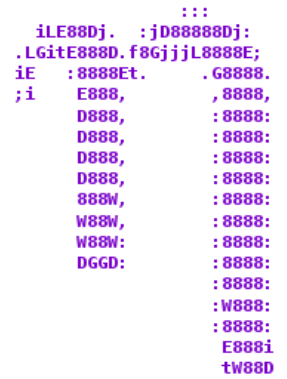
Introduction to Python Programming by...
Aplikácie • Books & reference

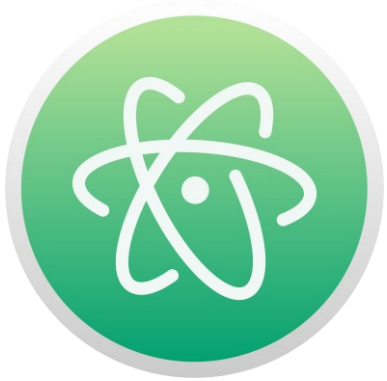
Python Playground
Aplikácie • Developer tools

Python Programming Manual For Absolute...
Aplikácie • Utilities & tools

Python User Tutorial
Aplikácie • Utilities & tools

Aký editor mám použiť?





IT ACADEMY

IDE ≠ editor

Dashboard

Welcome, itacademysk

CPU Usage: 0% used – 0.00s of 100s. Resets in 1 hour, 56 minutes More Info

File storage: 0% full – 100.0 KB of your 512.0 MB quota More Info

Upgrade Account

Recent
Consoles

You have no recent consoles.

New console:

$ Bash >>> Python More...

Version

2.7 3.7 3.8

Recent
Files

/home/itacademysk/.bashrc

/home/itacademysk/.gitconfig

/home/itacademysk/.profile

/home/itacademysk/.pythonstartup.py

/home/itacademysk/.vimrc

+ Open another file Browse files

Recent
Notebooks

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

All
Web apps

You don't have any web apps.

Open Web tab

MySQL

Postgres

Initialize MySQL

Let's get started! The first thing to do is to initialize a MySQL server:

Enter a new password in the form below, and note it down: you'll need it to access the databases once you've created them. You will only need to do this once.

New password:

Confirm password:

Initialize MySQL

This should be different to your main PythonAnywhere password, because it is likely to appear in plain text in any web applications you write.

Settings

Editor › Live Templates

By default expand with Tab

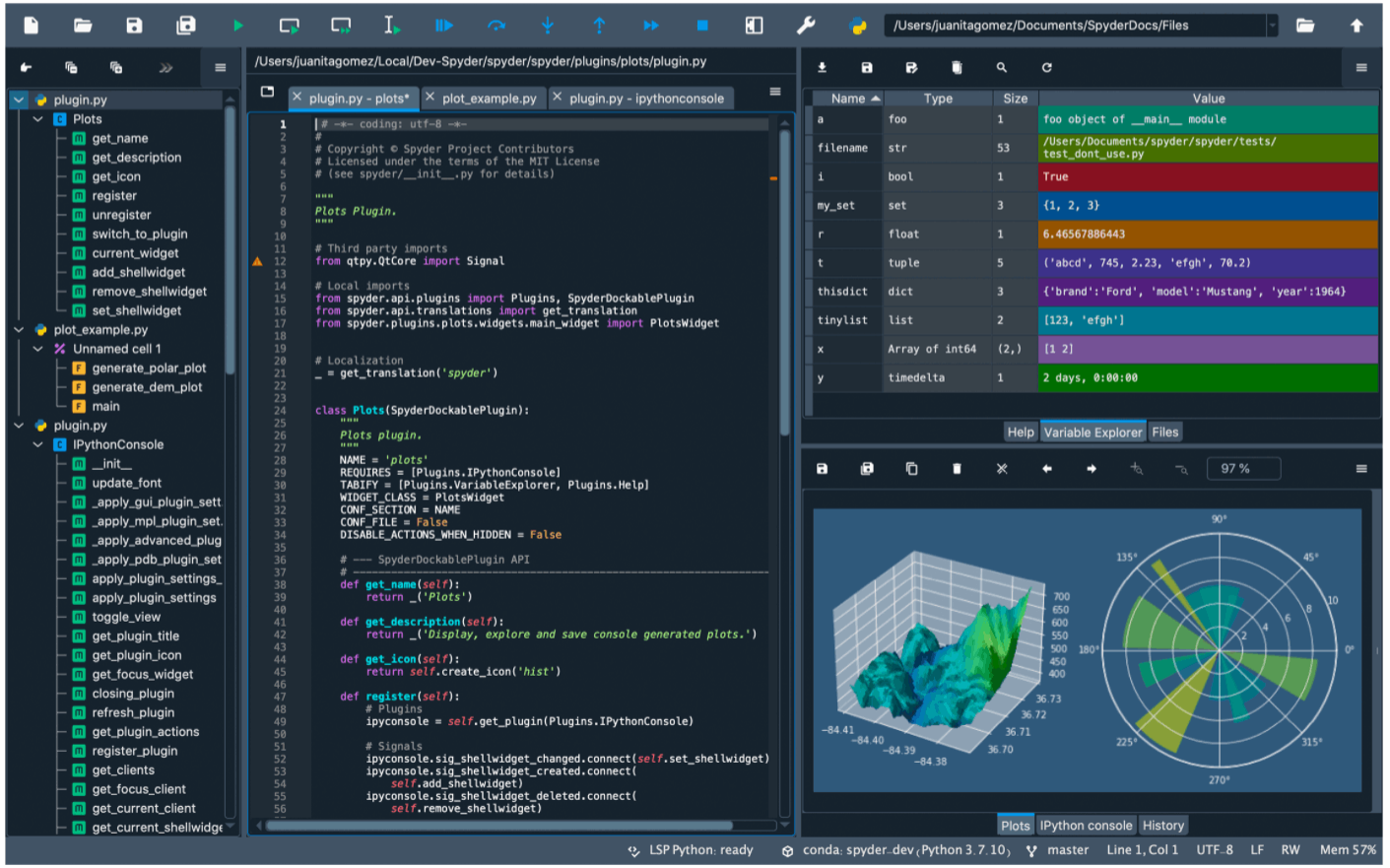
- Python
  - compd (Dict comprehension)
  - compdi (Dict comprehension with 'if')
  - compg (Generator comprehension)
  - compgi (Generator comprehension with 'if')
  - compl (List comprehension)
  - compli (List comprehension with 'if')
  - comps (Set comprehension)
  - compsi (Set comprehension with 'if')
  - iter (Iterate (for ... in ...))
  - itere (Iterate (for ... in enumerate))
  - main (if __name__ == '__main__')
  - prop (Property getter)
  - props (Property getter/setter)
  - propsd (Property getter/setter/deleter)
  - super ('super(...)' call)
- R
- React

No live templates are selected

- Appearance & Behavior
- Keymap
- Editor
  - General
  - Font
  - Color Scheme
  - Code Style
  - Inspections
  - File and Code Templates
  - File Encodings
  - Live Templates
  - File Types
  - Emmet
  - Images
  - Intentions
  - Language Injections
  - Spelling
  - TextMate Bundles
  - TODO
- Plugins
- Version Control

spyder

The Scientific Python Development Environment



VERSION

Spyder 5

Search ...

WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

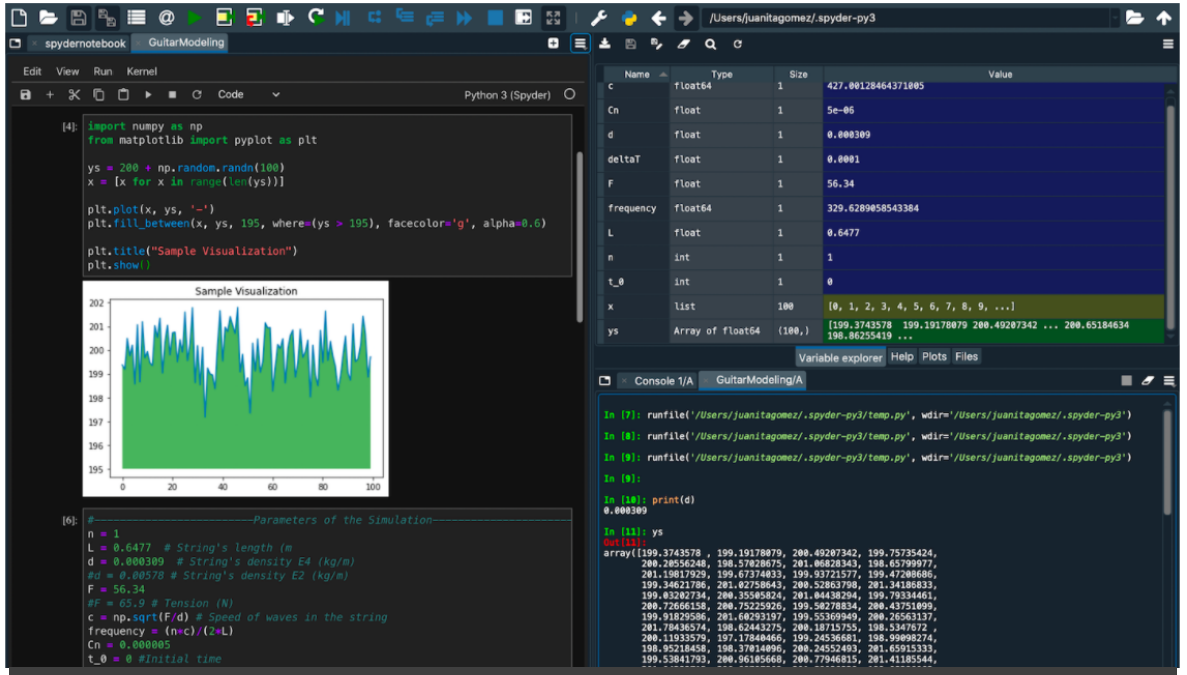
▶ WORKSHOPS

FAQ

Spyder Notebook

> **Warning**
>
> Currently, this plugin is still being ported to Spyder 5, and will likely not yet work or experience serious issues on this version of Spyder. A compatible version is expected soon. Thanks for your patience!

Spyder-notebook is a plugin that allows you to open, edit and interact with Jupyter Notebooks right inside Spyder.



conda install spyder-notebook -c spyder-ide

Edit this page

On this page

Installing the Notebook

Using the Notebook

Connecting an IPython Console

Additional Options

Kurz_Python_Den1.ipynb

Súbor Upraviť Zobraziť Vložiť Runtime Nástroje Pomocník Zmeny sa nedajú uložiť

+ Kód + Text Kopírovať na Disk

Útržky kódu

Filtrovať útržky kódu

- Adding form fields
- Camera Capture
- Cross-output communication
- display.Javascript to execute Jav...
- Downloading files or importing da...

Adding form fields Vložiť

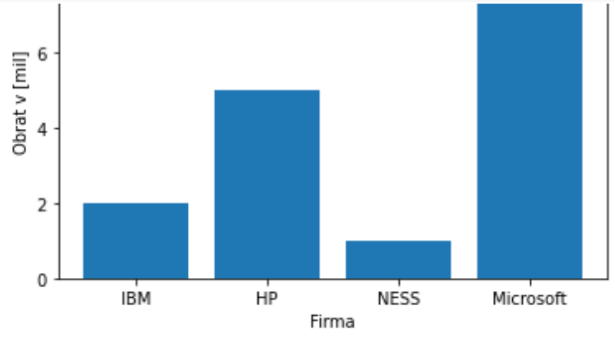
Forms example

Forms support multiple types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

```
#@title Example form fields
#@markdown Forms support many type

no_type_checking = ''  #@param
string_type = 'example'  #@param
slider_value = 142  #@param {type
number = 102  #@param {type: "numb
date = '2010-11-05'  #@param {type
pick_me = "monday"  #@param ['mond
select_or_input = "apples" #@param
#@markdown ---
```

Zobraziť zdrojový zápisník

```
tiene = True
# r g b, c m y k, w
farby_vlastne = ["black","pink", "b", "#CCCC00"]

print(type(y))
print(y)

plt.pie(y, labels = menovky, startangle = 0, explode = vykrojenie, shadow = tiene, colors = farby_vlastne)
plt.legend(title = "Produkty ABC s.r.o.")
plt.title("Analyza predaja produktov za Q1-2022")
plt.show()

x1 = np.array(["IBM", "HP", "NESS", "Microsoft"])
y1 = np.array([2, 5, 1, 9])

plt.bar(x1, y1)
plt.title("Porovnanie IT firiem 2022")
plt.xlabel("Firma")
plt.ylabel("Obrat v [mil]")
plt.show()

nahoda = np.random.normal(100, 10, 200)
print(nahoda)
plt.hist(nahoda)
plt.show()
```




```
[ 79.76069196  99.4155264  114.29926387 101.33767141  88.49106384
 111.70288892  91.32702578 102.53587004 108.38846479 114.34889501
  98.79114202 117.40488367  89.26174251  94.12100639 101.96805716
```

0 s dokončené

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\miros

welcome | untitled0 | Python_I_Za...

Edit View Run Kernel

Code | Python 3 (Spyder)

Kurz Python - 1. deň

Miroslav Reiter | miroslav.reiter@it-academy.sk | https://www.linkedin.com/in/miroslav-reiter/
Kurz Python | https://www.it-academy.sk/kurz/python/ | https://github.com/miroslav-reiter

Komentáre, kódovanie, tlač a docstringy

[1]:
```
# -*- coding: utf-8 -*-
# Toto je komentár (jednoriadkový)
"""Toto je docstring (document string)"""

# Pozor na nespravne zalomovanie riadku (Enter)
# SyntaxError: EOL while scanning string literal
# Emotikony https://emojipedia.org/
print("Python je fajnovy jazyk!")
print(__doc__)
print("🖱 🎲 🔮")
print("Co bolo skorej? --> ", min(['\N{CHICKEN}', '\N{EGG}']))
```

```
Python je fajnovy jazyk!
Toto je docstring (document string)
🖱 🎲 🔮
Co bolo skorej? -->  🐔
```

Premenné a typy

```
...
```

```
Milujem Python
Milujem Python
Milujem Python

Nazov produktu je: Hypoteka pre mladych 2021
Splatka vasej hypoteky je: 600 Eur
Urokova sadzba je: 1.5 % p.a.
Je k dispozicii: False
```

Editor | Notebook

Name | Type | Size | Value

Variable explorer | Help | Plots | Files

Console 1/A

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]:
```

IPython console | History

LSP Python: ready | Kite: ready (no index) | conda: base (Python 3.8.5) | main [86] | Line 8, Col 1 | UTF-8 | CRLF | RW | Mem 47% | CPU 17% | 09:26

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\miros\.spyder-py3

C:\Users\miros\AppData\Local\Temp\kite_tutorial.py

temp.py kite_tutorial.py

```
# Welcome to...
#
#          `hmy+.                  ://:
#        .mMMMMMNho:`              NMMm
#       :NMMMMMMMMMMds/.`          NMMm             :ss:
#      +NMMMMMMMMMMMMMMMMmy+       NMMm            -MMMM-   ---
#    `oMMMMMMMMMMMMMMMMMMMMMo      NMMm             /ss/   :MMM+
#   `yMMMMMMMMNshmNMMMMMMMN`       NMMm                    /MMM+
#  .dMMMMMMMMm/hmhssydmMMM+        NMMm     `/yhhy. shhy ohmMMMmhhhh.  ./ydmmmdho-
#  omMMMMMMMd/mMMMMMmhsosy`        NMMm  .omMMmo.  mMMN odmMMMmdddd. omMNdsoshNMNy`
#   .+dMMMMy/mMMMMMMMMMm-          NMMm-yNMMh/`    mMMN   /MMM+     sMMN:`   `:NMMy
#     `-ymo/NMMMMMMMMMMd           NMMMNMMN/       mMMN   :MMM+     MMMNddddddNMMN
#        ``hMMMMMMMMMMM:           NMMm+mMMNs.     mMMN   :MMM+     MMMh//////////:
#          `:yNMMMMMMMh            NMMm `/dMMNy-   mMMN   :MMM+  `. sMMNo`    `-:
#            .+mMMMMM-             NMMm   `/dMMNy- mMMN   .MMMNddNN/ +NMMNdhydNNMs
#              `:yMMMy             yhhs     `/hhhh shhs    :ymmmdho:  `/sdmmmmhs/`
#                 `om.

""" Kite is your Python programming copilot. Kite will try to show you the
    right information at the right time as you code to prevent you from context
    switching out of your current line of thought.

    This tutorial will teach you how to use all of Kite's core features. You
    should be able to learn everything in 5 minutes.

    If you get stuck at any point, please visit https://help.kite.com/ or file
    an issue at https://github.com/kiteco/issue-tracker.
"""



""" PART 0: BEFORE WE START ============================================

    Spyder will by default try to start the Kite backend when the editor first
    starts. You can change this behavior by opening settings, clicking on
    "Completion and linting", "Advanced", and then changing Kite's "Start Kite
    Engine on editor startup" setting.

