

Štandardná Knižnica

Python



Lektor: RNDr. PhDr. Ing. Miroslav Reiter, DiS., MBA, DBA
Kontakt: riaditel@itacademy.sk

Ako Začneme?

1. Pridajte si ma na **LinkedIn**

www.linkedin.com/in/miroslav-reiter

2. Stiahnite si Cvičný NTB Súbor

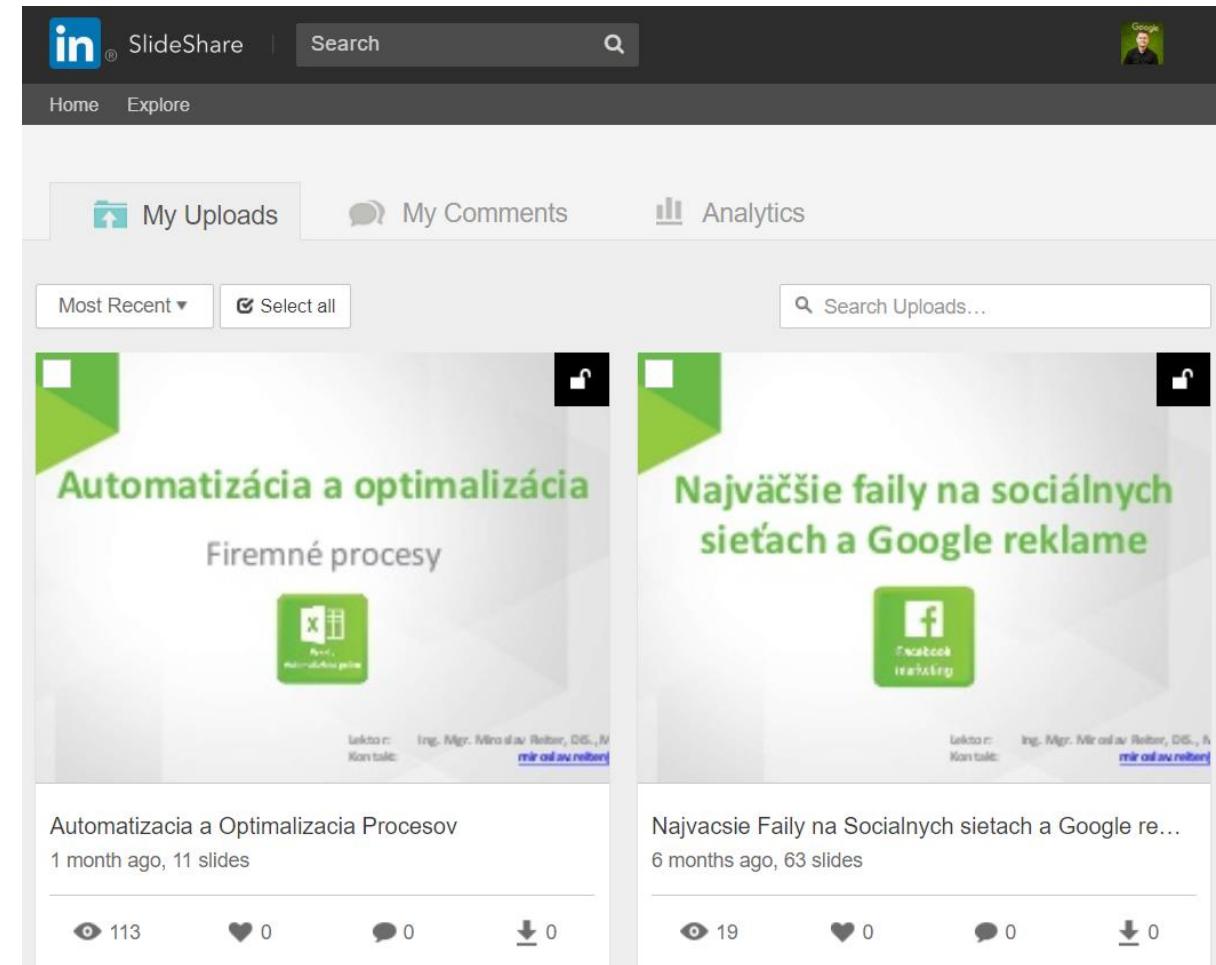
github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R

3. Prezentácia po prednáške

<https://github.com/miroslav-reiter/>

4. Videá na YouTube

www.youtube.com/@VITA-Academy





miroslav-reiter

 Type to search

Overview

Repositories 108

Projects

Packages

Stars 115

Sponsoring



Miroslav Reiter

miroslav-reiter

Founder of VITA Academy ★ Microsoft Certified Trainer ★ Google, Android Certified Trainer ★ ISTQB Trainer

[Edit profile](#)

78 followers · 2 following

VITA Academy

Bratislava

miroslav.reiter@it-academy.sk

www.vita.sk

https://orcid.org/0000-0003-1804-651X

@VITA_Academy_SK

in/miroslav-reiter

VitaAcademySK

@VITA-Academy

miroslav-reiter / README.md



About me

- 🌟 Hi, I am @Miroslav-Reiter
- 🧑 I am a Google Certified Trainer GCT, Microsoft Certified Trainer MCT, Microsoft Most Valuable Professional MVP, ISTQB/GASQB Trainer, PRINCE2/ITIL4/ArchiMate/TOGAF/UML/BPMN/Scrum Trainer
- 💻 My preferred programming languages are Java 🍒, Python 🐍, JavaScript (Google Apps Script), VBA and R
- 🎓 I teach programming and how to use IT technologies effectively at [VITA](#) and [IT Academy](#) My online courses can be bought directly at [VITA.sk](#). I recommend buying an annual subscription with all courses
[Online Akreditované Kurzy a Skolenia VITA](#)
- 🗣 I frequently speak at conferences and workshops. Mostly about IT (programming, automation, certification, testing, enterprise architecture and modeling), **management** (Project management, PRINCE2, ITIL4), **education** and **online marketing** with ads (Google Ads, Google Analytics, Copywriting, Social media, YouTube)
- 🎥 I regularly upload a new videos to [YouTube channel Miroslav Reiter - VITA Academy](#)
 YouTube IT Academy 9.4k
- 🔮 Fun fact: I have graduated from 14 different Universities. I am currently studying at 4 other Universities, 3 of which are PhDs. Check out my [LinkedIn](#) profile

My Socials



YOUTUBE



VITA



FACEBOOK



LINKEDIN



VITA ACADEMY



LINKEDIN



TWITTER



INSTAGRAM



Google VITA



Google IT Academy

Languages, Tools and Frameworks





Code

Issues

Pull requests

Discussions

Actions

Projects

Wiki

Security

Insights

Settings



Kurzy_SAV_DataScience_Python_BI_AI_ML

Public

Unpin

Unwatch 16

Fork 23

Starred 22

main

1 Branch

0 Tags

Go to file

Add file

Code



miroslav-reiter Update README.md

0ec8ff4 · now

217 Commits



Datasety

Add files via upload

10 months ago



Prezentacie

Add files via upload

2 weeks ago



Zdrojove-Kody

Delete Zdrojove-Kody/Python-3-OOP2/a

2 months ago



Cvicensia-Jemny-Uvod-do-AI-ChatGPT-2024.pdf

Add files via upload

6 months ago



Klavesnica-Python.pdf

Add files via upload

3 years ago



README.md

Update README.md

now



README



Kurzy SAV DataScience, Python, R, Julia, BI, AI/ML, ChatGPT

Materiály, Zdrojové Kódy a Projekty, Prezentácie ku kurzom SAV DataScience, Python, OOP, R, BI, Analytika

Python je interpretovaný, interaktívny, open-source programovací jazyk. Python beží na mnohých variantoch Unixu, na Macu a Windows (súčasťou kurzu bude inštalácia na vašom systéme). Pre absolvovanie kurzu je potrebné mať k dispozícii vlastný notebook (s ľubovoľným operačným systémom podporujúcim Python).

R je programovací jazyk a softvérové prostredie pre štatistickú analýzu a vizualizáciu. R je voľne dostupný pod GNU licenciou a existujú verzie pre mnohé operačné systémy ako Linux, Windows a Mac.