    Look for the Kite indicator in the bottom left corner of Spyder's status
    bar — It will tell you if Kite is ready and working. If the indicator reads
    "not running", then you'll have to start the Kite Engine manually before
    proceeding with the rest of this tutorial.
"""
```

Name	Type	Size	Value
a	int	1	5
b	str	1	Karol

Variable explorer Help Plots Files

Console 1/A

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')

In [2]: runfile('C:/Users/miros/.spyder-py3/temp.py', wdir='C:/Users/miros/.spyder-py3')
5
Karol

In [3]:
```

IPython console History

Štart

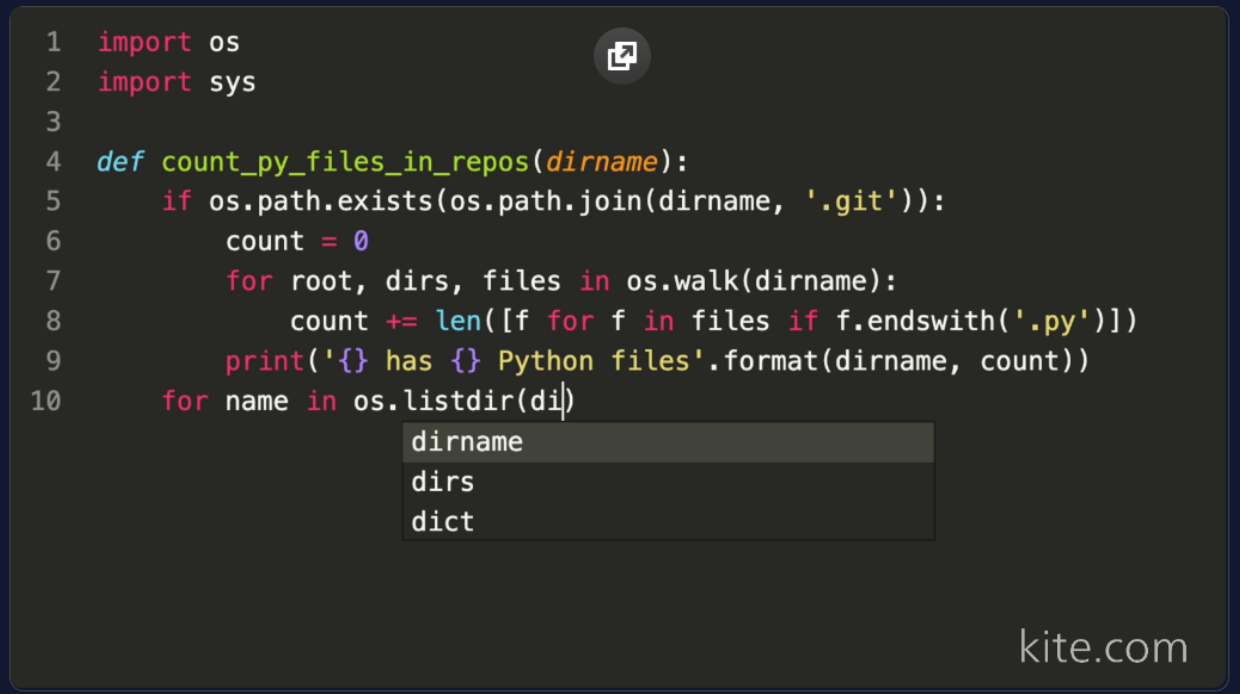
LSP Python: ready Kite: ready conda: base (Python 3.8.5) main [86] Line 19, Col 1 UTF-8-GUESSED CRLF RW Mem 41% CPU 6% 21:57

kite

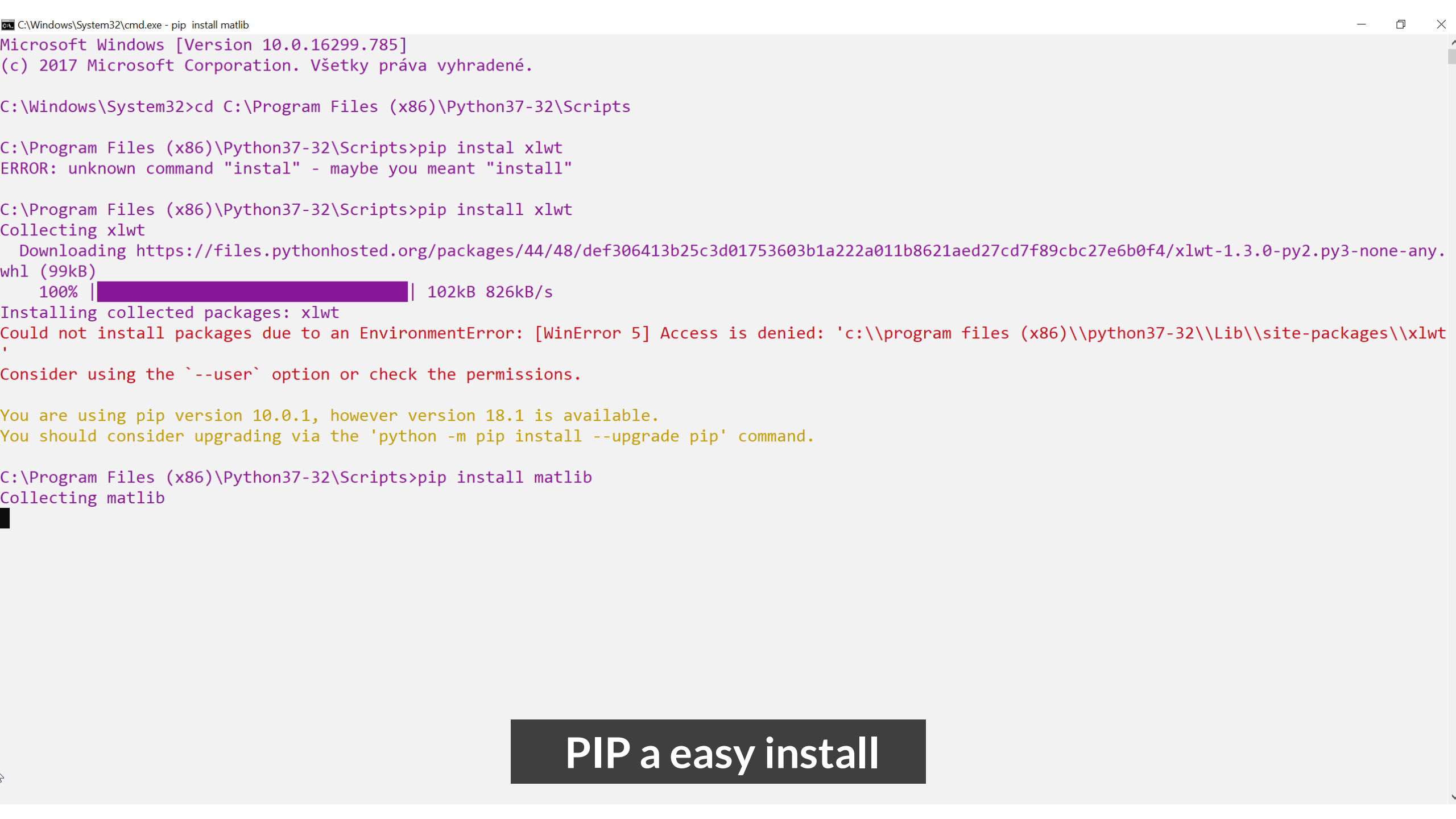
Code Faster. Stay in Flow.

Kite adds AI powered code completions to your code editor, giving developers superpowers.

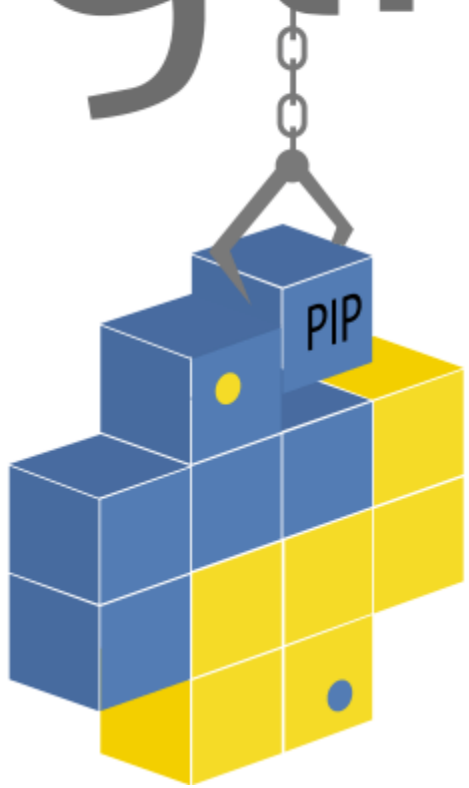
Download for Free



```
1  import os
2  import sys
3
4  def count_py_files_in_repos(dirname):
5      i
```

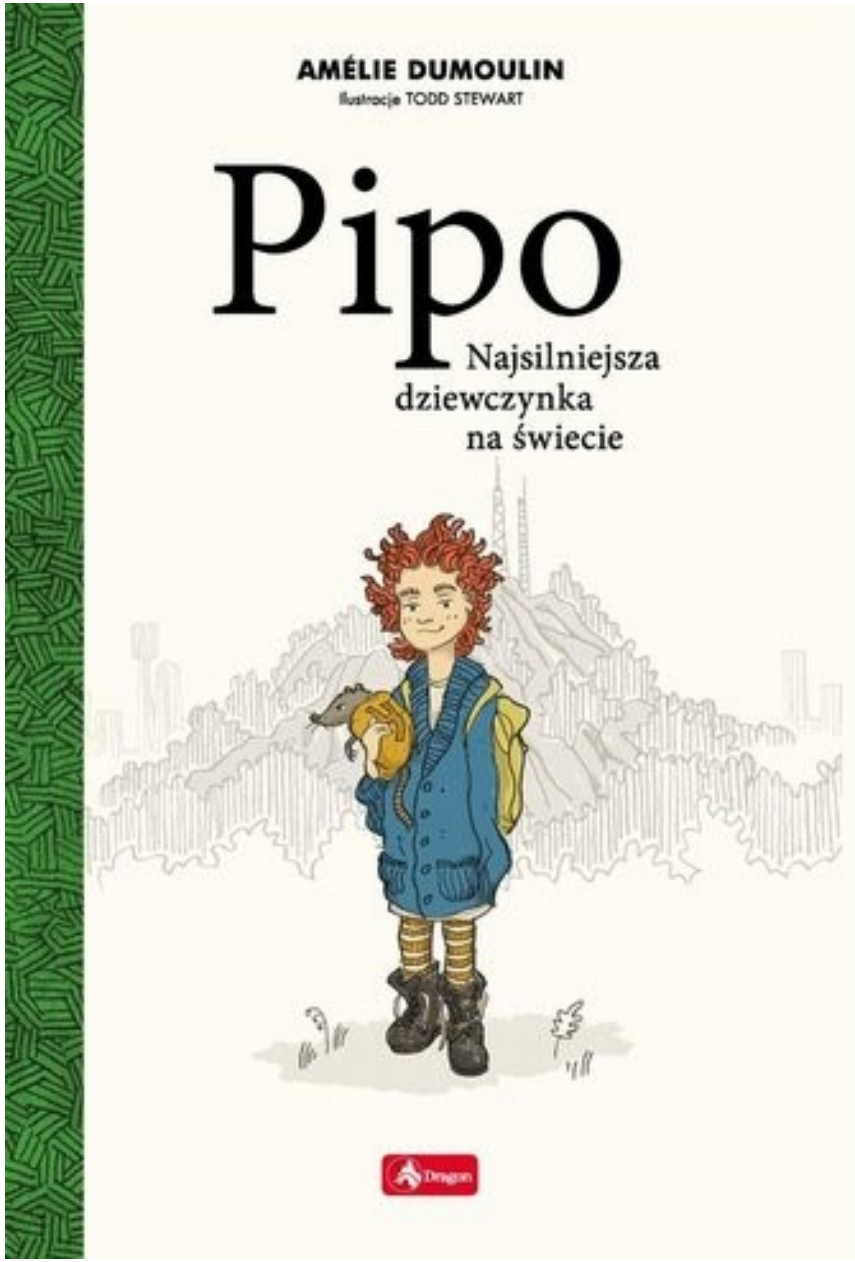

C:\Windows\System32\cmd.exe - pip install matlib
Microsoft Windows [Version 10.0.16299.785]
(c) 2017 Microsoft Corporation. Všetky práva vyhradené.
C:\Windows\System32>cd C:\Program Files (x86)\Python37-32\Scripts
C:\Program Files (x86)\Python37-32\Scripts>pip instal xlwt
ERROR: unknown command "instal" - maybe you meant "install"
C:\Program Files (x86)\Python37-32\Scripts>pip install xlwt
Collecting xlwt
Downloading https://files.pythonhosted.org/packages/44/48/def306413b25c3d01753603b1a222a011b8621aed27cd7f89cbc27e6b0f4/xlwt-1.3.0-py2.py3-none-any.whl (99kB)
100% | 102kB 826kB/s
Installing collected packages: xlwt
Could not install packages due to an EnvironmentError: [WinError 5] Access is denied: 'c:\\program files (x86)\\python37-32\\Lib\\site-packages\\xlwt'
Consider using the `--user` option or check the permissions.
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
C:\Program Files (x86)\Python37-32\Scripts>pip install matlib
Collecting matlib
PIP a easy install

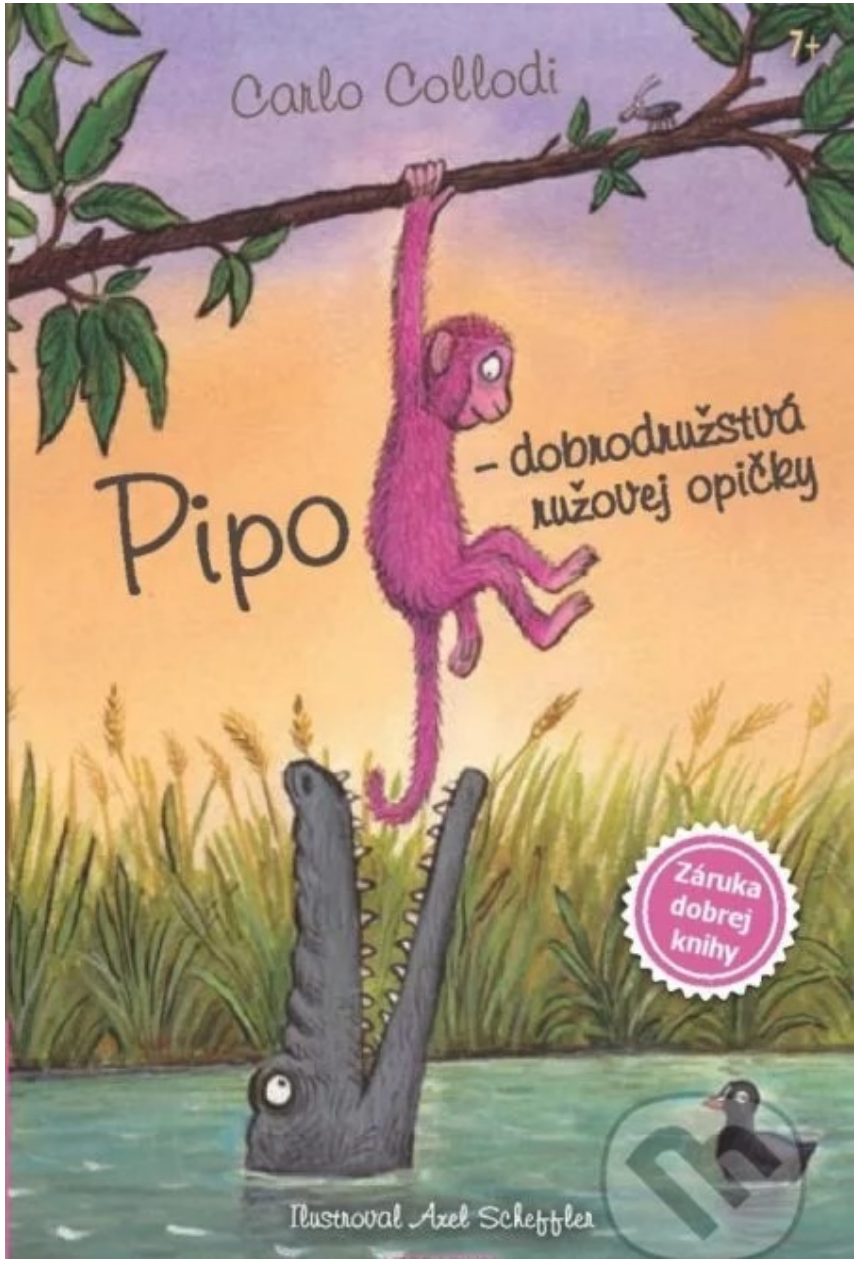
python™



pip Installation


Pipo
a Pogo
MLADÉ
LETA


AMÉLIE DUMOULIN
Ilustracje TODD STEWART
Pipo
Najsilniejsza
dziewczynka
na świecie
Dragon


Carlo Collodi
7+
Pipo
– dobrodružstvá
ružovej opičky
Záruka
dobrej
knihy
Ilustroval Axel Scheffler

sys Variables

argv	Command line args
builtin_module_names	Linked C modules
byteorder	Native byte order
check_interval	Signal check frequency
exec_prefix	Root directory
executable	Name of executable
exitfunc	Exit function name
modules	Loaded modules
path	Search path
platform	Current platform
stdin, stdout, stderr	File objects for I/O
version_info	Python version info
winver	Version number

sys.argv for *$ python foo.py bar -c qux --h*

sys.argv[0]	foo.py
sys.argv[1]	bar
sys.argv[2]	-c
sys.argv[3]	qux
sys.argv[4]	--h

os Variables

altsep	Alternative sep
curdir	Current dir string
defpath	Default search path
devnull	Path of null device
extsep	Extension separator
linesep	Line separator
name	Name of OS
pardir	Parent dir string
pathsep	Patch separator
sep	Path separator

Note Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"

Class Special Methods

__new__(cls)	__lt__(self, other)
__init__(self, args)	__le__(self, other)
__del__(self)	__gt__(self, other)
__repr__(self)	__ge__(self, other)
__str__(self)	__eq__(self, other)
__cmp__(self, other)	__ne__(self, other)
__index__(self)	__nonzero__(self)
__hash__(self)	
__getattr__(self, name)	
__getattribute__(self, name)	
__setattr__(self, name, attr)	
__delattr__(self, name)	
__call__(self, args, kwargs)	

String Methods

capitalize() *	lstrip()
center(width)	partition(sep)
count(sub, start, end)	replace(old, new)
decode()	rfind(sub, start ,end)
encode()	rindex(sub, start, end)
endswith(sub)	rjust(width)
expandtabs()	rpartition(sep)
find(sub, start, end)	rsplit(sep)
index(sub, start, end)	rstrip()
isalnum() *	split(sep)
isalpha() *	splitlines()
isdigit() *	startswith(sub)
islower() *	strip()
isspace() *	swapcase() *
istitle() *	title() *
isupper() *	translate(table)
join()	upper() *
ljust(width)	zfill(width)
lower() *	

Note Methods marked * are locale dependant for 8-bit strings.

List Methods

append(item)	pop(position)
count(item)	remove(item)
extend(list)	reverse()
index(item)	sort()
insert(position, item)	

File Methods

close()	readlines(size)
flush()	seek(offset)
fileno()	tell()
isatty()	truncate(size)
next()	write(string)
read(size)	writelines(list)
readline(size)	

Indexes and Slices (of *a=[0,1,2,3,4,5]*)

len(a)	6
a[0]	0
a[5]	5
a[-1]	5
a[-2]	4
a[1:]	[1,2,3,4,5]
a[:5]	[0,1,2,3,4]
a[:-2]	[0,1,2,3]
a[1:3]	[1,2]
a[1:-1]	[1,2,3,4]
b=a[:]	Shallow copy of a

Datetime Methods

today()	fromordinal(ordinal)
now(timezoneinfo)	combine(date, time)
utcnow()	strptime(date, format)
fromtimestamp(timestamp)	
utcfromtimestamp(timestamp)	

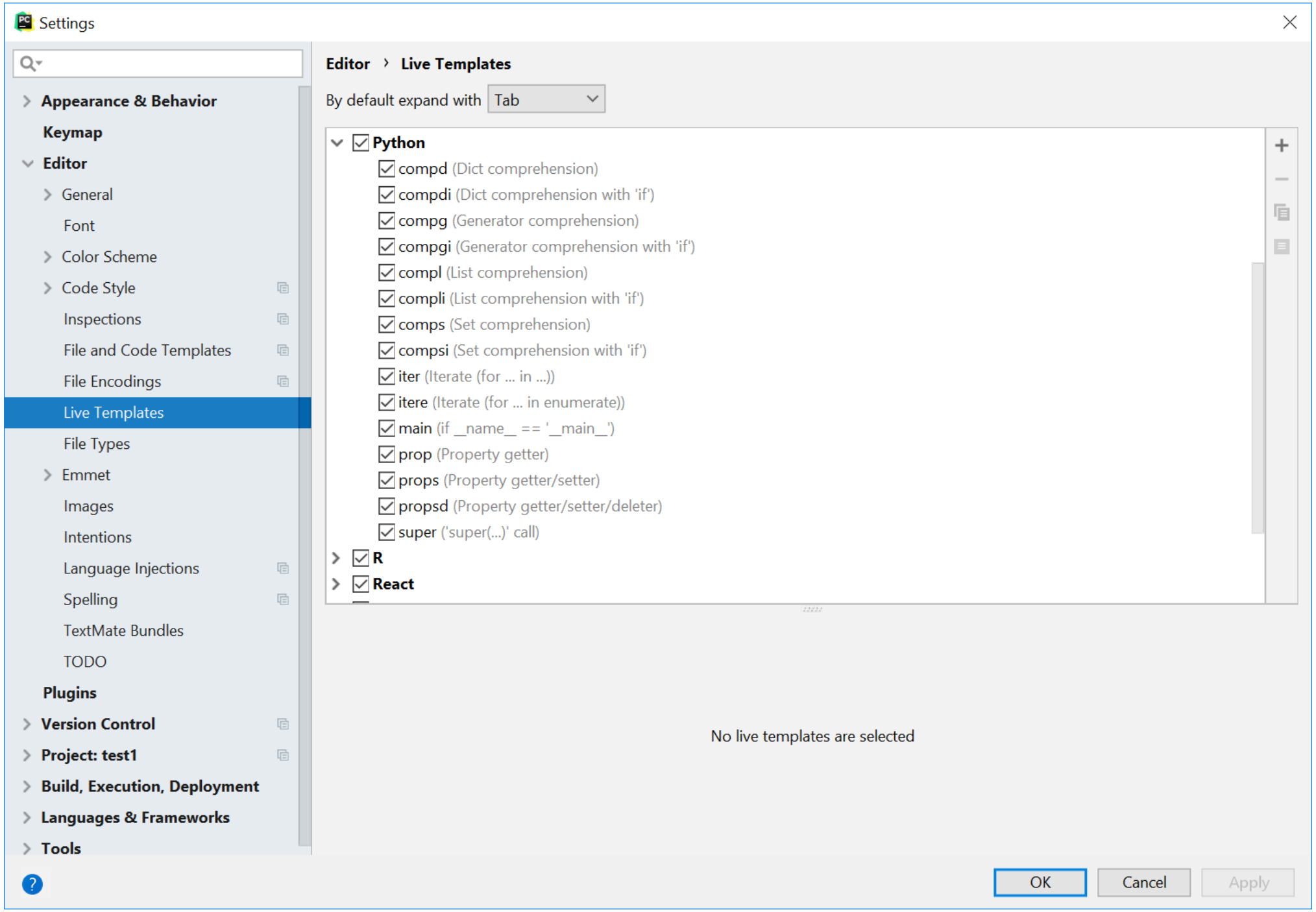
Time Methods

replace()	utcoffset()
isoformat()	dst()
__str__()	tzname()
strftime(format)	

Date Formatting (strftime and strptime)

%a	Abbreviated weekday (Sun)
%A	Weekday (Sunday)
%b	Abbreviated month name (Jan)
%B	Month name (January)
%c	Date and time
%d	Day (leading zeros) (01 to 31)
%H	24 hour (leading zeros) (00 to 23)
%I	12 hour (leading zeros) (01 to 12)
%j	Day of year (001 to 366)
%m	Month (01 to 12)
%M	Minute (00 to 59)
%p	AM or PM
%S	Second (00 to 61 [4])
%U	Week number [1] (00 to 53)
%w	Weekday [2] (0 to 6)
%W	Week number [3] (00 to 53)
%x	Date
%X	Time
%y	Year without century (00 to 99)
%Y	Year (2008)
%Z	Time zone (GMT)
%%	A literal "%" character (%)

1. Sunday as start of week. All days in a new year preceding the first Sunday are considered to be in week 0.
2. 0 is Sunday, 6 is Saturday.
3. Monday as start of week. All days in a new year preceding the first Monday are considered to be in week 0.
4. This is not a mistake. Range takes account of leap and double-leap seconds.


Settings
Appearance & Behavior
Keymap
Editor
General
Font
Color Scheme
Code Style
Inspections
File and Code Templates
File Encodings
Live Templates
File Types
Emmet
Images
Intentions
Language Injections
Spelling
TextMate Bundles
TODO
Plugins
Version Control
Project: test1
Build, Execution, Deployment
Languages & Frameworks
Tools
Editor › Live Templates
By default expand with Tab
Python
compd (Dict comprehension)
compdi (Dict comprehension with 'if')
compg (Generator comprehension)
compgi (Generator comprehension with 'if')
compl (List comprehension)
compli (List comprehension with 'if')
comps (Set comprehension)
compsi (Set comprehension with 'if')
iter (Iterate (for ... in ...))
itere (Iterate (for ... in enumerate))
main (if __name__ == '__main__')
prop (Property getter)
props (Property getter/setter)
propsd (Property getter/setter/deleter)
super ('super(...)' call)
R
React
No live templates are selected
OK
Cancel
Apply

Altair: Declarative Visualization in Python

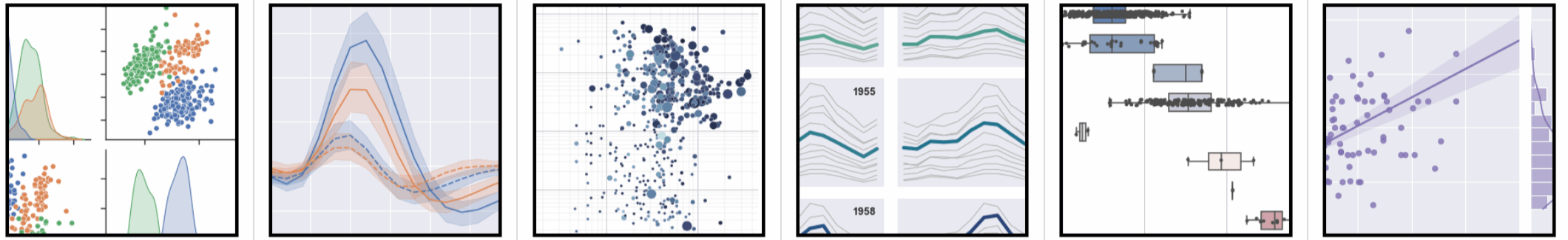
Altair is a declarative statistical visualization library for Python, based on Vega and Vega-Lite, and the source is available on GitHub.

With Altair, you can spend more time understanding your data and its meaning. Altair's API is simple, friendly and consistent and built on top of the powerful Vega-Lite visualization grammar. This elegant simplicity produces beautiful and effective visualizations with a minimal amount of code.

Getting Started

- Overview
- Installation
- Dependencies
- Development Install
- Basic Statistical Visualization

seaborn: statistical data visualization



Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see some of the things that you can do with seaborn, and then check out the tutorial or API reference to find out how.

To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stackoverflow or discourse, which have dedicated channels for seaborn.

Contents

Features

- Relational: API | Tutorial
- Distribution: API | Tutorial
- Categorical: API | Tutorial
- Regression: API | Tutorial
- Multiples: API | Tutorial
- Style: API | Tutorial
- Color: API | Tutorial

Back to top

Modeling Diagram

INTRODUCTION

To improve a process, it's important to first reflect on exactly what that process entails. Use our business process model diagram (BPMN) template to map out your business steps in a flowchart format.

Visualizing actions from start to finish enables you and your team to model your way of work and optimize it. By reviewing what is and isn't working, you may surprise yourselves in discovering competitive advantages. Ready, set, flow!

Participants 1+ | **Time** 1 - 3 hours | **Difficulty** Beginner

TOOL TIPS

Create connections at the speed of thought:

Hold + click and drag to **draw a connector.**

Turn on **connector points** to create new diagrams faster.

Click on the **connector points** to instantly add new connectors and shapes.

Change **connector styles.**

Switch between different shapes and sticky notes quickly.

Bulk edit objects of the same type by filtering your selection from the toolbar.

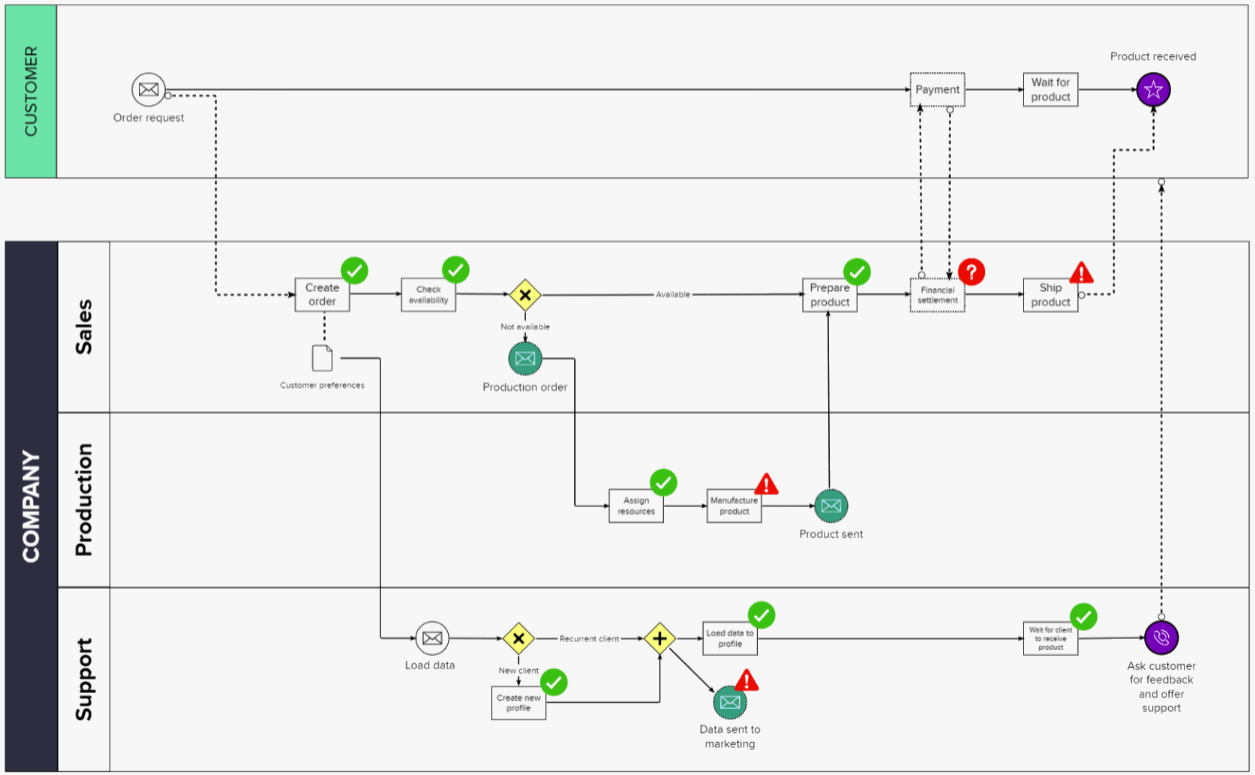
Extra: Add swimlanes and expand canvas (video)

Add labels to connectors double click on the line or choose the "+T" symbol from the connector's toolbar, then type your text.

Process: Sales interactions from order to delivery

2 Mapping

Gather with the team and from left to right map your agreed process from start point to end point.



Were included.

Events

Start | Intermediate | End

Gateway

Exclusive | Event based | Parallel | Inclusive | Complex

Activity

Task | Transaction

Flow

Sequence | Message | Association

Pool and swim lane

Pool | Swim lane | Swim lane

Artifact

Data | Group | Annotation

Tip!

You can use symbols for visual aid inside BPMN elements

Message | Success | Call | Issue

Work area

3 Assessment

Review the process.
- What is working?
- What is not working?
- What needs to work in order to *optimize* practice?

Copy/paste (or duplicate holding alt + drag and drop) to complete this task.

Efficient | Inefficient | Unknown task

4 Thinkering

Model the process.
- What changes need to be made?
- Who should be involved in those changes?

5 Planning

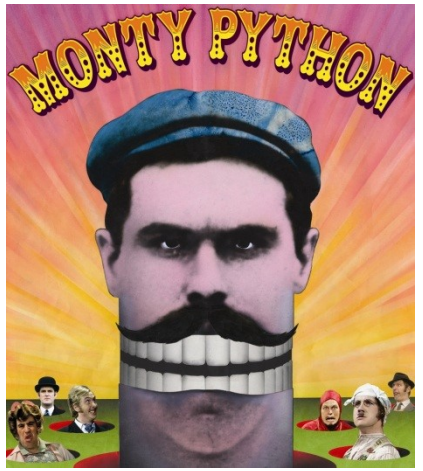
Next steps.
- Action plan for moving forward.

Next steps ideas | Moving forward!

Čo je Python?

Multi-paradigmový programovací jazyk

python™



Open source projekt

python™

Paradigm	Object-oriented, imperative, functional, procedural, reflective
Designed by	Guido van Rossum
Developer	Python Software Foundation
First appeared	1990; 28 years ago[1]
Stable release	3.7.1 / 20 October 2018; 3 days ago[2] 2.7.15 / 1 May 2018; 5 months ago[3]
Typing discipline	Duck, dynamic, strong; and **since version 3.5:** Gradual[4]
License	Python Software Foundation License
Filename extensions	.py, .pyc, .pyd, .pyo (prior to 3.5),[5] .pyw, .pyz (since 3.5)[6]
Website	www.python.org

python™

```
print("Hello, world!")
```

Guido van Rossum

@gvanrossum

Thanks for all the support (email and Twitter). I'm overwhelmed by the responses and won't be replying to most emails in person (except from core devs) but it's much appreciated. I'm still going to be around in the background!

1:07 AM - Jul 13, 2018

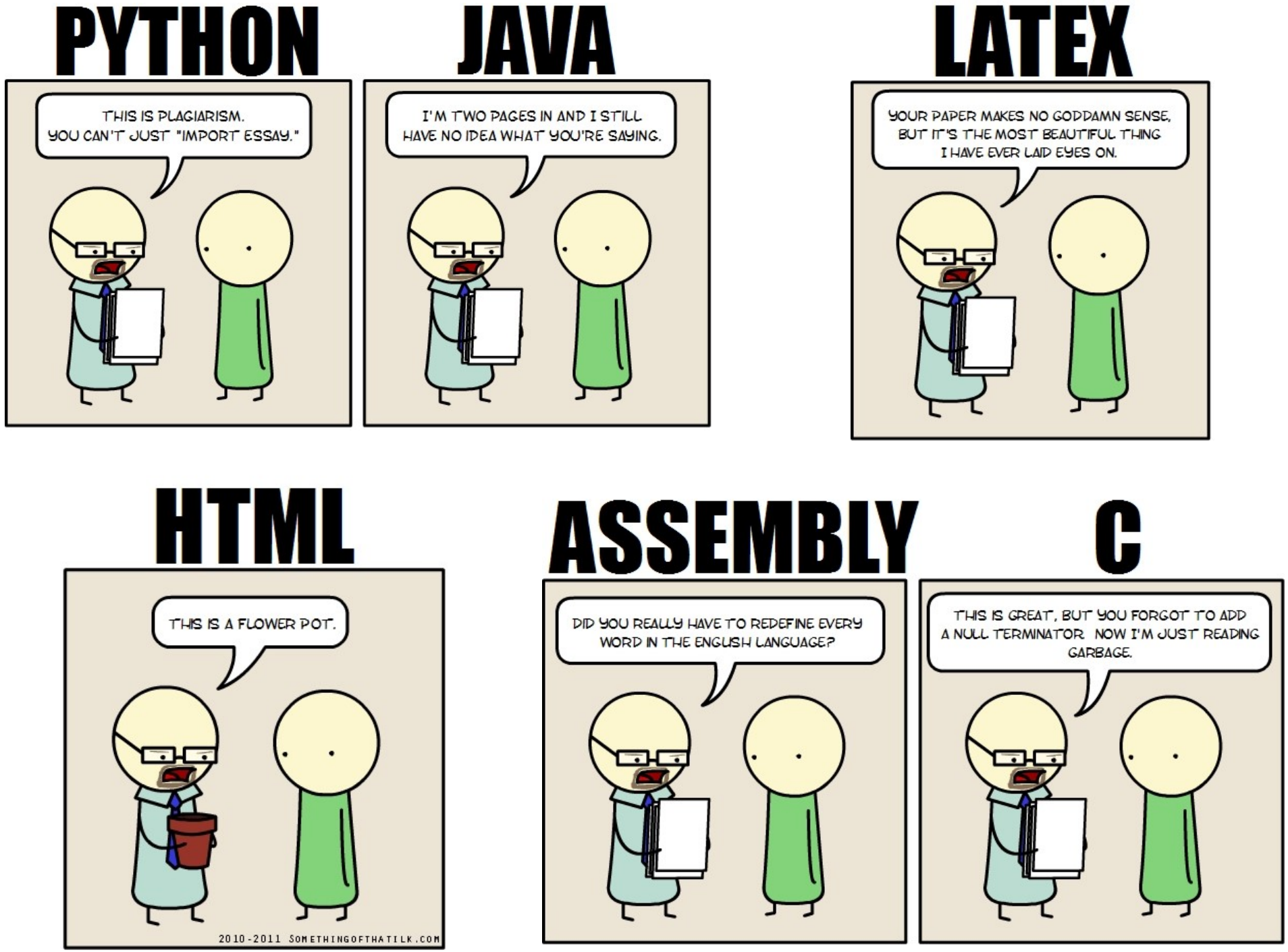
5,400 1,105 people are talking about this

IT ACADEMY

Začiatky v Pythone...

- Problémy pri interpretovaní
- Syntaktické chyby
- Sémantické chyby
- Problémy pri spúšťaní
- Chybové hlásenia
- Verzie

~~Python 2.*~~ alebo Python 3.*


PYTHON
THIS IS PLAGIARISM.
YOU CAN'T JUST "IMPORT ESSAY."
JAVA
I'M TWO PAGES IN AND I STILL HAVE NO IDEA WHAT YOU'RE SAYING.
LATEX
YOUR PAPER MAKES NO GODDAMN SENSE, BUT IT'S THE MOST BEAUTIFUL THING I HAVE EVER LAID EYES ON.
HTML
THIS IS A FLOWER POT.
2010-2011 SOMETHINGOFTHATILK.COM
ASSEMBLY
DID YOU REALLY HAVE TO REDEFINE EVERY WORD IN THE ENGLISH LANGUAGE?
C
THIS IS GREAT, BUT YOU FORGOT TO ADD A NULL TERMINATOR. NOW I'M JUST READING GARBAGE.