ChatGPT je pokročilý jazykový model umelej inteligencie vyvinutý spoločnosťou OpenAI. Je to variant modelu GPT (Generative Pre-trained Transformer), ktorá je špeciálne navrhnutá na generovanie textu a interakciu v dialógovej

About

Materiály, Zdrojové Kódy, Prezentácie ku kurzom SAV Python, OOP, R, BI, Data Science

python, jupyter, numpy, oop, jupyter-notebook, pandas, python3, matplotlib, sav, beautifulsoup, reiter, matplotlib-pyplot

Readme

Activity

22 stars

16 watching

23 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

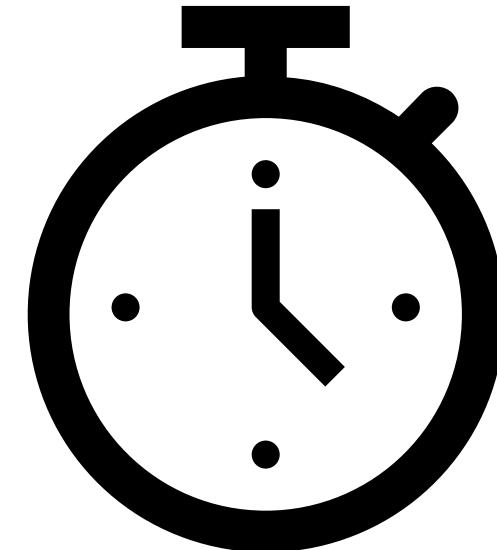
Languages



Suggested workflows

Úvodné Informácie

- Časový rozvrh (9:00-13:30)
 - Programujeme (50 min)
 - Prestávky (10 min)
 - Obedová prestávka
 - Mobilné telefóny a zariadenia
-
- Priprav si otázky a rovno sa pýtaj
 - Interaktívna forma



O mne - Miroslav Reiter

6

40000+
klientov a
1000+ firiem

IT Architekt
Programátor
Manažér

Microsoft
Google
ISTQB tréner

134
certifikácií

151 príručiek
a publikácií

14 škôl

62 projektov

Vlastná firma



MOTIVÁCIA

Študuje 5 odborov a absolvoval už 12 univerzít. Ako zvláda stres a manažovanie času?



Foto: Jakub Kovalík pre FMK UCM | Miroslav Reiter na prednáške Grow with Google na FMK UCM.

Nikola Kotláriková

19. júl 2022 · 8 min. čítania



Miroslav Reiter



1. PhDr. VŠM (Podnikový 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 inf. systémov)
12. Ing. Paed. IGIP STU
13. Mgr. PEVŠ (Bezpečnosť informačných systémov)
14. RNDR. PEVŠ (Bezpečnosť informačných systémov)



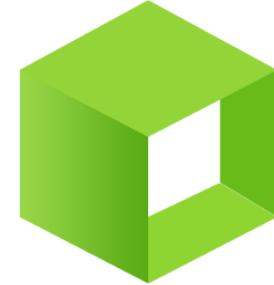
FAKULTA MANAGEMENTU
Univerzita Komenského
v Bratislave

**DIGITÁLNA
UNIVERZITA**

STU
FIIT



 **VITA**
ACADEMY



IT ACADEMY

Most Valuable Professionals

[About](#) [Events](#) [Find an MVP](#)[Profile](#) [Events](#)

Headline

🌟 IT Architect and Programmer 📚 Hard worker 🎓
Lecturer and Certified Trainer

Biography

👉 I'm a hard worker, intellectual and joker. I love learning and teaching as well. My main objective is to teach people and improve their IT knowledge. To create truly practical knowledge necessary for life and present it in an interesting way. I don't like snake charmers and people who cannot...

[▼ Read more](#)

Miroslav Reiter

 Slovakia IT Academy s.r.o.

Most Valuable Professionals

High Impact Activities

Award Category

M365

Technology Area

Visio, Excel

Languages

English, Slovak

Social



This community leader has not added a high impact activity yet.

Znalec

PhDr. Ing. Miroslav Reiter

Evidenčné číslo: 915864

Miesto výkonu činnosti

Tomášikova 50
83104 Bratislava
Slovenská republika
[Zobrazit na mape](#)

Kontaktné údaje

Mobil: +421 908 163 084
E-mail: znapec@it-academy.sk

Odbory a odvetvia

Odbor / Odvetvie	História	Stav
100000 - Elektrotechnika		
100400 - Riadiaca technika, výpočtová technika (hardware)	14.02.2023 - Zápis	AKTÍVNY
100800 - Nosiče zvukových a zvukovoobrazových záznamov	14.02.2023 - Zápis	AKTÍVNY
100900 - Počítačové programy (software)	14.02.2023 - Zápis	AKTÍVNY
101000 - Bezpečnosť a ochrana informačných systémov	14.02.2023 - Zápis	AKTÍVNY

Vyberte si online kurz

Naučte sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zamerajte na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia domova.

Ročné Predplatné na všetky Online Kurzy

~~2290 €~~**490 €**

Priístup pre Vás do všetkých Aktuálnych aj
Pripravovaných Online Kurzov

12 mesačná platnosť

Kúpiť teraz

[Zistiť viac](#)

560 kurzov v ponuke



Zábavné online lekcie



Akreditované kurzy



13 rokov skúseností



Certifikovaní profesionálni lektori

Odporučame Kurzy špeciálne pre vás



Online kurz SAP I.
Začiatočník
298,00€ 398,00€



Balík SAP Profesionál
998,80€ 998,80€



Online kurz ChatGPT a DALL-E AI
186,00€ 254,00€



Online kurz SQL I.
Začiatočník
196,00€ 276,00€



Program MPA Manažér Štátnej a Verejnej Správy
1 940,00€ 2 540,00€



Online kurz Lektor (Akreditovaný Kurz Lektor)
238,00€ 296,00€

Chat

[Všetko](#) [Videá](#) [Obrázky](#) [Správy](#) [Krátke videá](#) [Knihy](#) [Web](#) [: Viac](#)[Nástroje](#)

www.youtube.com › watch

[Ako začať programovať v Pythone? - Online kurz Python a ...](#)

Akreditovaný online kurz Python I. Začiatočník: →
<https://www.vita.sk/online-kurz-python-i-zaciatocnik/> Akreditovaný prezenč...

YouTube · Miroslav Reiter - VITA Academy · 10. 11. 2021

16 kľúčových momentov v tomto videu ▾

www.youtube.com › watch

[Online Kurz Python I. Začiatočník - Ukážka - IDE, Pycharm ...](#)

Online Kurz Python I. Začiatočník - Ukážka - IDE, Pycharm, Print,
Pre..., Konverzie, Netbeans · Comments6.

YouTube · Miroslav Reiter - VITA Academy · 2. 3. 2020

10 kľúčových momentov v tomto videu ▾

www.youtube.com › watch

[Online kurz Python Základy - Práca s Vstupom \(input\) v ...](#)

Prednášajúci: PhDr. Ing. Mgr. et Mgr. et Mgr. et Mgr. Miroslav Reiter, DiS.,
MBA, MPA, MSC, DBA, Ing. Paed. IGIP Miesto: SAV v Bratislave ...

YouTube · Miroslav Reiter - VITA Academy · 25. 8. 2023

4 kľúčové momenty v tomto videu ▾

www.youtube.com › watch

[Online kurz Python - Používanie AI pri Programovaní v ...](#)

... Online kurz Python - Používanie AI pri Programovaní v Pythone a
Generovanie Zdrojových Kódov Akreditovaný online kurz Programovanie...

YouTube · Miroslav Reiter - VITA Academy · 25. 9. 2023

4 kľúčové momenty v tomto videu ▾

www.youtube.com › watch

[Online kurz Python - Filozofia, Princípy a Dokumentácia ...](#)

Comments4 ; Online kurz Python - Cyklus While, break a continue. Miroslav
Reiter - VITA Academy · 156 views ; Online kurz Jazyk Julia - Ako ...

[Všetko](#) [Obrázky](#) [Videá](#) [Správy](#) [Knihy](#) [Finance](#) [Krátke videá](#) [: Viac](#)[Nástroje](#)[www.youtube.com > watch](#)

Ako začať programovať v Pythone? - Online kurz Python a ...



... Python I. Začiatočník: → <https://www.it-academy.sk/kurz/python/> Online kurz Python ... Online #Kurzy #Školenia #Programovanie #Python...

YouTube · Miroslav Reiter - VITA Academy · 10. 11. 2021

[16 kľúčových momentov v tomto videu](#) ▾[www.youtube.com > watch](#)

Online kurz Python Základy - Práca s Vstupom (input) v ...



... Online #Kurzy #Školenia #Programovanie #Jetbrains #Datalore #Jupyter.
... <https://www.vita.sk/on...> Akreditovaný online kurz Programovanie ...

YouTube · Miroslav Reiter - VITA Academy · 25. 8. 2023

[4 kľúčové momenty v tomto videu](#) ▾[www.youtube.com > watch](#)

Online Kurz Python I. Začiatočník - Ukážka - IDE, Pycharm ...



... kurz/python/ Online kurz Python I. Začiatočník je pre teba vhodný, ak si ...
... Online #Kurzy #Školenia #Programovanie #Jazyk #Python #Idle.

YouTube · Miroslav Reiter - VITA Academy · 2. 3. 2020

[10 kľúčových momentov v tomto videu](#) ▾[www.youtube.com > watch](#)

Online kurz Python - Používanie AI pri Programovaní v ...



... Online kurz Python - Používanie AI pri Programovaní v Pythone a Generovanie Zdrojových Kódov Akreditovaný online kurz Programovanie...

YouTube · Miroslav Reiter - VITA Academy · 25. 9. 2023

[4 kľúčové momenty v tomto videu](#) ▾[www.youtube.com > watch](#)

Online kurz Python - Čo je to Python? - Na čo a kde sa Python ...



Comments3 · Online Kurz Python Objektové Programovanie - Čo je to Polymorfizmus v Pythone? · Online Kurz Úvod do Programovania - Ukážk...



Luigi, Mário
a Yoshi

Čo Robíte?

1. Študent/Učiteľ

2. Zamestnanec

3. Podnikateľ

4. Nezamestnaný/materská

5. Dievča pre všetko



Vaše Ciele a Očakávania

17

1. Doplniť si znalosti z jazyka Python

2. Práca s dátami a súbormi

3. Práca s časom a generovanie dát

4. Efektívna práca s dátovými štruktúrami

5. Logovanie a diagnostika

Zábava je v zaručená v každom bode 😊



Ako používať Štandardnú Knižnicu

AKREDITOVANÝ KURZ





Čo sa naučíme?

1. Čo je to štandardná knižnica?
2. Aké obsahuje a má moduly?
3. Aké sú zabudované funkcie?
4. Ako sa zorientovať v moduloch?
5. Ako používať moduly?
6. Aké sú kľúčové moduly?
7. Kde nájdeme dokumentáciu?
8. Profit

Čo je to Štandardná Knižnica?



- **Súbor modulov**
- Je **súčasťou samotného jazyka Python**
- Poskytuje **množstvo nástrojov na riešenie bežných úloh**
- Práca so **súbormi, prístup k systémovým funkciám, spracovanie dát, logovanie...**
- Obsahuje **základné dátové typy** ako **čísla** a **zoznamy**
- Čalej **funkcie** a **výnimky dostupné bez importu**

Python 3.13.2 documentation

Welcome! This is the official documentation for Python 3.13.2.

Documentation sections:

[What's new in Python 3.13?](#)

Or all "What's new" documents since Python 2.0

[Tutorial](#)

Start here: a tour of Python's syntax and features

[Library reference](#)

Standard library and builtins

[Language reference](#)

Syntax and language elements

[Python setup and usage](#)

How to install, configure, and use Python

[Python HOWTOs](#)

In-depth topic manuals

[Installing Python modules](#)

Third-party modules and PyPI.org

[Distributing Python modules](#)

Publishing modules for use by other people

[Extending and embedding](#)

For C/C++ programmers

[Python's C API](#)

C API reference

[FAQs](#)

Frequently asked questions (with answers!)

[Deprecations](#)

Deprecated functionality

Indices, glossary, and search:

[Global module index](#)

All modules and libraries

[Search page](#)

Search this documentation

[General index](#)

All functions, classes, and terms

[Complete table of contents](#)

Lists all sections and subsections



[Previous topic](#)

10. Full Grammar specification

[Next topic](#)

Introduction

[This Page](#)[Report a Bug](#)[Show Source](#)

The Python Standard Library

While [The Python Language Reference](#) describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

In addition to the standard library, there is an active collection of hundreds of thousands of components (from individual programs and modules to packages and entire application development frameworks), available from the [Python Package Index](#).

- «
- [Introduction](#)
 - [Notes on availability](#)
 - [Built-in Functions](#)
 - [Built-in Constants](#)
 - [Constants added by the site module](#)
 - [Built-in Types](#)
 - [Truth Value Testing](#)
 - [Boolean Operations — and, or, not](#)
 - [Comparisons](#)
 - [Numeric Types — int, float, complex](#)
 - [Boolean Type - bool](#)
 - [Iterator Types](#)

Hlavné Moduly Python Lib

1. Moduly pre spracovanie dát a výpočty

- math, random, csv, json, xml, sqlite3, uuid

2. Manipulácia so súbormi a adresármi

- os, pathlib, shutil

3. Prácu s časom a dátumom

- datetime, time

4. Spracovanie a analýza dátových štruktúr

- collections, itertools

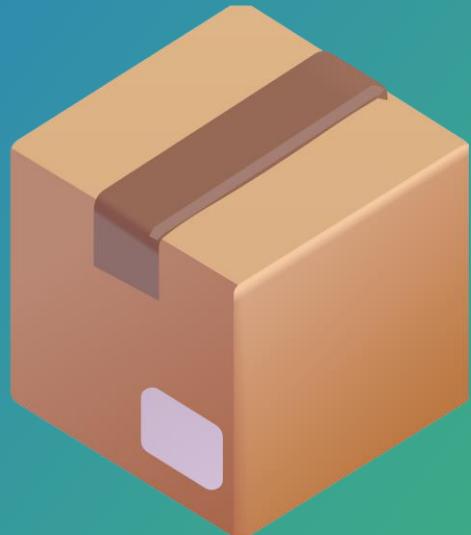
5. Siete, procesy, emails, požiadavky

- socket, email, subprocess, http

6. Logovanie, diagnostika a ladenie

- logging, sys, pdb

Python Modul



1. Súbor kódu v jazyku Python
2. Obsahuje definície funkcií, tried, premenných a často aj spustiteľný kód
3. Slúži na organizovanie a zoskupovanie súvisiacich funkcií a tried do 1 súboru
4. Umožňuje zjednodušiť správu a opäťovné použitie kódu (DRY)
5. Celkovo 200 modulov (24 deprecated), 176 aktuál (Python 3.13.2)

Typy Modulov

1. Štandardné moduly

- Súčasť Pythonu a poskytujú množstvo preddefinovaných nástrojov

2. Externé moduly

- Tieto moduly sú dodávané tretími stranami a môžu byť nainštalované pomocou nástroja ako pip, easy install, poetry

3. Vlastné moduly

- Tieto moduly si vytvoríte sami, aby ste mohli organizovať svoj kód a využívať ho v rôznych projektoch.

Python Module Index

_ | [a](#) | [b](#) | [c](#) | [d](#) | [e](#) | [f](#) | [g](#) | [h](#) | [i](#) | [j](#) | [k](#) | [l](#) | [m](#) | [n](#) | [o](#) | [p](#) | [q](#) | [r](#) | [s](#) | [t](#) | [u](#) | [v](#) | [w](#) | [x](#) | [z](#)

-

[__future__](#) Future statement definitions

[__main__](#) The environment where top-level code is run. Covers command-line interfaces, import-time behavior, and ```__name__ == '__main__'`''.

[_thread](#) Low-level threading API.

[_tkinter](#) A binary module that contains the low-level interface to Tk/Tcl.

a

[abc](#) Abstract base classes according to :pep:`3119`.

[aifc](#) **Deprecated:** Removed in 3.13.

[argparse](#) Command-line option and argument parsing library.

[array](#) Space efficient arrays of uniformly typed numeric values.

[ast](#) Abstract Syntax Tree classes and manipulation.

[asynchat](#) **Deprecated:** Removed in 3.12.

[asyncio](#) Asynchronous I/O.

[asyncore](#) **Deprecated:** Removed in 3.12.

[atexit](#) Register and execute cleanup functions.

Table of Contents

- statistics** — Mathematical statistics functions
 - Averages and measures of central location
 - Measures of spread
 - Statistics for relations between two inputs
 - Function details
 - Exceptions
 - **NormalDist** objects
 - Examples and Recipes
 - Classic probability problems
 - Monte Carlo inputs for simulations
 - Approximating binomial distributions
 - Naive bayesian classifier

Previous topic

[random](#) — Generate pseudo-random numbers

Next topic

[Functional Programming Modules](#)

This Page

[Report a Bug](#)

[Show Source](#)

statistics — Mathematical statistics functions

Added in version 3.4.

Source code: [Lib/statistics.py](#)

This module provides functions for calculating mathematical statistics of numeric ([Real](#)-valued) data.

The module is not intended to be a competitor to third-party libraries such as [NumPy](#), [SciPy](#), or proprietary full-featured statistics packages aimed at professional statisticians such as Minitab, SAS and Matlab. It is aimed at the level of graphing and scientific calculators.

Unless explicitly noted, these functions support [int](#), [float](#), [Decimal](#) and [Fraction](#). Behaviour with other types (whether in the numeric tower or not) is currently unsupported. Collections with a mix of types are also undefined and implementation-dependent. If your input data consists of mixed types, you may be able to use [map\(\)](#) to ensure a consistent result, for example: `map(float, input_data)`.