IT ACADEMY

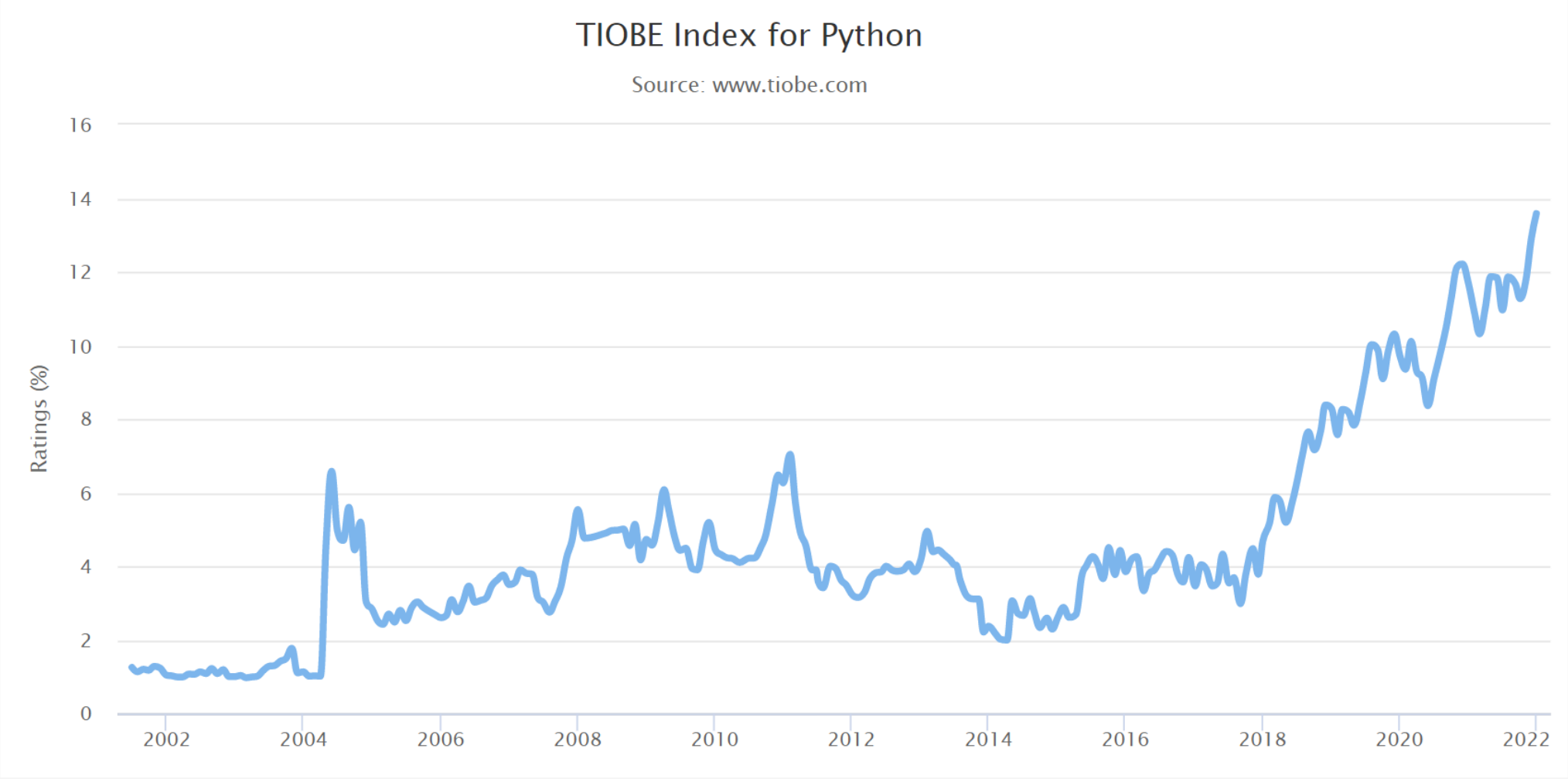
Jan 2022	Jan 2021	Change	Programming Language	Ratings	Change
1	3	▲	Python	13.58%	+1.86%
2	1	▼	C	12.44%	-4.94%
3	2	▼	Java	10.66%	-1.30%
4	4		C++	8.29%	+0.73%
5	5		C#	5.68%	+1.73%
6	6		Visual Basic	4.74%	+0.90%
7	7		JavaScript	2.09%	-0.11%
8	11	▲	Assembly language	1.85%	+0.21%
9	12	▲	SQL	1.80%	+0.19%
10	13	▲	Swift	1.41%	-0.02%
11	8	▼	PHP	1.40%	-0.60%
12	9	▼	R	1.25%	-0.65%

TIOBE INDEX

Highest Position (since 2001): #1 in Jan 2022

Lowest Position (since 2001): #13 in Feb 2003

Language of the Year: 2007, 2010, 2018, 2020



"You can't just copy-pase pseudocode into a program and expect it to work"

IT ACADEMY

Download

Download these documents

Docs by version

Python 3.11 (in development)
Python 3.10 (stable)
Python 3.9 (stable)
Python 3.8 (security-fixes)
Python 3.7 (security-fixes)
Python 3.6 (EOL)
Python 3.5 (EOL)
Python 2.7 (EOL)
All versions

Other resources

PEP Index
Beginner's Guide
Book List
Audio/Visual Talks
Python Developer's Guide

Python 3.10.2 documentation

Welcome! This is the official documentation for Python 3.10.2.

Parts of the documentation:

What's new in Python 3.10?

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

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!)

Indices and tables:

Global Module Index

quick access to all modules

General Index

all functions, classes, terms

Search page

search this documentation

Complete Table of Contents

Náš prvý program

python™

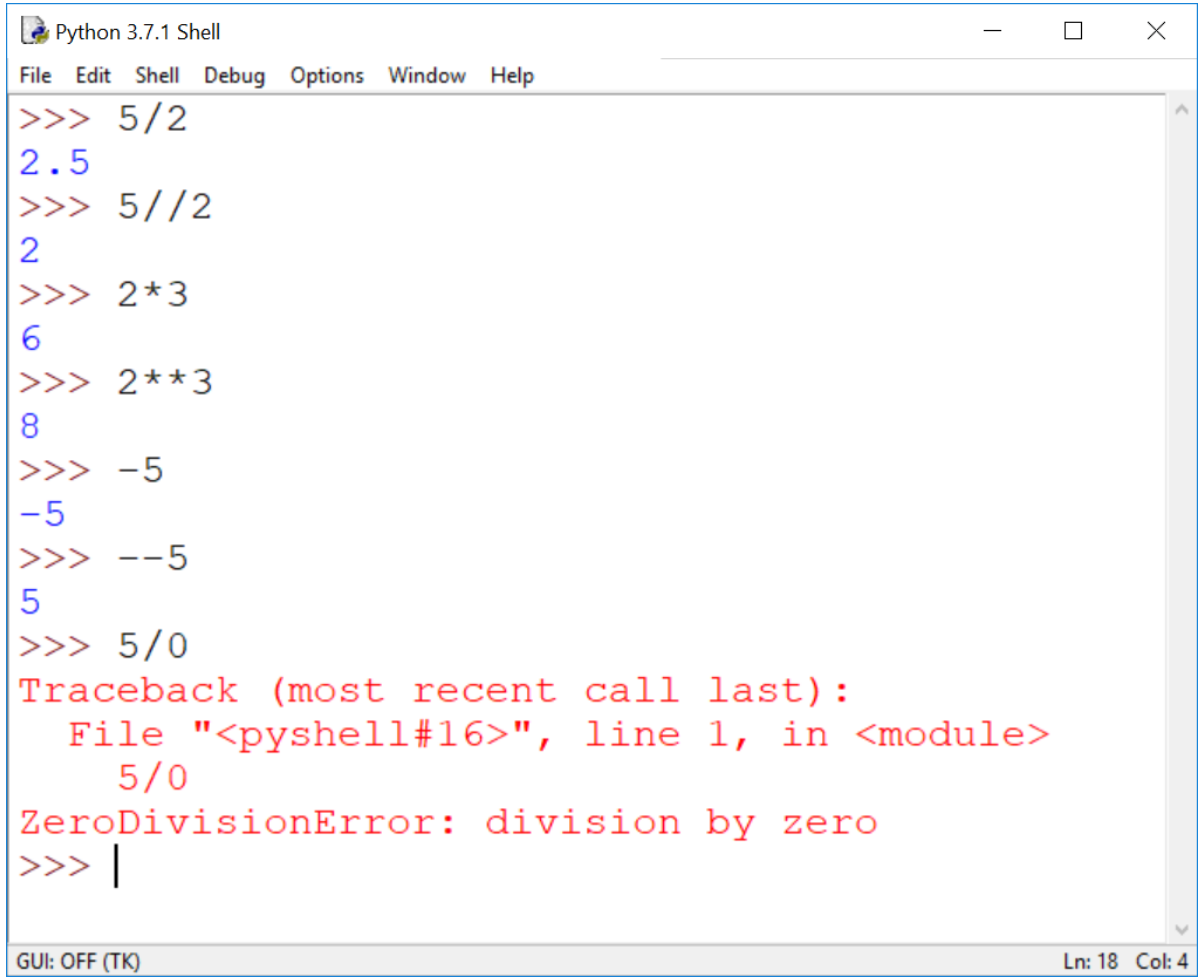
```
print("Hello, world!")
```

```
>>> print("Budem python programatorom")
Budem python programatorom
>>> print "Budem python programatorom"
Budem python programatorom
```

IT ACADEMY

filetype:py

Prvé výpočty



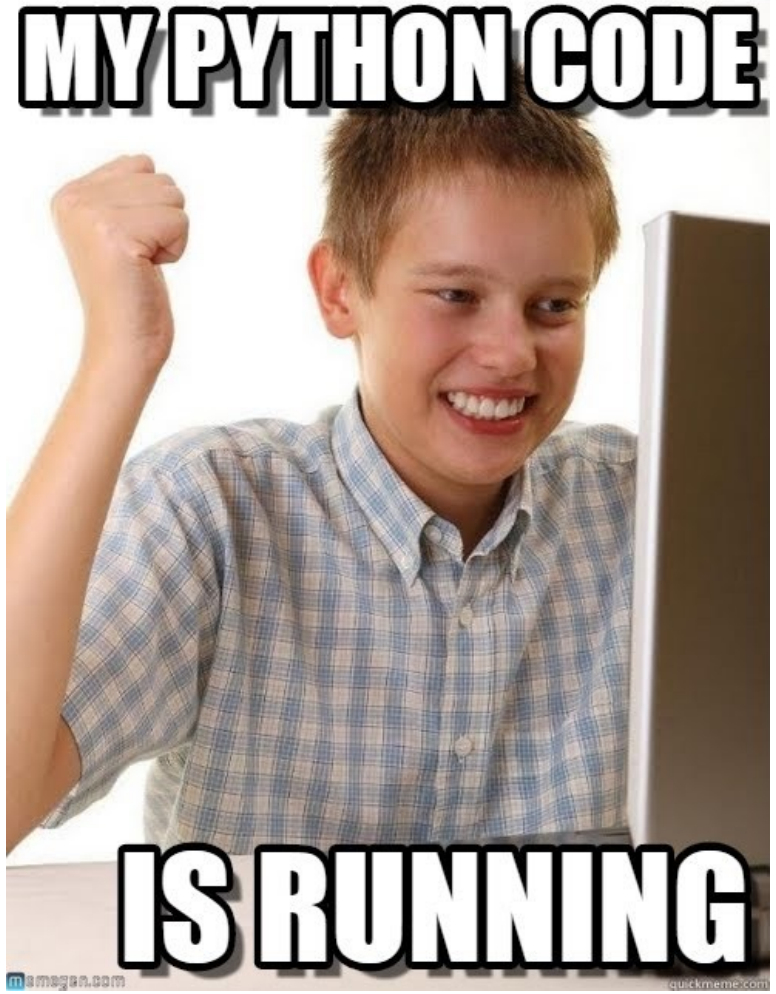
IT ACADEMY

Aritmetické operácie +, -, *, /, **,//

```
print(“Python is my
   favorite language.)

File "<stdin>", line 1
  print("Python is my favorite language)
                                        ^
SyntaxError: EOL while scanning string literal
```

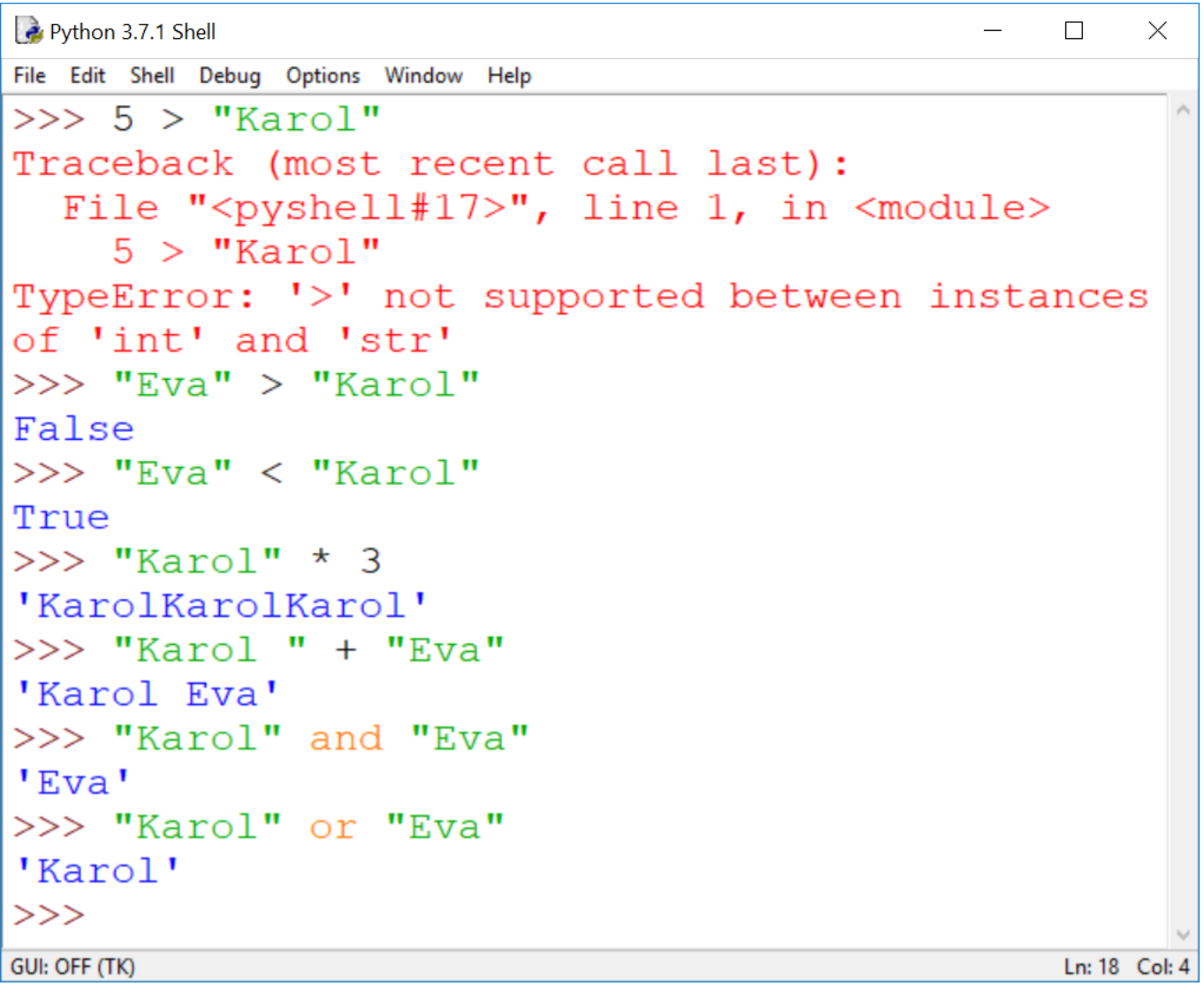
História a príkazy

```
>>> 1+1
2
>>> _
2
>>> print "Lala"
Lala
>>> _
2
>>> __

Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    __
NameError: name '__' is not defined
```



Ctrl + L

Práca s operátormi



Názov	Popis
+	sčítanie
-	odčítanie
*	násobenie
/	delenie
**	umocnenie
and	logické a (vracia 0 alebo 1)
or	logické alebo (vracia 0 alebo 1)
xor	vylučujúce alebo (vracia 0 alebo 1)
<	menší ako (vracia 0 alebo 1)
<=	menší ako alebo rovný (vracia 0 alebo 1)
==	rovný (vracia 0 alebo 1)
>=	väčší ako alebo rovný (vracia 0 alebo 1)
>	väčší ako (vracia 0 alebo 1)
!=	nie je rovný (vracia 0 alebo 1)

Nemiešaj typy a operácie

Základné príkazy pre výstup a vstup

- **print** – štandardný výstup
  - print("Stanem sa Python programatorom")
- **input** – štandardný vstup
  - **Čísla a vyrazy**:

```
meno = input("Vloz meno: ")
print meno
```

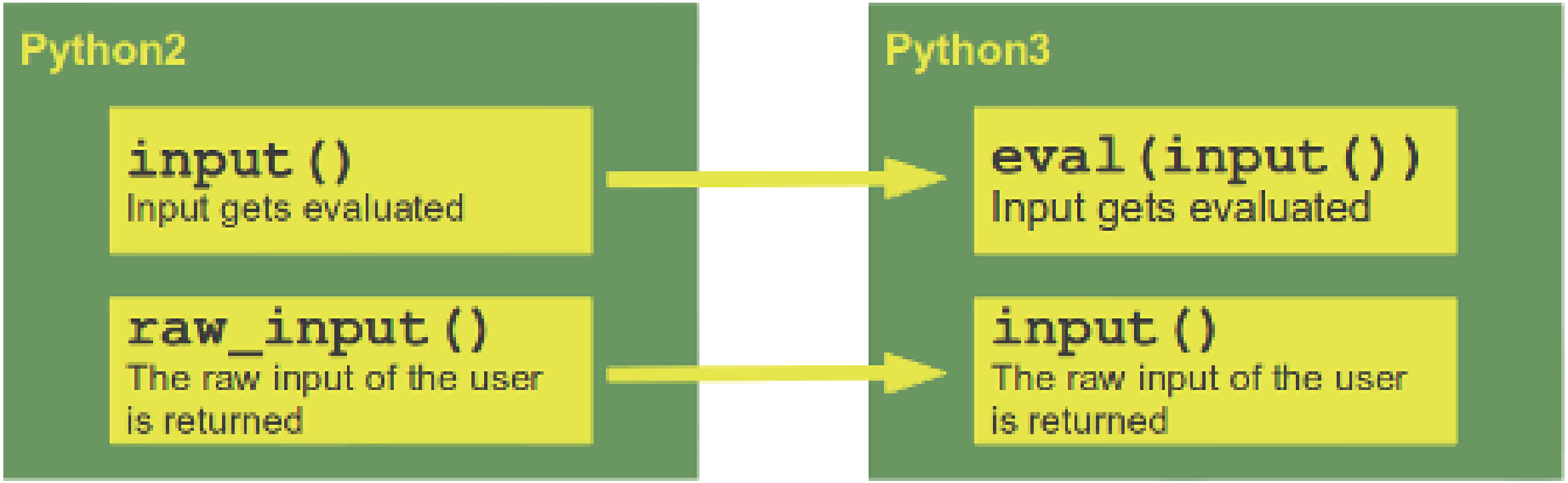
  - **Reťazce(stringy):**

```
meno2 = raw_input("Vloz meno: ")
print meno2
```

- **import** – include knižníc
  - import math

Ako sa po anglicky povie tlač a vstup?

Práca so vstupom



IT ACADEMY

Python + AI/ML

"We used machine learning algorithms to greet the user with a personalized message"

```
name = input()
print("Hello " + name)
```

Nepoužívaj globálne premenné!

2.5 Global variables

Avoid global variables.

2.5.1 Definition

Variables that are declared at the module level or as class attributes.

Lebo zhoríš v pekle!

You Retweeted

Jake VanderPlas @jakevdp · Mar 17

Fun fact: in Python 3, ಠ_ಠ is a valid identifier.

Use this power wisely...

```
[1]: ಠ_ಠ = "hmm..."
     print(ಠ_ಠ)

     hmm...
```

43 914 3K

Show this thread

The Best Linux Blog In the Unixverse @nixcraft · 9m

styleguide

Google Python Style Guide

▶ Table of Contents

1 Background

Python is the main dynamic language used at Google. This style guide is a list of *dos and don'ts* for Python programs.

To help you format code correctly, we've created a settings file for Vim. For Emacs, the default settings should be fine.

Many teams use the yapf auto-formatter to avoid arguing over formatting.

2 Python Language Rules

2.1 Lint

Run `pylint` over your code using this pylintrc.