Some datasets use NaN (not a number) values to represent missing data. Since NaNs have unusual comparison semantics, they cause surprising or undefined behaviors in the statistics functions that sort data or that count occurrences. The functions affected are `median()`, `median_low()`, `median_high()`, `median_grouped()`, `mode()`, `multimode()`, and `quantiles()`. The NaN values should be stripped before calling these functions:

```
<>> from statistics import median
<>> from math import isnan
<>> from itertools import filterfalse

<>> data = [20.7, float('NaN'), 19.2, 18.3, float('NaN'), 14.4]
<>> sorted(data) # This has surprising behavior
[20.7, nan, 14.4, 18.3, 19.2, nan]
<>> median(data) # This result is unexpected
16.35

<>> sum(map(isnan, data)) # Number of missing values
2
<>> clean = list(filterfalse(isnan, data)) # Strip NaN values
<>> clean
[20.7, 19.2, 18.3, 14.4]
<>> sorted(clean) # Sorting now works as expected
```

Python Ekosystém

Modul

- Súbor Python kódu, ktorý obsahuje definície funkcií, tried, premenných a iného kódu. Môže byť použitý na zdieľanie a opäťovné použitie kódu
- Menší súbor kódu (jediný .py súbor)
- Umožňuje organizovať kód do samostatných jednotiek, ktoré sa dajú importovať
- math.py, random.py
- Nie je potrebné inštalovať, ak ide o vlastný alebo štandardný modul

Balíček

- Je adresár, ktorý obsahuje viacero modulov (súborov .py) a môže obsahovať aj podadresáre. Balíček je štruktúrovaný spôsob organizácie viacerých modulov
- Veľkosť: Adresár so súbormi .py a prípadne súbory ako __init__.py, ktoré umožňujú Pythonu identifikovať adresár ako balíček.
- Použitie: Umožňuje rozdeliť veľké knižnice do viacerých modulov a podmodulov
- numpy/, pandas/ (adresáre obsahujúce moduly ako numpy.linalg.py, numpy.random.py).
- Balíčky sa obvykle inštalujú pomocou pip

Knižnica

- Je kolekcia modulov alebo balíčkov, ktoré poskytujú rozsiahle nástroje a funkcie pre špecifické úlohy. Môže obsahovať mnoho balíčkov, ktoré sú navzájom prepojené a slúžia na rozličné potreby
- Rozsiahla kolekcia, ktorá môže obsahovať mnoho balíčkov a modulov
- Umožňuje vykonávať špecifické úlohy, ako napríklad numerické výpočty, manipuláciu s dátami, alebo vývoj webových aplikácií
- NumPy, Pandas, Flask (sú knižnice, ktoré môžu obsahovať viacero balíčkov a modulov)
- Knižnice sú zvyčajne inštalované pomocou nástroja pip a distribuované cez Python Package Index (PyPI)

Kritérium	Modul 	Balíček 	Knižnica 
Definícia	Súbor kódu (<code>.py</code> súbor)	Adresár obsahujúci viacero modulov a súbor <code>__init__.py</code>	Kolekcia modulov alebo balíčkov, ktoré poskytujú špecifické nástroje a funkciu
Veľkosť	Malý (jeden <code>.py</code> súbor)	Adresár s viacerými <code>.py</code> súbormi	Rozsiahla kolekcia, môže obsahovať balíčky a moduly
Účel	Rozdelenie kódu do funkčných blokov	Organizovanie viacerých súvisiacich modulov	Poskytovanie rozsiahlych nástrojov pre špecifické úlohy
Príklad	<code>math.py</code> , <code>random.py</code>	<code>numpy/</code> , <code>pandas/</code>	<code>NumPy</code> , <code>Pandas</code> , <code>Flask</code>
Inštalácia	Bez potreby inštalácie, ak ide o vlastný alebo štandardný modul	Balíček sa inštaluje ako celok cez <code>pip</code>	Knižnica sa inštaluje cez <code>pip</code> a obsahuje balíčky alebo moduly
Závislosti	Závisí na iných moduloch	Závisí na iných balíčkoch alebo moduloch	Môže mať mnoho závislostí na iných knižniciach, balíčkoch alebo moduloch

Table of Contents

Introduction

- Notes on availability
 - WebAssembly platforms
 - Mobile platforms

Previous topic

The Python Standard Library

Next topic

Built-in Functions

This Page

[Report a Bug](#)

[Show Source](#)

Introduction

The “Python library” contains several different kinds of components.

It contains data types that would normally be considered part of the “core” of a language, such as numbers and lists. For these types, the Python language core defines the form of literals and places some constraints on their semantics, but does not fully define the semantics. (On the other hand, the language core does define syntactic properties like the spelling and priorities of operators.)

The library also contains built-in functions and exceptions — objects that can be used by all Python code without the need of an [`import`](#) statement. Some of these are defined by the core language, but many are not essential for the core semantics and are only described here.

The bulk of the library, however, consists of a collection of modules. There are many ways to dissect this collection. Some modules are written in C and built in to the Python interpreter; others are written in Python and imported in source form. Some modules provide interfaces that are highly specific to Python, like printing a stack trace; some provide interfaces that are specific to particular operating systems, such as access to specific hardware; others provide interfaces that are specific to a particular application domain, like the World Wide Web. Some modules are available in all versions and ports of Python; others are only available when the underlying system supports or requires them; yet others are available only when a particular configuration option was chosen at the time when Python was compiled and installed.

This manual is organized “from the inside out:” it first describes the built-in functions, data types and exceptions, and finally the modules, grouped in chapters of related modules.

This means that if you start reading this manual from the start, and skip to the next chapter when you get bored, you will get a reasonable overview of the available modules and application areas that are supported by the Python library. Of course, you don’t *have* to read it like a novel — you can also browse the table of contents (in front of the manual), or look for a specific function, module or term in the index (in the back). And finally, if you enjoy learning about random subjects, you choose a random page number (see module [`random`](#)) and read a sec-

[Previous topic](#)

Introduction

[Next topic](#)

Built-in Constants

[This Page](#)[Report a Bug](#)[Show Source](#)

Built-in Functions

The Python interpreter has a number of functions and types built into it that are always available. They are listed here in alphabetical order.

Built-in Functions			
A	E	L	R
abs()	enumerate()	len()	range()
aiter()	eval()	list()	repr()
all()	exec()	locals()	reversed()
anext()			round()
any()			
ascii()			
B	F	M	S
bin()	filter()	map()	set()
bool()	float()	max()	setattr()
breakpoint()	format()	memoryview()	slice()
bytearray()	frozenset()	min()	sorted()
bytes()			staticmethod()
C	G	N	O
callable()	getattr()	next()	object()
chr()	globals()		oct()
classmethod()			open()
compile()			ord()
complex()			
D	H	P	T
delattr()	hasattr()	pow()	tuple()
dict()	hash()	print()	type()
dir()	help()	property()	
	hex()		
	I	V	Z
	id()	vars()	
	input()		
	int()		
	isinstance()		

Úlohy

Štandardná

Knižnica

1. Čo je to štandardná knižnica?
2. Aké obsahuje a má moduly?
3. Aké sú hlavné funkcie?
4. Ako sa zorientovať v moduloch?
5. Ako používať moduly?
6. Kde nájdeme dokumentáciu?



Ako pracovať s Dátami a Súbormi?

AKREDITOVANÝ KURZ





Čo sa naučíme?

1. Čo je to modul os?
2. Čo je to modul shutil?
3. Čo je to modul pathlib?
4. Ako sa tieto moduly používajú?
5. Aké sú alternatívy?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?
8. Profit

Čo je Modul os?

- Poskytuje **rozhranie na interakciu s operačným systémom**
- Umožňuje **manipuláciu so súbormi a adresármí, získavanie informácií o systéme**
- **Spúšťanie príkazov v shelli a prácu s environmentálnymi premennými (environs).**

Modul os

1. Získanie aktuálneho pracovného adresára

python

Kopírovať

Upraviť

```
import os
```

```
print(os.getcwd()) # Vráti cestu k aktuálnemu adresáru
```

Používa `os.getcwd()` na získanie aktuálneho adresára.

2. Zmena pracovného adresára

python

Kopírovať

Upraviť

```
import os
```

```
os.chdir('/path/to/directory') # Zmení aktuálny adresár
```

Používa `os.chdir(path)` na zmenu adresára.

3. Vypísanie súborov a priečinkov v adresári

python

Kopírovať

Upraviť

```
import os
```

```
print(os.listdir('.')) # Vypíše obsah aktuálneho adresára
```

Používa `os.listdir(path)` na získanie zoznamu súborov a priečinkov.

Kedy použiť os?

1.  Na manipuláciu so súbormi a adresármi (vytváranie, mazanie, premenovanie)
 2.  Na získanie informácií o operačnom systéme a environmentálnych premenných
 3.  Na vykonávanie shell príkazov z Python skriptu
- ◆ **Alternatívy:**
 - A. **pathlib** – modernejší, OO prístup na prácu s cestami
 - B. **shutil** – pokročilé operácie ako kopírovanie a presúvanie súborov

Čo je Modul shutil?

- Poskytuje **funkcie** na **vysokú úroveň manipulácie** so **súbormi** a **adresármi**
- Kopírovanie, presúvanie, mazanie a archivovanie
- Umožňuje **efektívnu prácu so súborovými systémami** bez nutnosti implementácie vlastných operácií

Modul shutil

1. Kopírovanie súboru

python

 Kopírovať

 Upraviť

```
import shutil
```

```
shutil.copy("subor.txt", "zaloha.txt") # Skopíruje obsah aj s oprávneniami
```

 Používa `shutil.copy(src, dst)` na kopírovanie súboru.

2. Kopírovanie súboru so zachovaním metaúdajov (časové pečiatky)

python

 Kopírovať

 Upraviť

```
import shutil
```

```
shutil.copy2("subor.txt", "zaloha.txt")
```

 Používa `shutil.copy2(src, dst)` na kopírovanie aj s časovými značkami.

3. Kopírovanie celého adresára

python

 Kopírovať

 Upraviť

```
import shutil
```

```
shutil.copytree("data", "backup")
```

 Používa `shutil.copytree(src, dst)` na rekurzívne kopírovanie adresára.

Kedy použiť shutil?

1.  Pri manipulácii so súbormi a adresármi na vyššej úrovni (kopírovanie, presun, archivácia)
 2.  Pri potrebe efektívneho mazania veľkých adresárov
 3.  Pri práci s diskovým priestorom a systémovými operáciami
- ◆ **Alternatívy:**
 - A. **os** – poskytuje základné funkcie pre manipuláciu so súbormi a adresármi
 - B. **zipfile** – umožňuje detailnejšiu prácu s archiváciou ZIP súborov

Čo je Modul pathlib?

- Poskytuje **objektovo orientovaný prístup** na manipuláciu s **cestami k súborom** a **adresárom**
- Umožňuje **jednoduchú** a **prehľadnú** prácu s **cestami** naprieč **rôznymi operačnými systémami**
- Je **odporúčanou náhradou** za modul **os** pri práci so **súbormi** a **adresármí**

Modul pathlib

1. Získanie aktuálneho pracovného adresára

python

[Kopírovať](#) [Upraviť](#)

```
from pathlib import Path
```

```
print(Path.cwd()) # Vráti cestu k aktuálnemu adresáru
```

Používa `Path.cwd()` na zistenie aktuálneho adresára.

2. Získanie domovského adresára používateľa

python

[Kopírovať](#) [Upraviť](#)

```
from pathlib import Path
```

```
print(Path.home()) # Vráti domovský adresár používateľa
```

Používa `Path.home()` na získanie cesty k domovskému adresáru.

3. Kontrola existencie súboru alebo adresára

python

[Kopírovať](#) [Upraviť](#)

```
from pathlib import Path
```

```
cesta = Path("subor.txt")
```

```
print(cesta.exists()) # True, ak súbor existuje
```

Používa `Path.exists()` na kontrolu existencie.

Kedy použiť pathlib?

1.  Pri manipulácii s cestami k súborom a adresárom v objektovo orientovanom štýle
 2.  Ked' chcete zjednodušiť prácu s cestami naprieč rôznymi operačnými systémami
 3.  Pri potrebe prehľadnej syntaxe a modernej alternatívy k modulu os
- ◆ **Alternatívy:**
 - A. **os** – tradičný modul/metóda manipulácie so súbormi a adresármi
 - B. **shutil** – pokročilé operácie ako kopírovanie, presúvanie a archivácia

Čo je Modul CSV?

- Poskytuje **nástroje** na **čítanie, zápis a manipuláciu** s **CSV súbormi** (**Comma-Separated Values**)
- CSV súbory sú **textové súbory**, kde sú **hodnoty oddelené čiarkami, bodkočiarkami alebo inými znakmi**
- Často sa používajú na **ukladanie tabuľiek** a **export údajov** z **databáz** alebo **Excelu**

CSV

1. Čítanie CSV súborov

python

Kopírovať Upraviť

```
import csv

with open('data.csv', newline='') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        print(row)
```

Používa `csv.reader()` na čítanie CSV súboru riadok po riadku.

2. Zápis do CSV súboru

python

Kopírovať Upraviť

```
import csv

data = [["Meno", "Vek", "Mesto"], ["Anna", 25, "Bratislava"], ["Peter", 30, "Košice"]]

with open('output.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile)
    writer.writerows(data)
```

Používa `csv.writer()` na zápis údajov do CSV súboru.

Kedy použiť CSV?

1.  Na prácu s veľkými tabuľkovými dátami v textovom formáte
2.  Pri exporte a importe údajov medzi aplikáciami
3.  Na spracovanie údajov získaných z Excelu alebo databáz

- ◆ **Alternatívy:**

Pri práci s veľkými CSV súbormi môžete použiť aj Pandas (`pd.read_csv()`) a Polars na efektívnejšie spracovanie dát.

Čo je Modul JSON?

- Poskytuje **nástroje na** prácu s **JSON** (JavaScript Object Notation)
- Populárny formát na ukladanie a výmenu dát medzi systémami
- Umožňuje **konverziu** medzi **JSON** a dátovými štruktúrami Pythonu, ako sú **slovníky** a **zoznamy**

Modul JSON

1. Konverzia Python objektu na JSON reťazec

python

`import json`

```
data = {"meno": "Anna", "vek": 25, "mesto": "Bratislava"}  
json_string = json.dumps(data)  
print(json_string) # Výstup: {"meno": "Anna", "vek": 25, "mesto": "Bratislava"}
```

[Kopírovať](#)

[Upraviť](#)

Používa `json.dumps(obj)` na prevod Python objektu do JSON reťazca.

2. Zápis JSON do súboru

python

`import json`

```
data = {"meno": "Anna", "vek": 25, "mesto": "Bratislava"}  
  
with open("data.json", "w") as file:  
    json.dump(data, file)
```

[Kopírovať](#)

[Upraviť](#)

Používa `json.dump(obj, file)` na zápis JSON dát do súboru.

3. Načítanie JSON zo súboru

python

`import json`

```
with open("data.json", "r") as file:  
    data = json.load(file)  
print(data) # Výstup: {'meno': 'Anna', 'vek': 25, 'mesto': 'Bratislava'}
```

[Kopírovať](#)

[Upraviť](#)

Používa `json.load(file)` na načítanie JSON dát zo súboru.

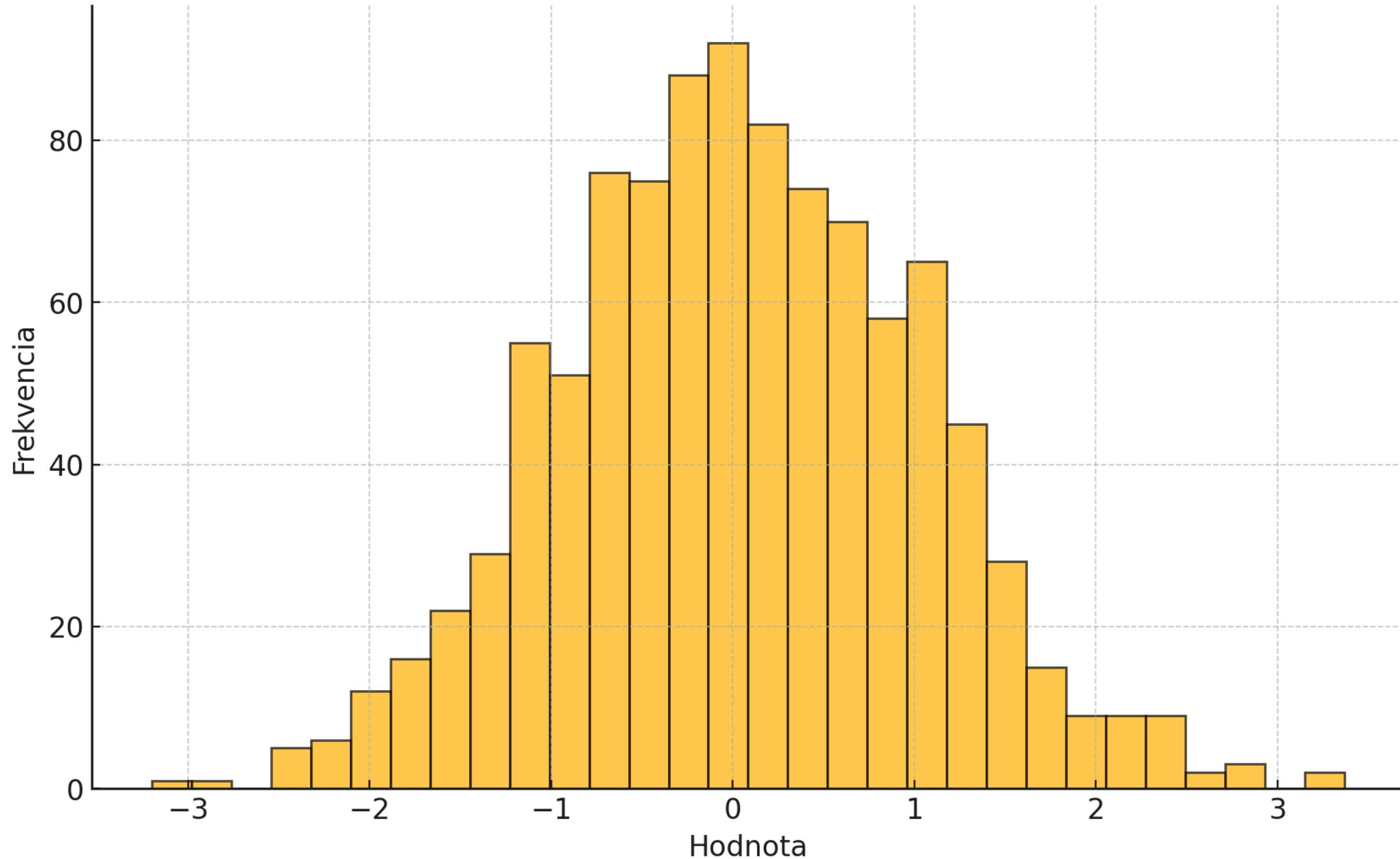
Kedy použiť JSON?