2.1.1 Definition

`pylint` is a tool for finding bugs and style problems in Python source code. It finds problems that are typically caught by a compiler for less dynamic languages like C and C++. Because of the dynamic nature of Python, some warnings may be incorrect; however, spurious warnings should be fairly infrequent.

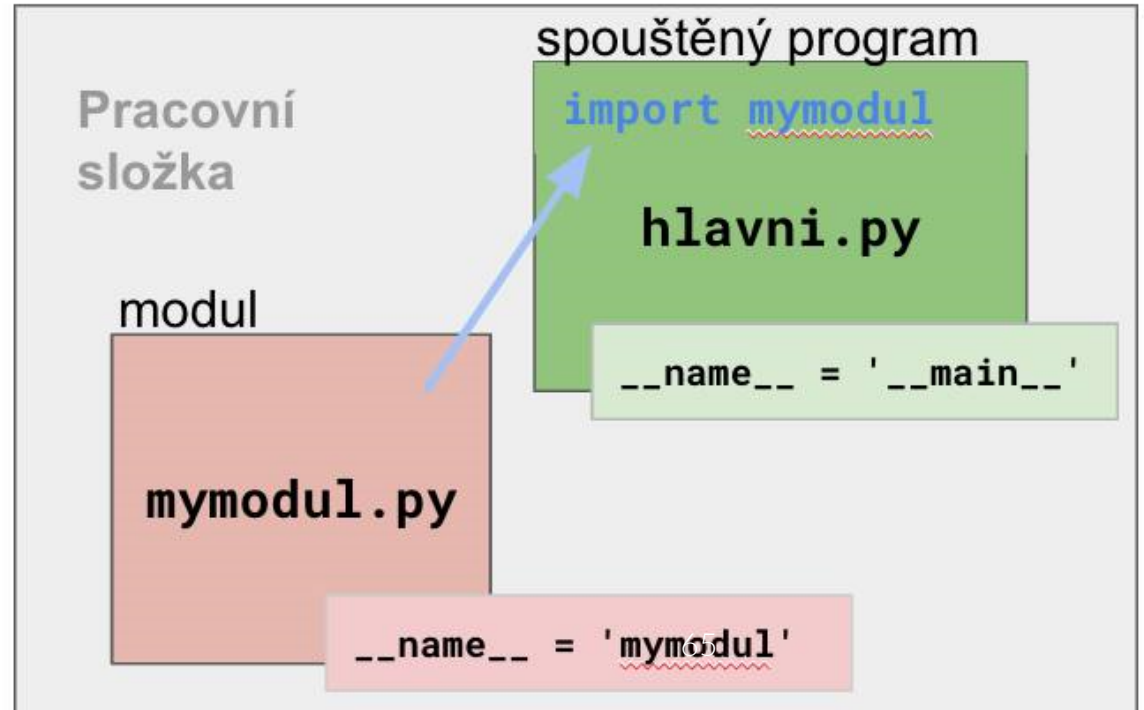
2.1.2 Pros

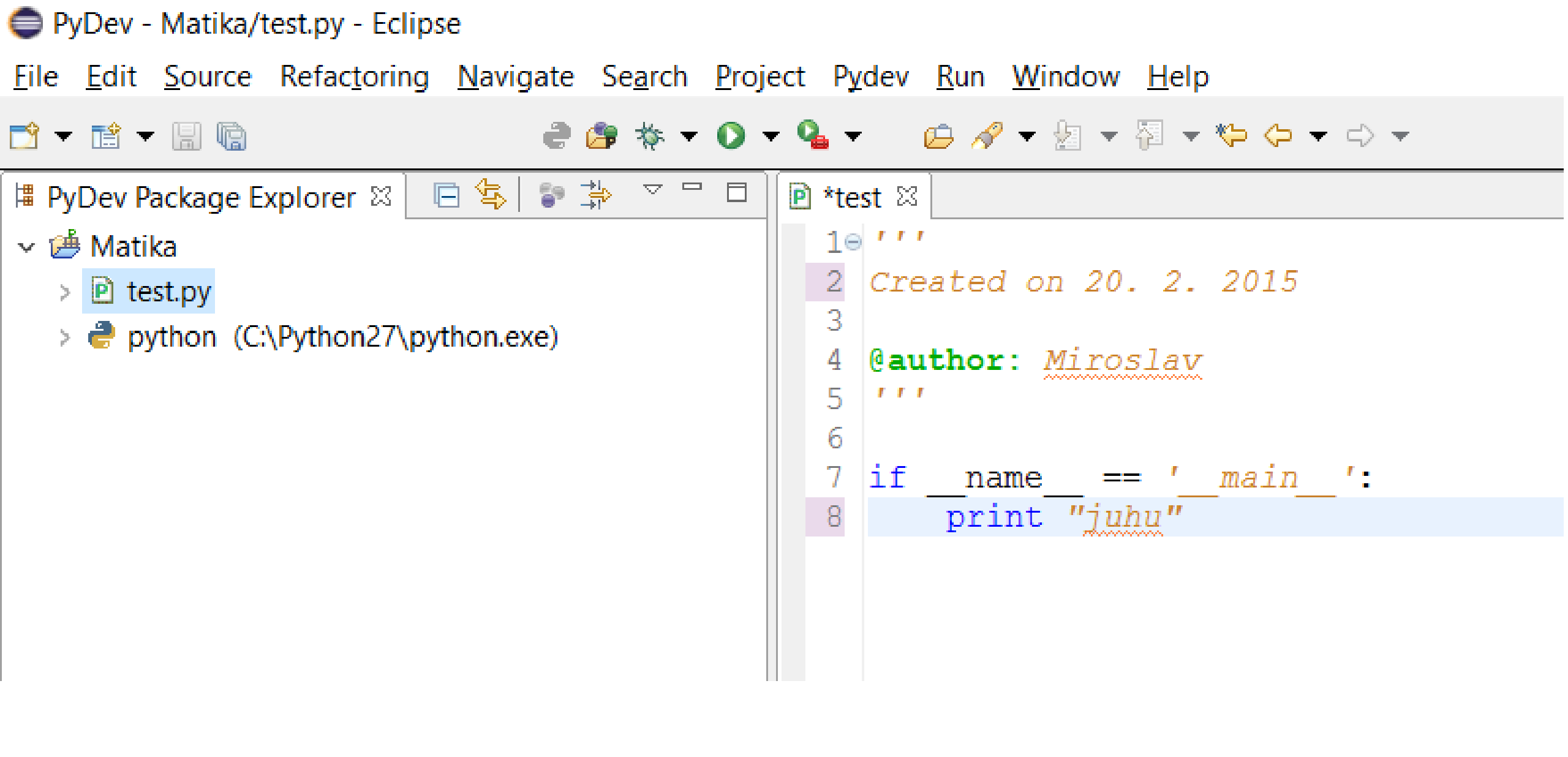
Moduly - '__main__' / __name__

```
### Hlavní program
def main():
    """ Main application """
    print ("Ahoj")

if __name__ == '__main__':
    """
    if the module is the main running
     module, run main() function
    """
    main()
```

Proměnná __name__
má hodnotu typu string obsahující:

- jméno modulu (je-li v načteném modulu)
- `'__main__'` (v hlavním,spuštěném, souboru)




PyDev - Matika/test.py - Eclipse
File Edit Source Refactoring Navigate Search Project Pydev Run Window Help
PyDev Package Explorer
Matika
test.py
python (C:\Python27\python.exe)
*test
1 '''
2 Created on 20. 2. 2015
3
4 @author: Miroslav
5 '''
6
7 if __name__ == '__main__':
8 print "juhu"

Operátor in a is (boolean)

```
meno = 'laco'
'a' in meno
'x' in meno

rodina = ['mama','otec','brat']
'sestra' in rodina
'mama' in rodina
```

```
>>> a = 'pub'
>>> b = ''.join(['p', 'u', 'b'])
>>> a == b
True
>>> a is b
False
```

Cyklus for

```
obchod=['chleba','mlieko','maso','maslo','ryza']

for jedlo in obchod:
    print "chcem " + jedlo
```

Prechádzanie cez zoznam

```
colors = ['red', 'green', 'blue', 'yellow']


for i in range(len(colors)):
    print colors[i]


for color in colors:
    print color
```

Has pseudocode gone too far?

must be stopped at all costs

Prechádzanie nazad cez zoznam

```
colors = ['red', 'green', 'blue', 'yellow']


for i in range(len(colors)-1, -1, -1):
    print colors[i]


for color in reversed(colors):
    print color
```

Prechádzanie cez zoznam a enumerovanie


```
colors = ['red', 'green', 'blue', 'yellow']


for i in range(len(colors)):
    print i, '-->', colors[i]


for i, color in enumerate(colors):
    print i, '-->', color
```

Prechádzanie cez 2 zoznamy

```
names = ['raymond', 'rachel', 'matthew']
colors = ['red', 'green', 'blue', 'yellow']


n = min(len(names), len(colors))
for i in range(n):
    print names[i], '-->', colors[i]


for name, color in zip(names, colors):
    print name, '-->', color


for name, color in izip(names, colors):
    print name, '-->', color
```

Prechádzanie cez zoradené dáta

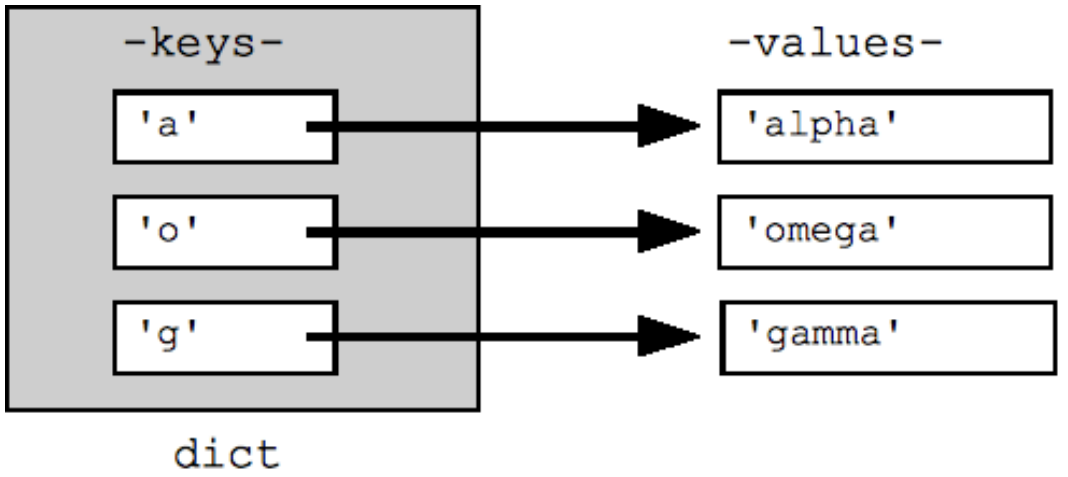
```
colors = ['red', 'green', 'blue', 'yellow']


for color in sorted(colors):
    print color


for color in sorted(colors, reverse=True):
    print color
```

Prechádzanie slovníkov

```
d = {'matthew': 'blue', 'rachel': 'green', 'raymond':
'red'}
```

```
for k in d:
    print k
```

```
for k in d.keys():
    if k.startswith('r'):
        del d[k]
```




```
d = {k : d[k] for k in d if not k.startswith('r')}
```

Prechádzanie slovníkov

```
for k in d:
    print k, '-->', d[k]
```

```
for k, v in d.items():
    print k, '-->', v
```

```
for k, v in d.iteritems():
    print k, '-->', v
```

Tvorba slovníkov z párov

```
names = ['raymond', 'rachel', 'matthew']
colors = ['red', 'green', 'blue']


d = dict(izip(names, colors))
{'matthew': 'blue', 'rachel': 'green', 'raymond': 'red'}
```

```
d = dict(enumerate(names))
{0: 'raymond', 1: 'rachel', 2: 'matthew'}
```

Počítame so slovníkmi

```
colors = ['red', 'green', 'red', 'blue', 'green', 'red']

d = {}
for color in colors:
    if color not in d:
        d[color] = 0
    d[color] += 1


{'blue': 1, 'green': 2, 'red': 3}

d = {}
for color in colors:
    d[color] = d.get(color, 0) + 1

d = defaultdict(int)
for color in colors:
    d[color] += 1
```

Rozbaľovanie sekvencií

```
p = 'Raymond', 'Hettinger', 0x30, 'python@example.com'


fname = p[0]
lname = p[1]
age = p[2]
email = p[3]


fname, lname, age, email = p
```

Aktualizácia viacerých stavov premenných


```
def fibonacci(n):
    x = 0
    y = 1
    for i in range(n):
        print x
        t = y
        y = x + y
        x = t
```

```
def fibonacci(n):
    x, y = 0, 1
    for i in range(n):
        print x
        x, y = y, x+y
```

Spájanie reťazcov

```
names = ['raymond', 'rachel', 'matthew', 'roger',
         'betty', 'melissa', 'judith', 'charlie']


s = names[0]
for name in names[1:]:
    s += ', ' + name
print s


print ', '.join(names)
```

Otváranie a zatváranie súborov

```
f = open('data.txt')
try:
    data = f.read()
finally:
    f.close()
```

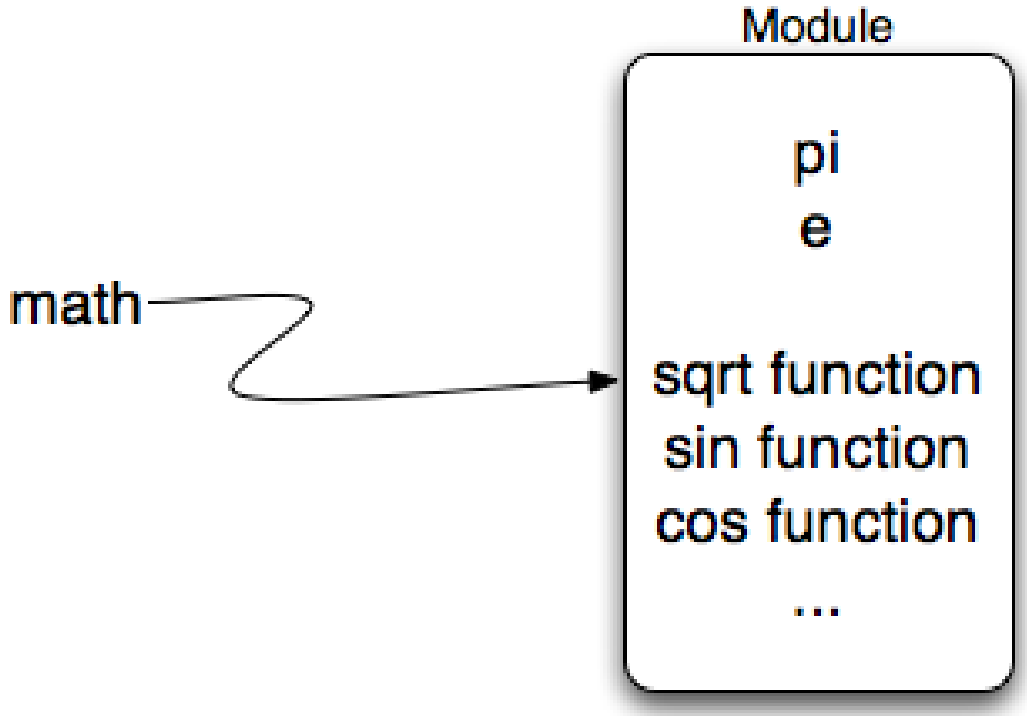
```
with open('data.txt') as f:
    data = f.read()
```

List comprehensions

```
result = []
for i in range(10):
    s = i ** 2
    result.append(s)
print sum(result)


print sum([i**2 for i in xrange(10)])


print sum(i**2 for i in xrange(10))
```

Práca s prvým modulom Math

Názov	Popis
acos(x)	inverzný kosínus
asin(x)	inverzný sínus
atan(x)	inverzný tangens
atan2(y,x)	rovnocenný zápisu *atan(y/x)*, ale efektívnejší
ceil(x)	najmenšie číslo integer väčšie alebo rovné *x*
cos(x)	kosínus *x*
cosh(x)	hyperbolický kosínus *x*
degrees(x)	konvertuje uhol v radiánoch na stupne
exp(x)	exponenciálna funkcia: *e* umocnené na *x*
fabs(x)	absolútna hodnota *x*
floor(x)	najväčšie číslo integer menšie alebo rovné *x*
fmod(x,y)	zvyšok po celočíselnom delení *x/y*
frexp(x)	vracia *n*-ticu (mantisa,exponent) tak, že *x=mantissa*(2**exponent)*, kde exponent je integer a *0.5 <=abs(m)<1.0*
hypot(x,y)	rovnocenný zápis je *sqrt(x*x+y*y)*
ldexp(x,y)	rovnocenný zápis je *x*(2**y)*
log(x)	prirodzený (základ *e*) logaritmus *x*
log10(x)	dekadický (základ *10*) logaritmus *x*
modf(x)	vracia zlomkovú a celočíselnú časť *x* ako *n*-ticu
pow(x,y)	*x* umocnené na *y*; rovnocenné zápisu *x**y*
radians(x)	konvertuje uhol v stupňoch na radiány
sin(x)	sínus *x*
sinh(x)	hyperbolický sínus *x*
sqrt(x)	druhá odmocnina z *x*
tan(x)	tangens *x*
tanh(x)	hyperbolický tangens *x*




```
import math
```

Modul sys

Almost every program uses the sys library

```
>>> import sys
>>> print sys.version
2.7 (r27:82525, Jul  4 2010, 09:01:59)
[MSC v.1500 32 bit (Intel)]
>>> print sys.platform
win32
>>> print sys.maxint
2147483647
>>> print sys.path
['',
 'C:\\WINDOWS\\system32\\python27.zip',
 'C:\\Python27\\DLLs', 'C:\\Python27\\lib',
 'C:\\Python27\\lib\\plat-win',
 'C:\\Python27', 'C:\\Python27\\lib\\site-packages']
```


sys.maxint len pre staršie verzie Python

Modules

All modules and their contents (functions, constants) can be found at
http://docs.python.org/modindex.html

```
>>> import math # mathematical functions
>>> dir(math)
['__doc__', '__name__', '__package__', 'acos', 'acosh', 'asin', 'asinh', 'atan',
'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp',
'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot',
'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'modf', 'pi', 'pow',
'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']

>>> math.pi  # a constant
3.141592653589793
>>> math.sqrt(16)  # a function
4.0
>>> math.log(8, 2)
3.0
```

4) fabs(x)

Return the absolute value of *x*

```
>>> fabs(-1.5)
1.5
```

5) fsum(iterable)

Return an accurate floating point sum of values in the iterable

```
>>> fsum([1,2,-5])
-2.0
```

6) factorial(x)

Return *x* factorial. Raises ValueError if *x* is not integer or is negative.

```
>>> factorial(4)
24
>>> factorial(-4)
Traceback (most recent call last):
ValueError: factorial() not defined for negative values
```