1.  Pri ukladaní a načítaní údajov v JSON formáte
2.  Pri práci s webovými API, ktoré vracajú dátá v JSON
3.  Pri serializácii a deserializácii dátových štruktúr

- ◆ **Alternatívy:**

- A. **ujson** – rýchlejšia alternatíva pre JSON spracovanie
- B. **orjson** – vysoko výkonná knižnica pre JSON v Pythone

Normálne rozdelenie (Gaussova distribúcia)



Úlohy Súbory

1. Čo je to modul os?
2. Čo je to modul shutil?
3. Čo je to modul pathlib?
4. Ako sa tieto moduly používajú?
5. Aké sú alternatívy?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?



Ako pracovať s Časom a Dátumami

AKREDITOVANÝ KURZ





Čo sa naučíme?

1. Ako sa pracuje s dátumom?
2. Aké sú alternatívy?
3. Ako sa pracuje s náhodou?
4. Aké sú alternatívy?
5. Ako sa tieto moduly používajú?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?
8. Profit

Čo je Modul datetime?

- Poskytuje **nástroje na prácu s dátumami a časom**
- Vrátane ich **získania, formátovania, výpočtov a porovnávania**
- Umožňuje **efektívnu manipuláciu s časovými údajmi** a je **široko používaný v rôznych aplikáciách**

Modul datetime

1. Získanie aktuálneho dátumu a času

python

[Kopírovať](#) [Upraviť](#)

```
from datetime import datetime

teraz = datetime.now()
print(teraz) # Výstup: 2025-03-19 14:30:45.123456
```

Používa `datetime.now()` na získanie aktuálneho dátumu a času.

2. Získanie aktuálneho dátumu bez času

python

[Kopírovať](#) [Upraviť](#)

```
from datetime import date

dnes = date.today()
print(dnes) # Výstup: 2025-03-19
```

Používa `date.today()` na získanie aktuálneho dátumu.

3. Vytvorenie konkrétneho dátumu a času

python

[Kopírovať](#) [Upraviť](#)

```
from datetime import datetime

vlastny_datum = datetime(2025, 5, 10, 15, 30)
print(vlastny_datum) # Výstup: 2025-05-10 15:30:00
```

Používa `datetime(rok, mesiac, deň, hodina, minúta, sekunda)` na vytvorenie vlastného dátumu a času.

Kedy použiť' datetime?

1.  Pri práci s dátumami a časom v aplikáciách, ktoré vyžadujú časové údaje
 2.  Pri potrebe formátovania dátumu a času pre výstupy alebo databázy
 3.  Pri výpočtoch rozdielov medzi dátumami (napr. odpočítavanie dní do udalosti)
- ◆ Alternatívy:
 - A. **time** – poskytuje základné časové funkcie, ale menej možností pre prácu s dátumami
 - B. **pytz** – rozširuje modul datetime o podporu časových zón

Čo je Modul random?

- Poskytuje **nástroje na generovanie náhodných čísel, výber náhodných prvkov zo zoznamov a náhodné usporiadanie sekvencií**
- Je užitočný pri **simuláciách, štatistických analýzach, hrách a generovaní testovacích údajov**

Modul random

1. Generovanie náhodného celého čísla v rozsahu

python

 Kopírovať

 Upravit

```
import random

cislo = random.randint(1, 100)
print(cislo) # Náhodné číslo od 1 do 100
```

 Používa `random.randint(a, b)` na vygenerovanie náhodného celého čísla.

2. Generovanie náhodného desatinného čísla (float)

python

 Kopírovať

 Upravit

```
import random

cislo = random.uniform(1, 100)
print(cislo) # Náhodné desatinné číslo od 1 do 100
```

 Používa `random.uniform(a, b)` na vygenerovanie náhodného `float` čísla.

3. Generovanie náhodného čísla medzi 0 a 1

python

 Kopírovať

 Upravit

```
import random

cislo = random.random()
print(cislo) # Náhodné číslo od 0.0 do 1.0
```

 Používa `random.random()` na získanie čísla v rozsahu `<0,1>`.

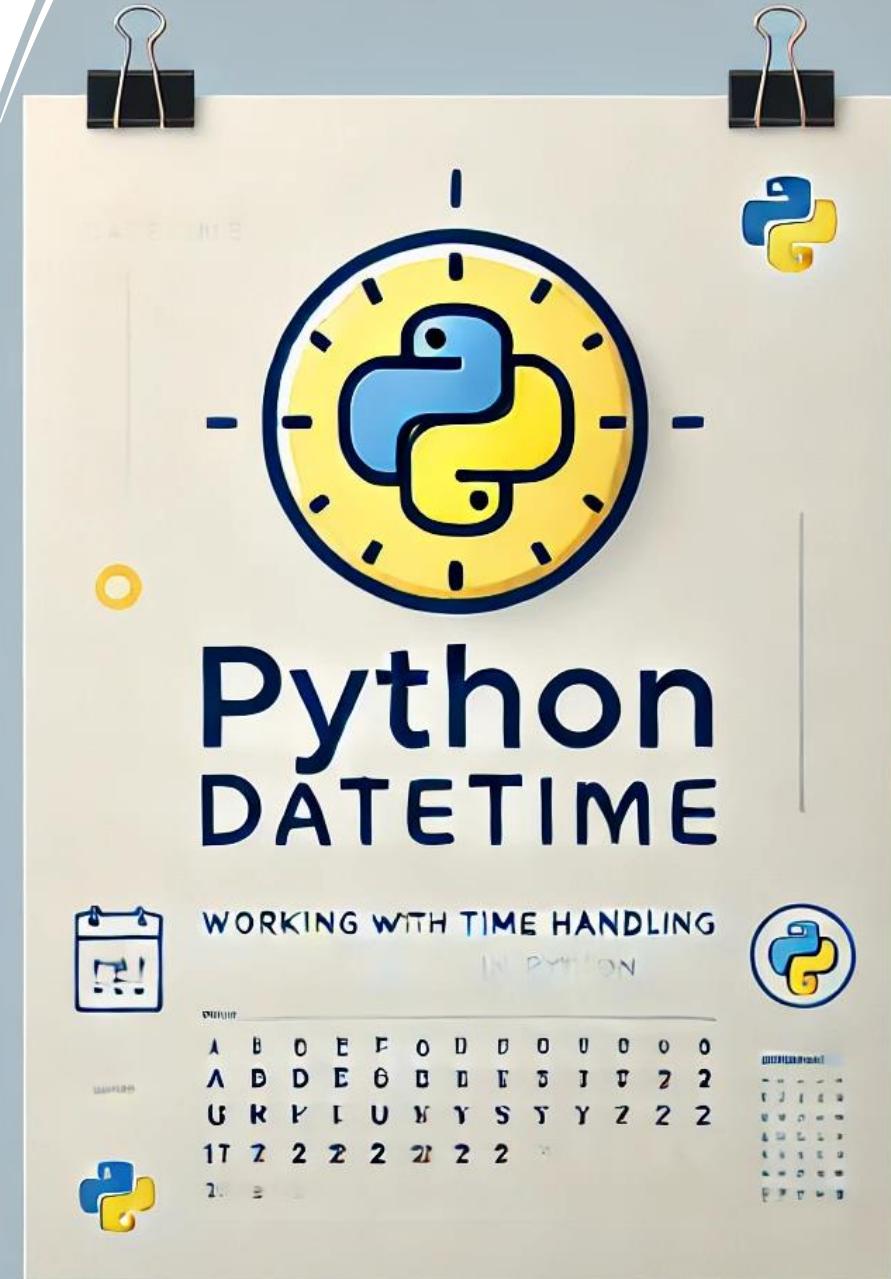
Kedy použiť random?

1.  Pri generovaní náhodných čísel pre simulácie, testovanie a hry
2.  Pri výbere náhodných prvkov zo zoznamu
3.  Pri náhodnom usporiadaní sekvencií a štatistických výpočtoch
 - ◆ Alternatívy:
 - A. **numpy.random** – ponúka pokročilé generovanie náhodných hodnôt a rozdelení
 - B. **secrets** – používa sa na bezpečné generovanie náhodných hodnôt pre kryptografické aplikácie

Úlohy

Dátumy

1. Ako sa pracuje s dátumom?
2. Aké sú alternatívy?
3. Ako sa pracuje s náhodou?
4. Aké sú alternatívy?
5. Ako sa tieto moduly používajú?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?



Ako používať Collections Python

AKREDITOVANÝ KURZ





Čo sa naučíme?

1. Čo je to **collections**?
2. Aké ma **zabudované štruktúry**?
3. Ako sa pracuje s **collections**?
4. Aké sú **alternatívy**?
5. Aké sú **vhodné prípady použitia**?
6. Kde nájdeme **dokumentáciu**?
7. Profit

Čo je Modul Collections?

- Poskytuje **rozšírené dátové štruktúry založené na vstavaných typoch**, ako sú **zoznamy, slovníky a množiny**
- Obsahuje **optimalizované verzie bežných kontajnerov**, ktoré umožňujú **efektívnejšiu manipuláciu s dátami**

Modul Collections

1. Counter – Počítanie výskytov prvkov v zozname

python

[Kopírovať](#) [Upravit](#)

```
from collections import Counter

zoznam = ["a", "b", "a", "c", "b", "a"]
pocitadlo = Counter(zoznam)
print(pocitadlo) # Výstup: Counter({'a': 3, 'b': 2, 'c': 1})
```

Používa `Counter(iterable)` na spočítanie výskytu prvkov.

2. Counter.most_common(n) – Získanie n najčastejších prvkov

python

[Kopírovať](#) [Upravit](#)

```
from collections import Counter

text = "mississippi"
pocitadlo = Counter(text)
print(pocitadlo.most_common(2)) # Výstup: [('i', 4), ('s', 4)]
```

Používa `most_common(n)` na získanie n najčastejších prvkov.

3. defaultdict – Slovník s predvolenou hodnotou

python

[Kopírovať](#) [Upravit](#)

```
from collections import defaultdict

slovnik = defaultdict(int) # Predvolená hodnota je 0
slovnik["a"] += 1
print(slovnik["b"]) # Výstup: 0 (namiesto KeyError)
```

Používa `defaultdict(typ)`, kde typ určuje predvolenú hodnotu.

Kedy použiť Collections?

1.  Pri práci s veľkými zoznamami a slovníkmi na zvýšenie efektivity
 2.  Pri potrebe zachovať poradie alebo používať predvolené hodnoty v slovníkoch
 3.  Pri optimalizácii operácií na zoznamoch pomocou dequeu
- ◆ Alternatívy:
 - A. **dict** – pre základné operácie so slovníkmi
 - B. **list** – pre jednoduché fronty namiesto deque
 - C. **dataclasses** – pokročilejšia alternatíva k namedtuple

Čo je Modul itertools?

- Poskytuje **vysoko optimalizované nástroje** na **prácu s iterátormi**
- Obsahuje efektívne **funkcie** na **generovanie permutácií, kombinácií, nekonečných sekvencií a agregáciu údajov**
- Užitočný pri **práci s veľkými dátovými súbormi, generovaní sekvencií a optimalizácii výpočtov**

Modul itertools

1. Iterácia s opakováním

`itertools.count(start, step)` – Nekonečné generovanie čísel

python

Kopírovať

Upraviť

```
import itertools
```

```
for n in itertools.count(10, 2):
    if n > 20:
        break
    print(n) # Výstup: 10, 12, 14, 16, 18, 20
```

Generuje čísla od `start` s krokom `step` (nekonečne, ak nie je zastavené).

`itertools.cycle(iterable)` – Nekonečné opakovanie sekvenie

python

Kopírovať

Upraviť

```
import itertools
```

```
cyklus = itertools.cycle(["A", "B", "C"])
for _ in range(6):
    print(next(cyklus)) # Výstup: A, B, C, A, B, C
```

Opakuje iterovateľný objekt do nekonečna.

`itertools.repeat(value, n)` – Opakovanie hodnoty `n`-krát

python

Kopírovať

Upraviť

```
import itertools
```

```
for x in itertools.repeat("Python", 3):
    print(x) # Výstup: Python, Python, Python
```

Opakuje hodnotu `n`-krát (alebo nekonečne, ak `n` nie je zadané).

Kedy použiť itertools?

1.  Pri generovaní kombinácií a permutácií
 2.  Pri efektívnom filtrovaní a spracovaní veľkých dátových množín
 3.  Pri optimalizácii práce s iterátormi na pamäťovo efektívne spracovanie dát
- ◆ **Alternatívy:**
 - A. **numpy** – na pokročilé numerické výpočty
 - B. **pandas** – na prácu s veľkými dátovými rámcmi a agregáciami

Úlohy Collections

1. Čo je to collections?
2. Aké ma zabudované štruktúry?
3. Ako sa pracuje s collections?
4. Aké sú alternatívy?
5. Aké sú vhodné prípady použitia?
6. Kde nájdeme dokumentáciu?



Ako pracovať s Logovaním Diagnostikou

AKREDITOVANÝ KURZ





Čo sa naučíme?

1. Čo je to modul Logging?
2. Čo sú to logy?
3. Aké existujú logovacie úrovne?
4. Čo je to modul sys?
5. Ako používať tieto moduly?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?
8. Profit

Čo je Modul logging?

- Poskytuje **flexibilný** a **robustný systém** na **zaznamenávanie správ (logov)** počas behu programu
- Umožňuje **sledovať chyby, upozornenia** a **dôležité udalosti** v **aplikáciách**
- Užitočné **pri ladení** a **analýze výkonu**

Modul logging

1. Základné logovanie

Vytvorenie jednoduchého logu

python

```
import logging
```

```
logging.basicConfig(level=logging.INFO)  
logging.info("Toto je informačná správa")
```

- Nastaví základnú konfiguráciu a zaznamená správu na úrovni `INFO`.

Záznam chybovej správy

python

```
import logging
```

```
logging.error("Toto je chybová správa")
```

- `logging.error()` zaznamená chybu.

Kedy použiť logging?

1.  Pri monitorovaní výkonu a behu aplikácie
 2.  Pri sledovaní chýb a laditeľných správ
 3.  Pri ukladaní histórie udalostí do súborov
- ◆ Alternatívy:
 - A. **loguru/logbook** – modernejší a flexibilnejší logovací nástroj
 - B. **print()** – nevhodné na rozsiahle logovanie

Logovacie Úrovne

Logging Levels

debug
level: 10

info
level: 20

warning
level: 30

error
level: 40

critical
level: 50

C:\Users\miros\AppData\Roaming\GitHub Desktop\logs\2024-02-03.desktop.production.log - Notepad++

Súbor Úpravy Hľadať Zobrazit Kódovanie Jazyk Nastavenia Nástroje Makro Spustiť Doplnky Okná ?