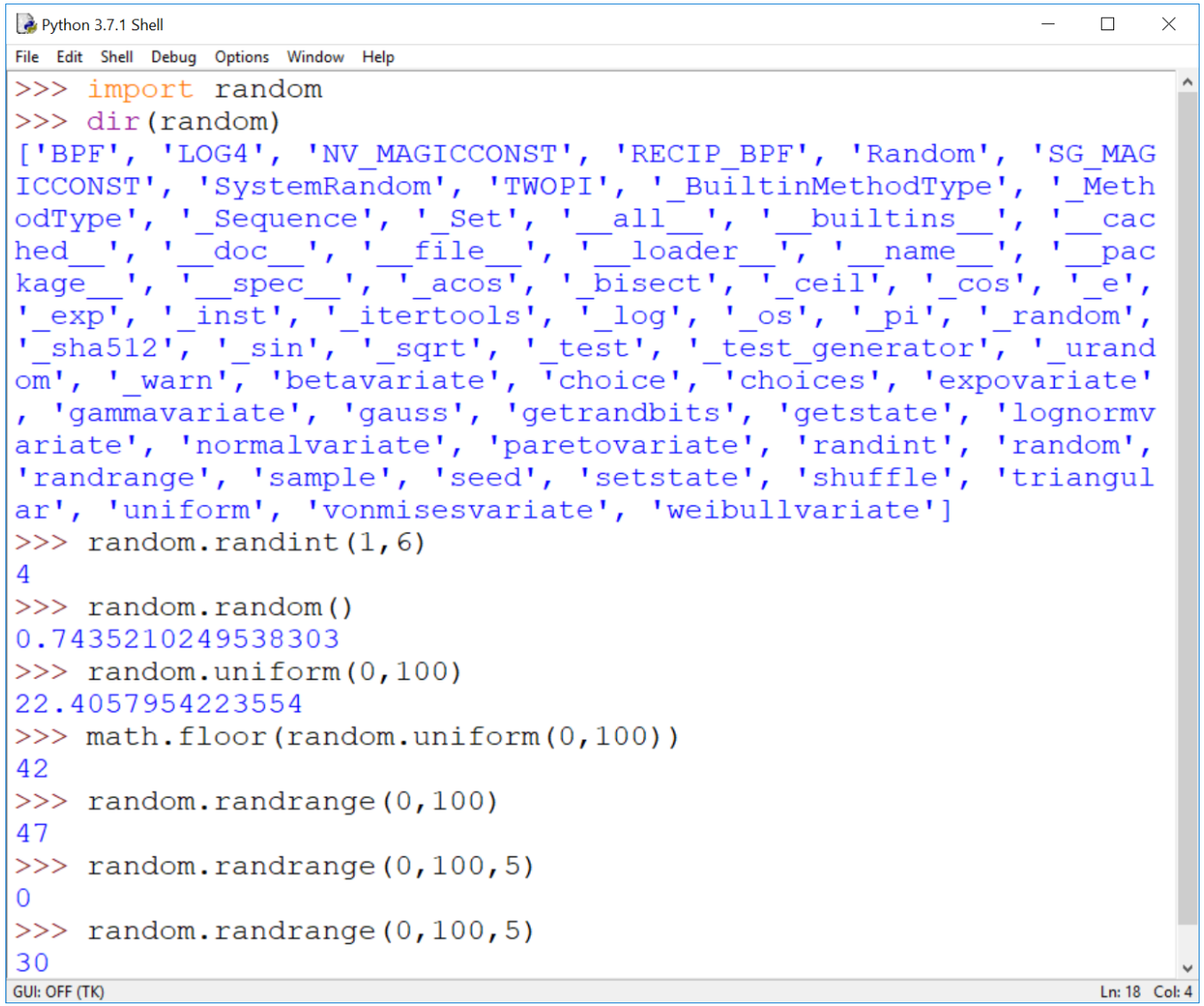
7) gcd(a, b)

Return the greatest common divisor of the integers *a* and *b*.

```
>>> gcd(10,25)
5
```

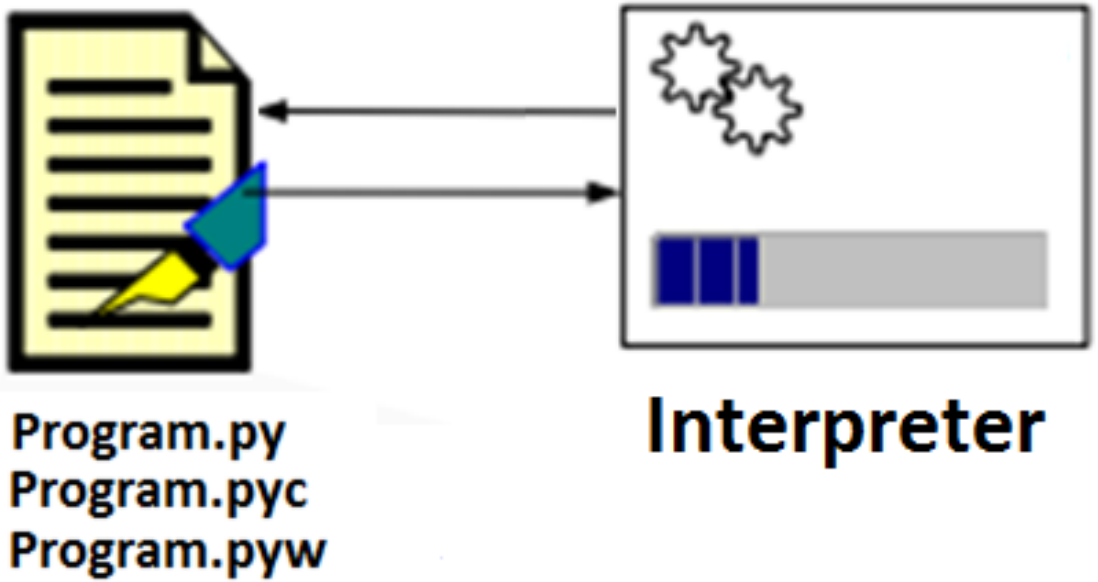
Fmod a modf

Modul random



IT ACADEMY

Python platforma



*.py – Zdrojový kód Python

*.pyc – kompilované verzie zdrojových súborov

*.pyw – Zdrojový kód Python GUI

IT ACADEMY

webs

IT ACADEMY

Práca s reťazcami

```
>>> print 'o' 'n' "e"
one
>>> print 't' r'\/\/' """o"""
t\/\/o
>>> text = ("Super extra dlhy string, "
        "pokracuje na novom riadku")
>>> text
'Super extra dlhy string, pokracuje na novom riadku'
>>> """Trojnásobné
dvojité
uvodzovky"""
'Trojn\xe1sobn\xe9\ndvojit\xe9\nuvodzovky'
```

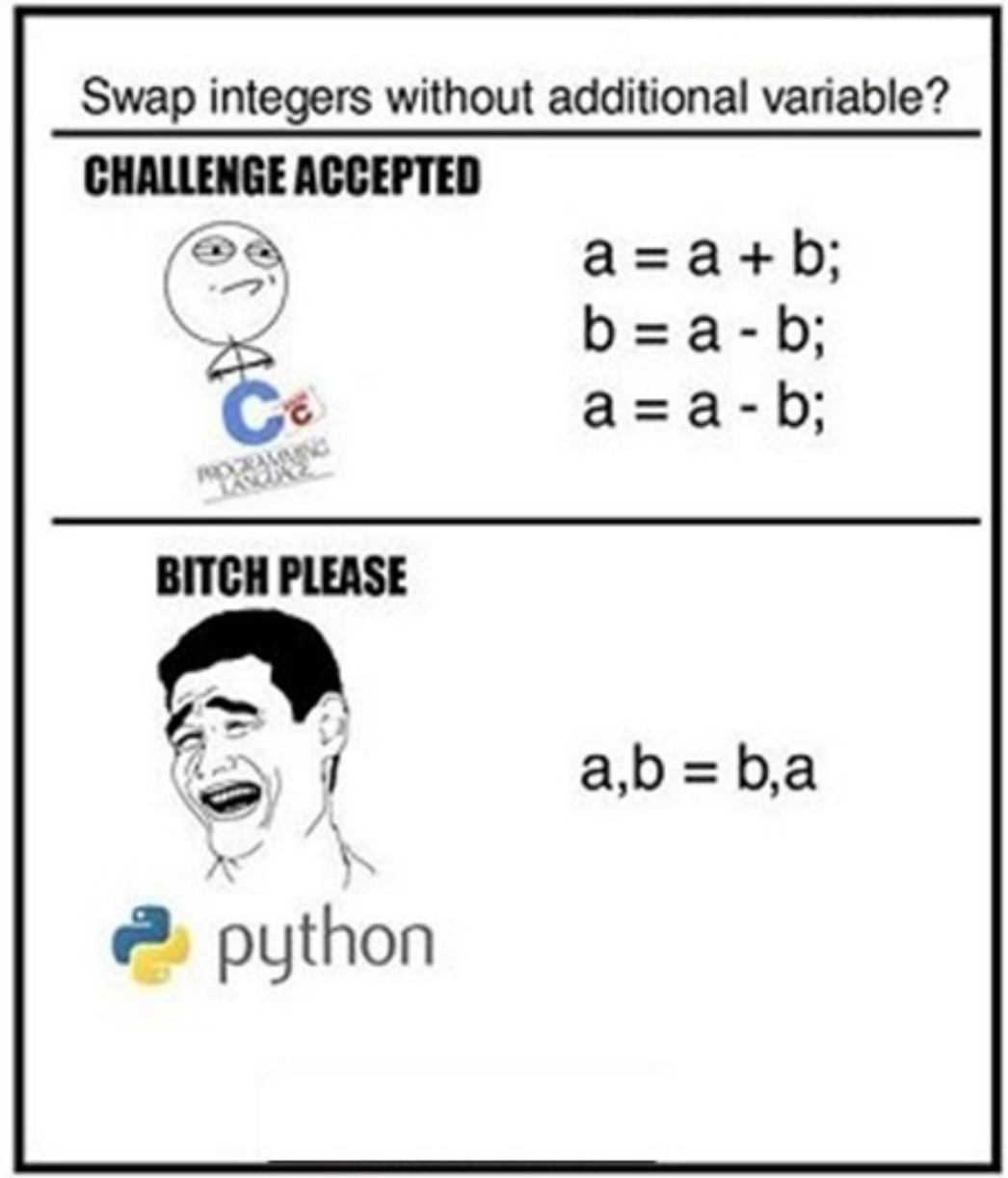
Komentáre a Docstringy

```
>>> #Docstringy | Tokeny | Action Itemy
>>> #!!! FIX:
>>> #!!! BUG:
>>> #!!! ???: Preco to tu je?
>>> #!!! NOTE: Nezabudni, ze...
>>> #!!! TODO: Sprav ...
>>> #!!! HACK:
>>> #!!! PENDING:
>>> #!!! XXX:
>>> #!!! OK:
```

IT ACADEMY

 # ALT + 35

```
>>> a = 1
>>> b = 2
>>> temp = a
>>> a = b
>>> b = temp
>>> a
2
>>> b
1
>>> temp
1
>>> b, a = a, b
>>> a
1
>>> b
```



```
>>> 1,
(1,)
>>> (1,)
(1,)
>>> (1)
1
>>> ()
()
>>> tuple()
()
>>> hodnota = 1,
>>> hodnota
(1,)
```

```
>>> # Tvorba Stringov zo Substringov
>>> farby = ["biela", "modra", "cervena"]
>>> # Takto nie
>>> vysledok = ""
>>> for s in farby:
        vysledok += s

>>> vysledok
'bielamodracervena'
>>> vysledok = ""
>>> vysledok = "".join(farby)
>>> vysledok
'bielamodracervena'
```

Prechádzanie zoznamom čísel

```
for i in [0, 1, 2, 3, 4, 5]:
    print i**2
```

```
for i in range(6):
    print i**2
```

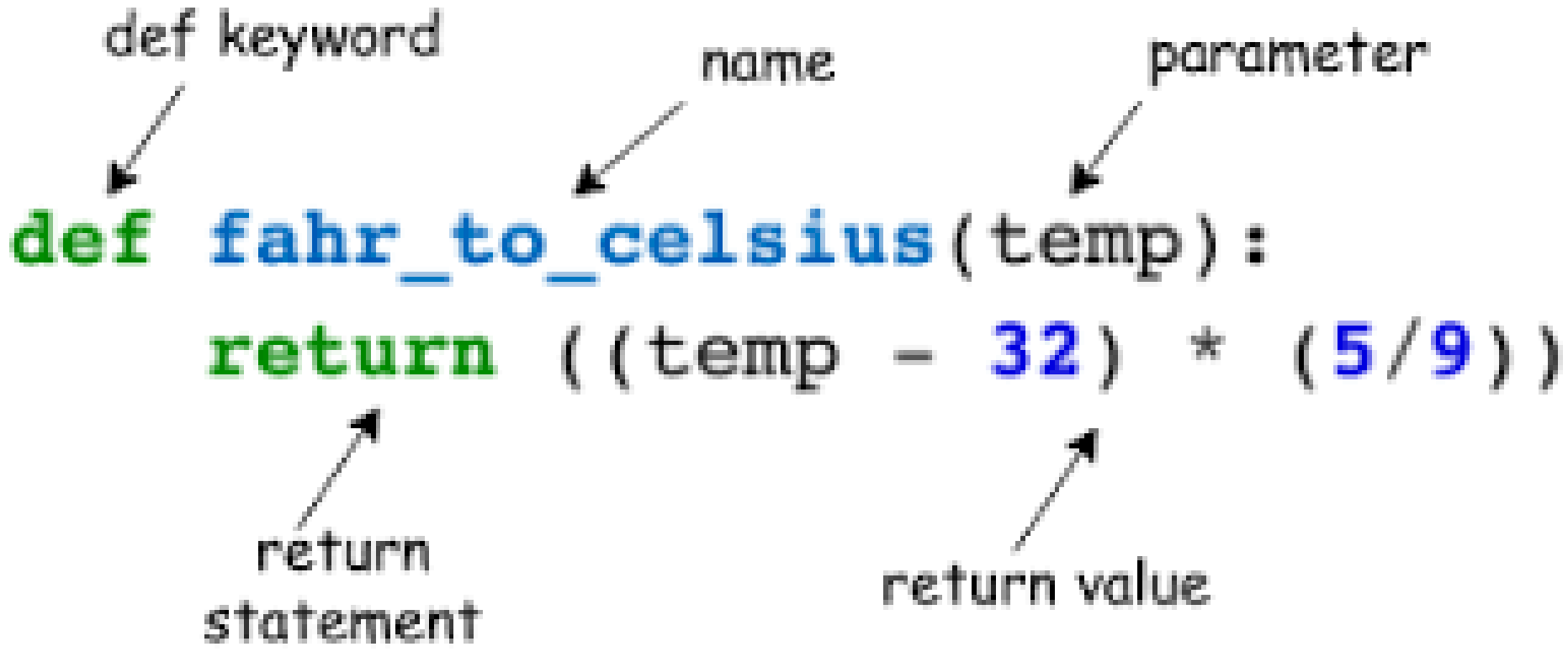
```
for i in xrange(6):
    print i**2
```

IT ACADEMY

Moduly pre dátových vedcov

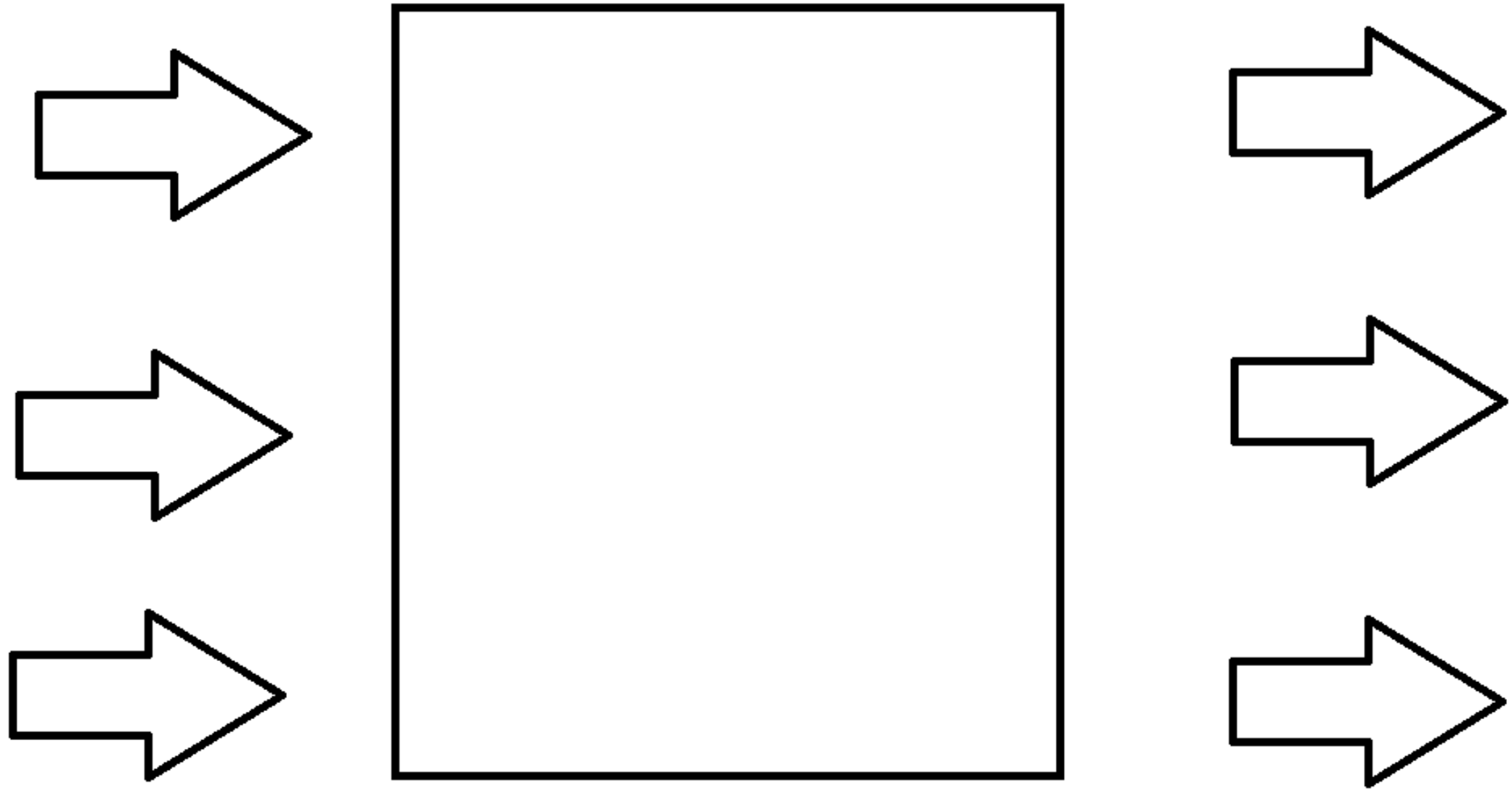
1. Numpy
2. Pandas
3. Matplotlib

Vlastné ~~funkcie metódy~~ definície



IT ACADEMY

Definicia/funkcia



Vlastné sortovanie

```
colors = ['red', 'green', 'blue', 'yellow']


def compare_length(c1, c2):
    if len(c1) < len(c2): return -1
    if len(c1) > len(c2): return 1
    return 0


print sorted(colors, cmp=compare_length)


print sorted(colors, key=len)
```

IT ACADEMY

Čo sa oplatí prečítať?

Slovensko a Česko

- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

Zahraničie

- O'Reilly
- Manning
- Packt
- Apress
- Wiley
- No Starch Press

YouTube tutorialy

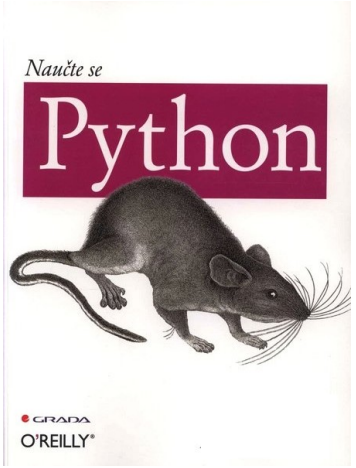
Packt Publishing

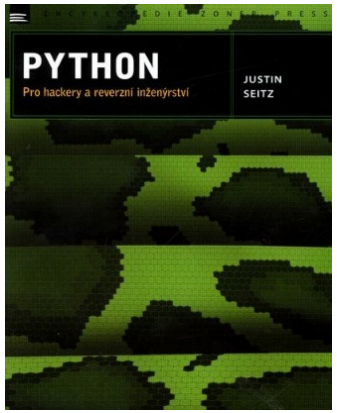
Čo ti odporúčam si pozrieť?

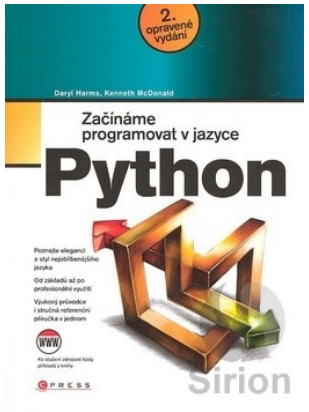
1. https://docs.python.org/3/
2. https://realpython.com/tutorials/best-practices/
3. https://google.github.io/styleguide/pyguide.html
4. https://docs.python.org/3/
5. http://python2013.input.sk/19prednaska
6. https://realpython.com/python3-object-oriented-programming/
7. https://jeffknupp.com/blog/2014/06/18/improve-your-python-python-classes-and-object-oriented-programming/
8. https://overiq.com/python-101/inheritance-and-polymorphism-in-python/
9. https://www.javatpoint.com/python-oops-concepts
10. https://www.programiz.com/python-programming/object-oriented-programming

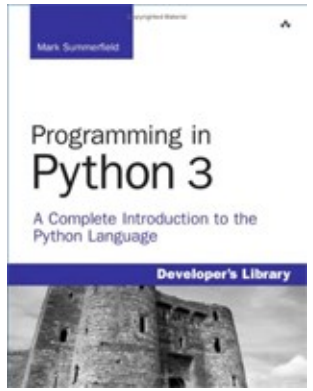
Šup do záložiek

Čo sa oplatí/neoplatí prečítať SK/CZ?









Mark Pilgrim

IT ACADEMY

Home

PUBLIC
Questions
Tags
Users

COLLECTIVES
Explore Collectives

FIND A JOB
Jobs
Companies

TEAMS
Create free Team

Tags

A tag is a keyword or label that categorizes your question with other, similar questions. Using the right tags makes it easier for others to find and answer your question.

Show all tag synonyms

python

Popular | Name | New

python

Python is a multi-paradigm, dynamically typed, multi-purpose programming language. It is designed to be quick to learn, understand, and...

1870168 questions — 695 asked today, 6387 this week

python-3.x

USE ONLY IF YOUR QUESTION IS VERSION-SPECIFIC. For questions about Python programming that are specific to version 3+ of the language...

303562 questions — 93 asked today, 836 this week

python-2.7

Python 2.7 is the last major version in the 2.x series, and is no longer maintained since January 1st 2020. Use the generic [python] tag on all Python...

94965 questions — 24 asked this week, 106 this month

python-requests

USE ONLY FOR THE PYTHON REQUESTS LIBRARY. Requests is a full-featured Python HTTP library with an easy-to-use, logical API.

18697 questions — 8 asked today, 57 this week

python-imaging-library

The Python Imaging Library (PIL) provides the Python language with a de-facto standard foundation for image work. PIL's API is lightweight but...

7883 questions 5 asked today, 38 this week

wxpython

wxPython is a Python wrapper for the cross-platform C++ GUI API wxWidgets.

7047 questions — 7 asked this week, 14 this month

ipython

IPython is a feature-rich interactive shell for Python, and provides a kernel for frontends such as IPython Notebook and Jupyter Notebook.

6886 questions — 5 asked this week, 26 this month

python-3.6

Version of the Python programming language released in December 2016. For issues specific to Python 3.6. Use more generic [python] and [python-3....

5602 questions — 11 asked this week, 24 this month

python-asyncio

to be used for the asyncio Python package which provides mechanisms for writing single-threaded concurrent code. The asyncio package provides...

5492 questions — 29 asked this week, 125 this month

python-import

For questions about importing modules in Python

5119 questions — 11 asked this week, 47 this month

python-multiprocessing

multiprocessing is a package that supports spawning processes using an API similar to the threading module in python programming language.

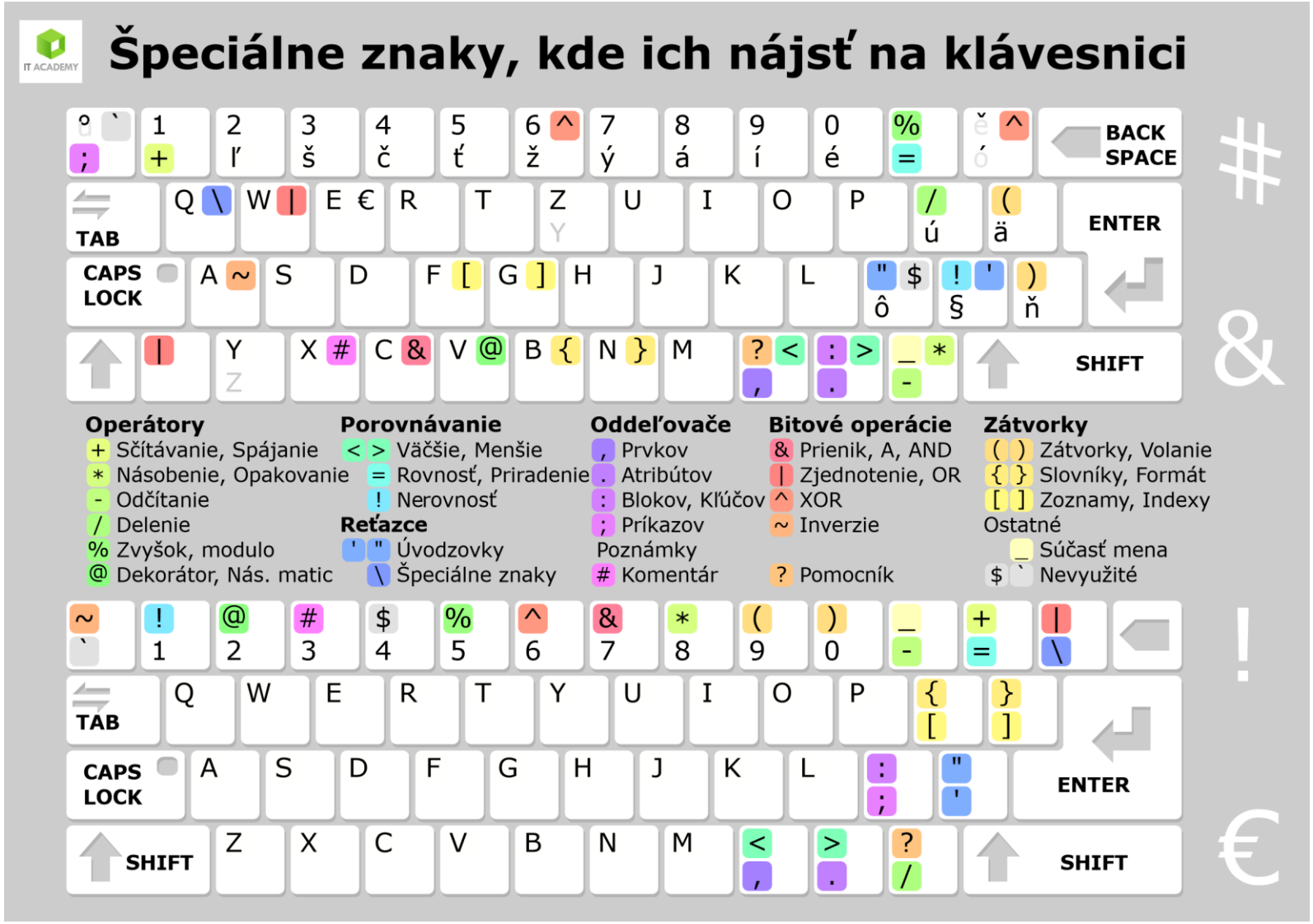
4036 questions — 12 asked this week, 46 this month

python-3.7

Version of the Python programming language released in June 27, 2018. For issues that are specific to Python 3.7. Use the more generic [python] and...

4034 questions — 5 asked this week, 21 this month

Efektívne používanie klávesnice



Najdôležitejšie klávesové skratky

Práca s IDE

- Ctrl + D — Delete zmaž riadok
- **Ctrl + Space** — Asistent kódu
- **Ctrl + /** — Komentáre
- Ctrl + A — Označ všetko
- **Alt + /** — Dokonči slovo
- Ctrl + F — Hľadanie a náhrady
- Ctrl + Shift + F — Kompakt režim
- Ctrl + Shift + S — Ulož všetko

Práca s browserom

- Ctrl + T — Vytvor nový tab
- Ctrl + W — Zatvor aktuálny tab
- Ctrl + Shift + W — Zatvor všetky taby
- **Ctrl + Shift + T** — Otvor posledný tab
- Ctrl + Shift + J/F12 — Web console
- **F11** — Fullscreen

IT ACADEMY

F5 nie je spustenie, ale Rehresh


NOVINKY
PROGRAM
EDUSUMMIT
SPONZORI
KÓDEX SPRÁVANIA
ABOUT
KÚP SI LÍSTOK
PYCON SK 2022
Bratislava
KÚP SI LÍSTOK

Pyonieri - Python SR & ČR

Verejná skupina · 5,5 tis. členov

Člen · Pozvať

Informácie · Diskusia · Vybrané · Témy · Ľudia · Podujatia · Médiá · Súbory

Napíšte niečo...

Živé video · Fotka/video · Anketa

Vybrané

Honza Javorek
28. januára 2015 ·

Vítej ve skupině pro všechny, které baví programovací jazyk Python a rozumí slovensky 🇸🇰 nebo česky 🇨🇿. (See bottom of this post for English 🇬🇧 intro)

Etiketa

Pravidlá

Here's what members group.

1 Než se zeptáš, r

2 Pracov dk

Informácie

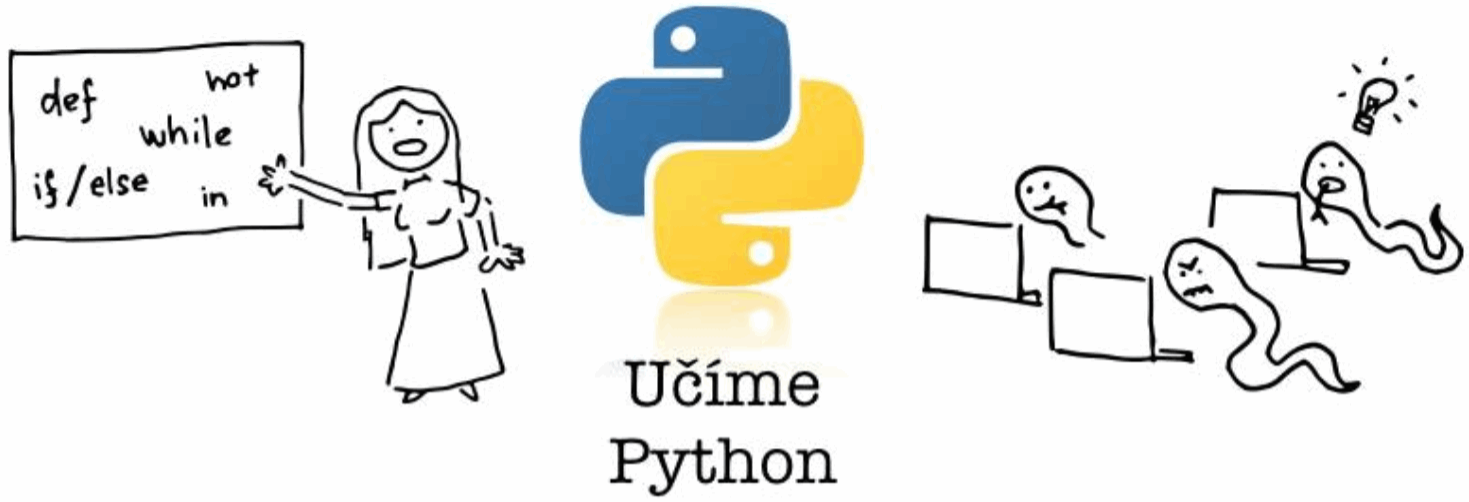
Skupina pre slovenských a českých milovnikov programovacieho jazyka Python.

Verejná
Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná
Túto skupinu nájde ktokoľvek

Česká republika · Slovensko

Všeobecné



Učíme Python

Verejná skupina · 992 členov

Člen · + Pozvať

Informácie · Diskusia · Vybrané · Témy · Ľudia · Podujatia · Médiá

Napíšte niečo...

Živé video · Fotka/video · Anketa

Vybrané

Honza Javorek
10. februára 2016 ·

Skupina pro kouče/učitele/mentory

Informácie

Skupina pro kouče/učitele/mentory proramovacího jazyka Python a pro ty, kteří je shánějí.

E-mailová skupina: https://groups.google.com/d/forum/ucime-python

Verejná
Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná

Data Analysts, Data Engineers & Data Scientists - Czech&Slovak Group

Verejná skupina · 2,6 tis. členov

Člen · + Pozvať

Informácie · Diskusia · Vybrané · Témy · Ľudia · Podujatia · Médiá · Súbory

Napíšte niečo...

Živé video · Fotka/video · Anketa

Vybrané

Informácie

Dataflow.cz Dataflow
Twitter: https://twitter.com/dataflowcz
Web: http://dataflow.cz... Zobraziť viac

Verejná
Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná

Skupina stránky Fred Nora

Python

Verejná skupina · **528,2 tis. členov**

Pridať sa do skupiny

Informácie | Diskusia | Príručky | Vybrané | Témy | Ľudia | Podujatia | Médiá | Súbory

Napíšte niečo...

Fotka/video | Anketa

Vybrané

Fred Nora aktualizoval opis.
8. júna 2020 ·

This is a group for Python

Pravidlá
Here's what members
group.

Informácie

This is a group for Python developers.

Verejná
Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná
Túto skupinu nájde ktokoľvek

Texas · Londrina · California

Vývojári

Miroslav | Domov | Vytvoriť

Vývojári

Verejná skupina

- Informácie
- **Diskusia**
- Oznámenia
- Členovia
- Podujatia
- Videá
- Fotky
- Súbory

Hľadať v tejto skupine

Skratky

- Podnikanie na Slove… 2
- UK Manazment Externe…
- Testovacia firma
- VITA - Virtual It Academy
- Startupisti 2
- Rubyslava 2
- Vývojári

Zobraziť viac

Ste člen | Upozornenia | Zdieľať | Viac

Napísať príspe… | Pridať fotku/vi… | Živé video | Viac

Napíšte niečo...

Fotka/video | Divácka párty | Označiť priat… | •••

NOVÁ AKTIVITA

Roland Mondek
10 h

POZVAŤ ČLENOV

Zadajte meno alebo e-mailovú adresu...

ČLENOVIA 5 505 členov

POPIS

Skupina softvérových vývojárov. Táto skupina by mala byť miestom... Zobraziť viac

TYP SKUPINY

Všeobecné

VAŠE STRÁNKY

- IT Academy
- VITA - Virtual It Academy

KONTAKTY

- Evka Rybárska
- Jarmila Palenčárová
- Stefan Orosi
- Ivana Ivka Jasaňov
  Hrá Word Blitz
- Ivana Pavlíková
- Martin Vanko
- Lucia Kovačičová 4 h
- Lošák Filip
- Andrej Nejedlik
- Gabika Zubrikova

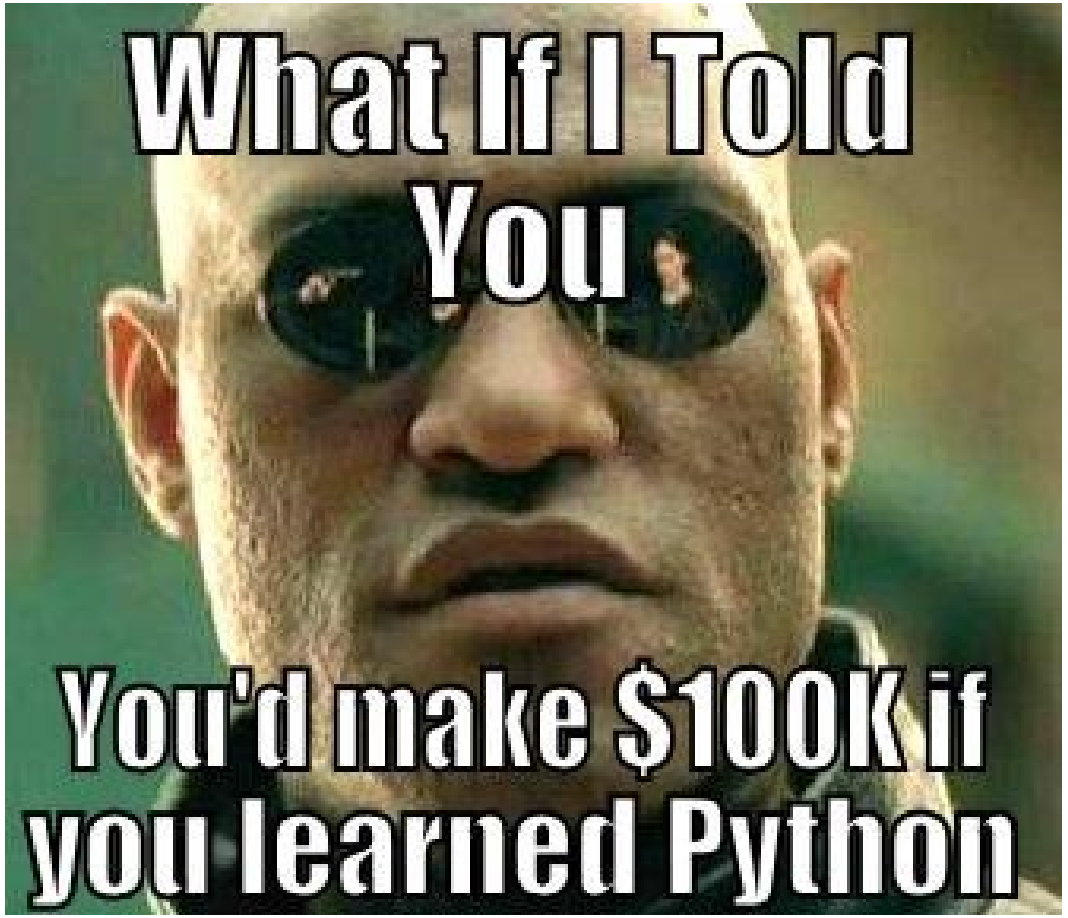
SKUPINOVÉ KONVERZÁCIE

Vytvoriť novú skupinu

Hľadať

Zen filozofia Pythonu

1. Krásny je lepší než škaredý
2. Explicitný je lepší ako implicitný
3. Jednoduchý je lepší ako zložitý
4. Zložitý je lepší ako komplikovaný
5. Plochý je lepší ako vnorený
6. Riedky je lepší ako hustý
7. Na čitateľnosti záleží
8. Praktickosť vyhráva nad čistotou




```
import this
```

Čaká nás krásna budúcnosť

```
>>> from __future__ import braces
SyntaxError: not a chance (<pyshell#13>, line 2)
>>>
```

IT ACADEMY

No future {} a ;

Mrkni na náš YouTube kanál a daj odber

➡ WWW.YOUTUBE.COM/C/IT-ACADEMYSK ⬅

IT ACADEMY

30 PROJECT IDEAS

1) To-do list app	2) Note taking app	3) Calendar application	4) Chat system	5) Weather application
6) Portfolio website	7) Image search	8) Chess game	9) Donation website	10) Budget tracker
11) Tic tac toe game	12) Form validator	13) Web scraper	14) Simple FTP client	15) Port scanner
16) MP3 player	17) Tetris game	18) Netflix clone	19) Discord bot	20) Video chat system
21) Pacman game	22) Alarm clock	23) Stock trading app	24) Issue tracker	25) Music store app
26) Twitter bot	27) Spam classifier	28) Content aggregator	29) Snake game	30) File manager

python Cheat Sheet

Math

Number Theoretic

ceil(x) copysign(x,y)
fabs(x) factorial(x)
floor(x) fmod(x,y)
frexp(x) fsum(iterable)
isinf(x) isnan(x)
ldexp(x,i) modf() trunc()

Power & Logarithmic

exp(x) log(x[, base])
log1p(x) log10(x)
pow(x,y) sqrt(x)

Trigonometric Functions

acos(x) asin(x) atan(x)
atan2(y,x) cos(x) hypot(x,y)
sin(x) tan(x)

Angular Conversion

degrees(x) radians(x)

Hyperbolic Functions

acosh(x) asinh(x)
atanh(x) cosh(x)
sinh(x) tanh(x)

Constants

math.pi
The mathmatical constant of pie = 3.141592.... up to the available precision
math.e
The mathmatical constant e = 2.718281.... up to the available precision

String Formatting

Formatting Operations

'd' Signed integer decimal 'i' Signed integer decimal 'o' Signed octal value 'u' Obsolete type - it was identical to 'd'
'x' Signed hexadecimal (lowercase) 'X' Signed hexidecimal (uppercase) 'e' Floating point exponential format (lowercase)
'E' Floating point exponential format (uppercase) 'f' Floating point decimal format 'F' Floating point decimal format
'g' Floating point format. Uses the lowercase exponential format if the exponent is less than -4 or not less than precision, otherwise it uses the decimal format
'G' Floating point format. Uses the uppercase exponential format if the exponent is less than -4 or not less than precision, otherwise it uses the decimal format
'c' Single character (accepts either integer or single character string) 'r' String (converts any Python object using repr())
's' String (converts any Python object using str()) '%' No argument is converted, adds a % character in the end result

OS

OS Variables

altsep Alternative separator
curdir Current dir string
defpath Default search path
devnull Path of null device
extsep Extension separator
pardir Parent dir string
pathsep Patch separator
sep Path separator
name name of OS
linesep Line separator

SYS

platform Current platform
stdin, stdout, stderr File objects for I/O
version_info Python version info
winver Version number

SYS Variables

argv Command line args
builtin_module_names Linked C modules
check_-interval Signal check frequency
exec_prefix Root directory
executable Name of Executable
exitfunc Exit function name
modules Loaded modules
path Search path

SYS Arg V

sys.argv[0] foo.py
sys.argv[1] bar
sys.argv[2] -c
sys.argv[3] qux
sys.argv[4] --h

File

Methods

close() flush() fileno()
isatty() next() read([size])
readline ([size]) readlines([size])
xreadlines() seek(offset[, whence])
tell() truncate([size])
write(str) writelines(sequence)

Attributes

closed encoding
errors mode
name newlines
softspace

Class

Special Methods

__new__(cls) __lt__(self, other) __init__(self, args)
__le__(self, other) __del__(self) __gt__(self, other)
__repr__(self) __ge__(self, other) __str__(self)
__eq__(self, other) __cmp__(self, other)
__ne__(self, other) __index__(self) __nonzero__(self)
__hash__(self) __getattr__(self, name)
__getattribute__(self, name) __setattr__(self, name, attr)
__delattr__(self, name) __call__(self, args, kwargs)

String

String Methods

capitalize() center(width[, fillchar]) count(sub[, start[, end]])
decode encode([encoding[, errors]]) isalnum()
endswith(suffix[, start[, end]]) expandtabs([tabsize])
find(sub[, start[, end]]) format(*args, **kwargs) isalpha()
index(sub[, start[, end]]) isdigit() islower() isspace() istitle()
isupper() join(iterable) ljust(width[, fillchar]) lower()
lstrip([chars]) partition(sep) replace(old, new[, count])
rfind(sub[, start[, end]]) rindex(sub[, start[, end]])
rjust(width[, fillchar]) rpartition(sep) rsplit([sep[, maxsplit]])
rstrip([chars]) split([sep[, maxsplit]]) splitlines([keepends])
startswith(prefix[, start[, end]]) strip([chars]), swapcase, title()
translate(table[, deletechars]), upper() zfill(width)
isnumeric() isdecimal()

Set & Mapping

Set Types

len(s) x in s x not in s isdisjoint(other)
issubset(others) issuperset union(other...)
intersection(other...) difference(other...)
symmetric_difference(other) copy() update()
intersection_update() difference_update()
symmetric_difference_update() add(elem)
remove() discard(elem) pop() clear()

Mapping Types

len(d) d[key] d[key]=value
del d[key] key in d key not in d
iter(d) clear() copy() items()
fromkeys(seq[, value]) keys()
get(key[, default]) has_key(key)
iteritems() iterkeys()
itervalues() popitem()
pop(key[, default])
setdefault(key[, default])
update([other])
values

Random

Functions

seed([x]) getstate() vonmisesvariate(mu,kappa)
setstate(state) jumpahead(n) paretovariate(alpha)
getrandbits(k) randint(a,b) weibullvariate(alpha,beta)
randrange([start], stop[, step]) lognormvariate(mu,sigma)
choice(seq) shuffle(x[, random]) normalvariate(mu, sigma)
sample(population,k) random() gammevariate(alpha,beta)
uniform(a,b) triangular(low,high,mode) gauss(mu,sigma)
betavariate(alpha,beta) expovariate(lambd)

Array

Array Methods

append(x) buffer_info()
byteswap() count(x)
extend(iterable) fromfile(f,n)
fromlist(list) fromstring(s)
fromunicode(s) index(x)
insert(i,x) pop([i]) remove(x)
reverse() tofile(f) tolist()
tostring() tounicode()

Indexes & Slices

a=[0,1,2,3,4,5]
b=a[:] Shallow copy of a
a[1:] [1,2,3,4,5]
a[5:] [0,1,2,3,4]
a[-2:] [0,1,2,3]
a[1:3] [1,2]
a[1:-1] [1,2,3,4]

len(a) 6
a[0] 0
a[5] 5
a[-1] 5
a[-2] 4

Date Time

Time Object

replace([hour[, minute[, second[, microsecond[, tzinfo]]]]])
isoformat() __str__() strftime() utcoffset() dst() tzname()

Datetime Object

date() time() timetz() toordinal() weekday() isoweekday() isocalendar()
replace([year[, month[, day[, hour[, minute[, second[, microsecond[, tzinfo]]]]]]]])
astimezone(tz) utcoffset() dst() tzname() timetuple() utctimetuple()
isoformat() __str__() ctime() strftime()

Date Object

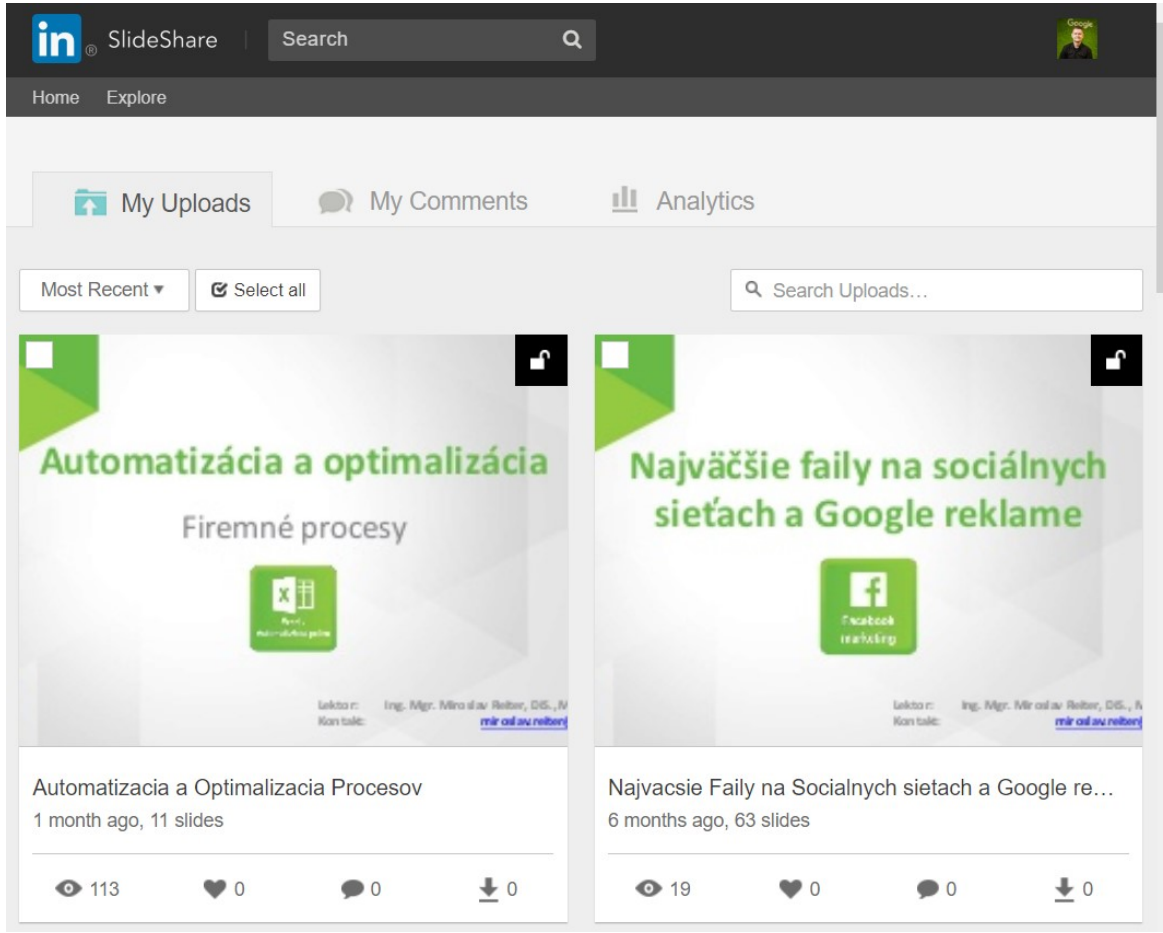
replace((year,month,day)) timetuple()
toordinal() weekday()() isoweekday()()
isocalendar()() isoformat() __str__()
ctime() strftime()

Date Formatting

%a Abbreviated weekday (Mon) %A Weekday (Monday)
%b Abbreviated month name (Nov) %B Month name (November)
%c Date and time %d Day (leading zeros) (01 to 31)
%H 24 hour (leading zeros) (00 to 23) %I 12 hour (leading zeros) (01 to 12)
%j Day of year (001 to 366) %m Month (01 to 12) %M Minute (00 to 59)
%p AM or PM %S Second (00 to 61?) %U Week number1 (00 to 53)
%w Weekday2 (0 to 6) %W Week number3 (00 to 53) %x Date
%X Time %y Year without century (00 to 99) %Y Year (2016)
%Z Time zone (EST) %% A literal "%" character (%)

Ako skončíme?

1. **Pridaj** si ma na **LinkedIn**
   - www.linkedin.com/in/miroslav-reiter
2. **Materiály po prednáške**
   - https://1drv.ms/p/s!AlrLrycbTQ1a19sfc1MmNNnYMaluWA?e=FTUlTc



IT ACADEMY

Grow with Google

Pomáhame Slovensku rásť

Našim záväzkom je **prispieť k oživeniu slovenskej ekonomiky** a do konca roka 2021

pomôcť ďalším 50,000 slovenským firmám a jednotlivcom

lepšie využívať internet k rastu svojho podnikania, rozvoja kariéry či nájdenia novej práce.

Online marketing	Technology & Tools	Business & Soft skills
Google Analytics	Workspace (G Suite)	Entrepreneurship
Online marketing strategy	Online Security & Safety	Leadership
Google for Nonprofits	Google for Education	Export
Shopping	AI/ML	#IamRemarkable*
YouTube		Entrepreneurship / Diversity for women
Google My Business		Critical thinking / Media literacy
Google Ads		

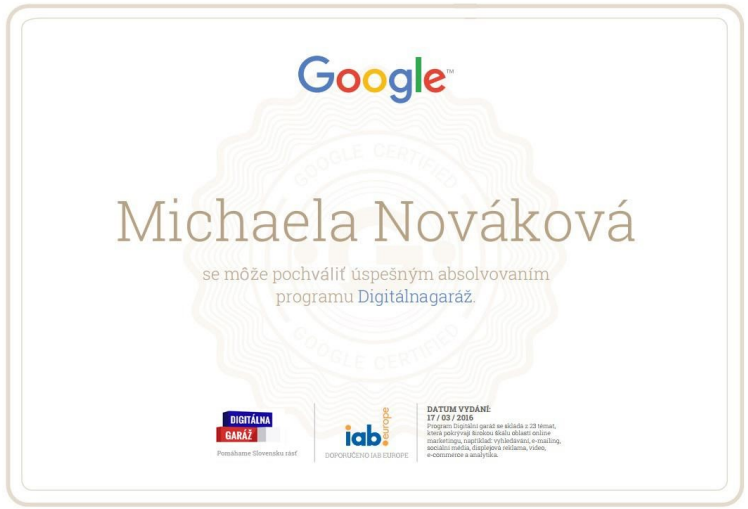
* Viac informácii na www.iamremarkable.sk

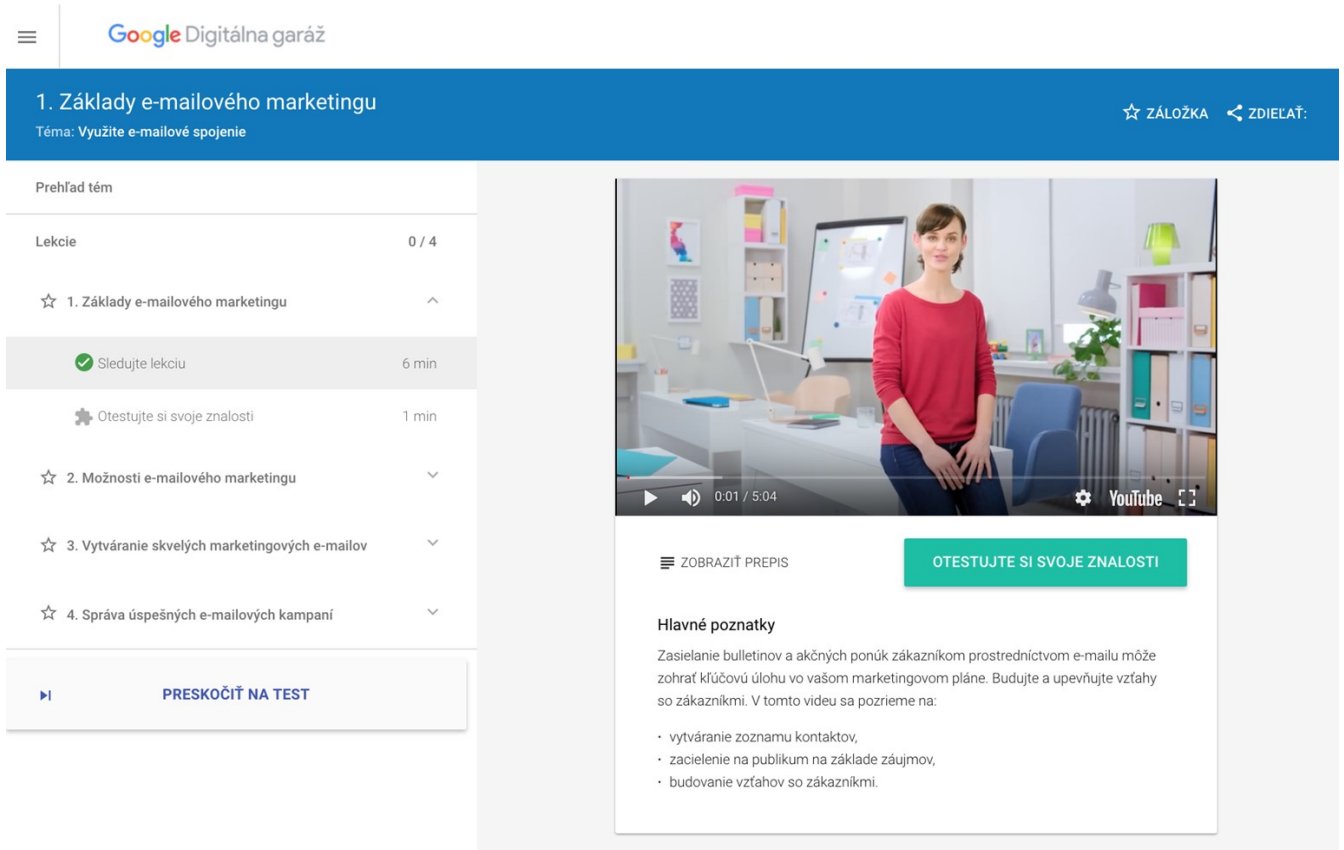
Google Digitálna garáž

Online platforma na výuku digitálneho marketingu a mäkkých zručností

26 tém | **106 lekcií** | **9 hodín obsahu**

Dostupné 24/7
Zadarmo
Na mieru





ITKurzy

IT kurzy CSČ SAV, v.v.i. - VS SAV

Domov | Ponuka kurzov | Ponuka prednášok

Domov » Python - Objektové programovanie

Python - Objektové programovanie

Kurz sa zameriava na pokročilejšie techniky práce s programovacím jazykom Python. Nosnou témou kurzu bude **objektovo orientované programovanie (OOP)**.

Zoznámime sa so základnými princípmi OOP:

1. **Dedičnosť**
2. **Zapuzdrenie**
3. **Polymorfizmus**

Ukážeme si ako v Pythone vytvoriť triedu, inštanciu na danú triedu, ako s nimi pracovať a využívať ich pri tvorbe programu. Zistíme načo slúži v triede konštruktor danej triedy, ako sa vyžíva pri dedičnosti tried a polymorfizme. Ukážeme si rôzne spôsoby dedičnosti, ich výhody a nevýhody v rôznych situáciách. Kurz predstavuje ucelený vstup do objektovo orientovaného programovania a oboznámenie sa so základnými pojmami tohto prístupu v prostredí jazyka Python.

Trvanie:
10 hodín (2 dni)
Cena:
€0,-
Kategória:
Python
Registrácia (počet prihlásených):
27. - 28. 4. 2022 (22/50)

Zadarmo

1. Kurzy SAV
2. Kurzy Grow with Google
3. YouTube kanál IT Academy

Platené

Moje kurzy na www.vita.sk

https://www.youtube.com/c/IT-AcademySK