2024-02-03.desktop.production.log

```
1 2024-02-03T19:20:24.553Z - error: [main] Error looking for toast activator CLSID in shortcut C:\Users\miros\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\GitHub, Inc\GitHub Desktop.lnk
2 Error: Failed to read shortcut link
3   at ne (C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\lib\find-toast-activator-clsid.ts:38:29)
4   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\lib\find-toast-activator-clsid.ts:26:33
5   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\notifications.ts:28:32
6   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\main.ts:713:7
7   at Object.<anonymous> (C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\main.ts:713:7)
8   at Module._compile (node:internal/modules/cjs/loader:1271:14)
9   at Object..js (node:internal/modules/cjs/loader:1326:10)
10  at Module.load (node:internal/modules/cjs/loader:1126:32)
11  at node:internal/modules/cjs/loader:967:12
12  at Function._load (node:electron/js2c/asar_bundle:2:13327)
13 2024-02-03T19:20:24.566Z - error: [main] Error looking for toast activator CLSID in shortcut C:\Users\miros\Desktop\GitHub Desktop.lnk
14 Error: Failed to read shortcut link
15   at ne (C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\lib\find-toast-activator-clsid.ts:38:29)
16   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\lib\find-toast-activator-clsid.ts:26:33
17   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\notifications.ts:28:32
18   at C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\main.ts:713:7
19   at Object.<anonymous> (C:\Users\miros\AppData\Local\SquirrelTemp\webpack:\app\src\main-process\main.ts:713:7)
20   at Module._compile (node:internal/modules/cjs/loader:1271:14)
21   at Object..js (node:internal/modules/cjs/loader:1326:10)
22   at Module.load (node:internal/modules/cjs/loader:1126:32)
23   at node:internal/modules/cjs/loader:967:12
24   at Function._load (node:electron/js2c/asar_bundle:2:13327)
25 2024-02-03T19:20:24.566Z - error: [main] Toast activator CLSID not found in any of the shortcuts. Falling back to known CLSIDs.
26 2024-02-03T19:20:24.566Z - info: [main] Using toast activator CLSID {27D44D0C-A542-5B90-BCDB-AC3126048BA2}
27 2024-02-03T19:20:25.701Z - info: [main] Received possible protocol arguments: 2
28 2024-02-03T19:20:25.734Z - error: [main] Malformed launch arguments received:
C:\Users\miros\AppData\Local\GitHubDesktop\app-3.3.8\GitHubDesktop.exe,--squirrel-firstrun
29 2024-02-03T19:20:25.734Z - info: [main] Using toast activator CLSID {27D44D0C-A542-5B90-BCDB-AC3126048BA2}
30 2024-02-03T19:20:26.533Z - info: [ui] Opt in reported.
31 2024-02-03T19:20:26.575Z - info: [ui] [AppStore] loading 0 repositories from store
32 2024-02-03T19:20:26.701Z - info: [ui] launching: 3.3.8 (Windows 10.0.22631)
33 2024-02-03T19:20:26.702Z - info: [ui] execPath: 'C:\Users\miros\AppData\Local\GitHubDesktop\app-3.3.8\GitHubDesktop.exe'
34 2024-02-03T19:20:26.735Z - info: [ui] [Welcome] no sign in step found. ignoring ...
35 2024-02-03T19:20:26.894Z - info: [ui] [Welcome] no sign in step found. ignoring ...
36 2024-02-03T19:20:26.897Z - info: [ui] Executing installGlobalFFFilter: git lfc install skin none (took 1.012s)
```

Čo je Modul sys?

- Poskytuje **priamy prístup k systémovým parametrom a funkciám**
- Umožňuje **manipuláciu so štandardnými vstupmi a výstupmi**
- Práca s **argumentmi príkazového riadku, správou výnimiek a ukončenie programu**

1. Prístup k argumentom príkazového riadku (sys.argv)

python

 Kopírovať  Upraviť

```
import sys  
  
print("Argumenty programu:", sys.argv)
```

 `sys.argv` obsahuje zoznam argumentov zadaných pri spustení skriptu.

📌 Použitie:

- `sys.argv[0]` – názov spusteného skriptu
- `sys.argv[1:]` – argumenty odovzdané skriptu

Modul sys

2. Ukončenie programu (sys.exit())

python

 Kopírovať  Upraviť

```
import sys  
  
print("Program sa ukončí.")  
sys.exit(0) # 0 znamená úspešné ukončenie  
print("Tento riadok sa už nevykoná.")
```

 `sys.exit(kód)` ukončí program s návratovou hodnotou `kód`.

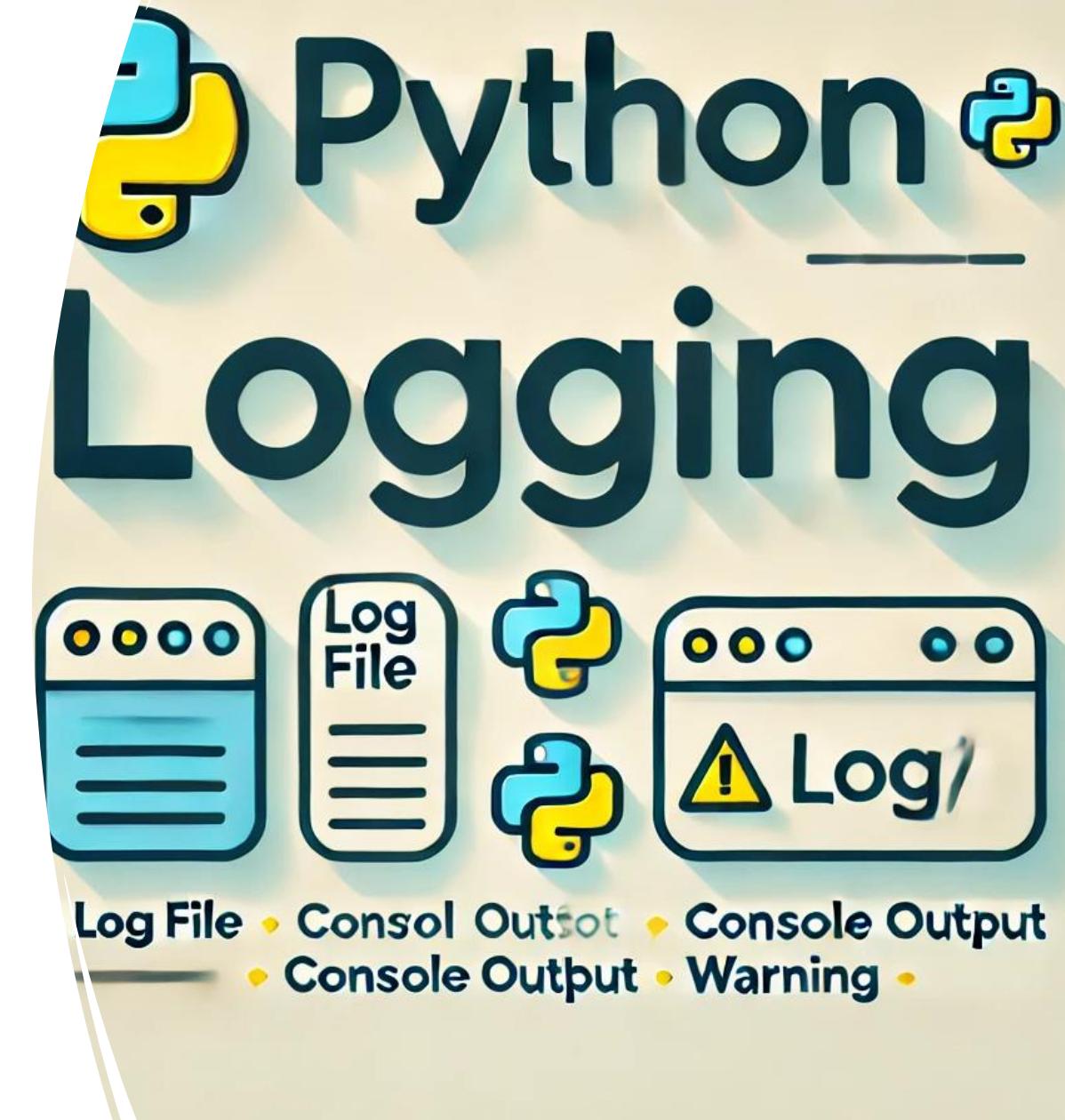
Kedy použiť sys?

1.  Pri práci s argumentmi príkazového riadku
 2.  Pri presmerovaní vstupov a výstupov
 3.  Pri správe pamäte a ukončení programu
- ◆ Alternatívy:
 - A. **argparse** – na pokročilú prácu s argumentmi príkazového riadku
 - B. **os** – na interakciu s operačným systémom

Úlohy

Logovanie

1. Čo je to modul Logging?
2. Čo sú to logy?
3. Aké existujú logovacie úrovne?
4. Čo je to modul sys?
5. Ako používať tieto moduly?
6. Aké sú vhodné prípady použitia?
7. Kde nájdeme dokumentáciu?



Dôležité Zdroje

Slovensko a Česko

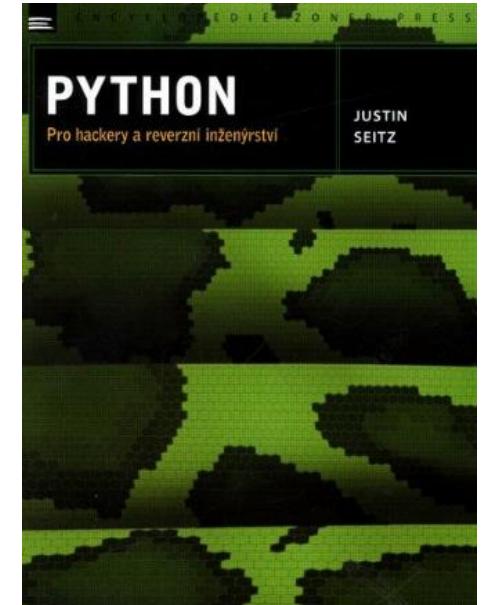
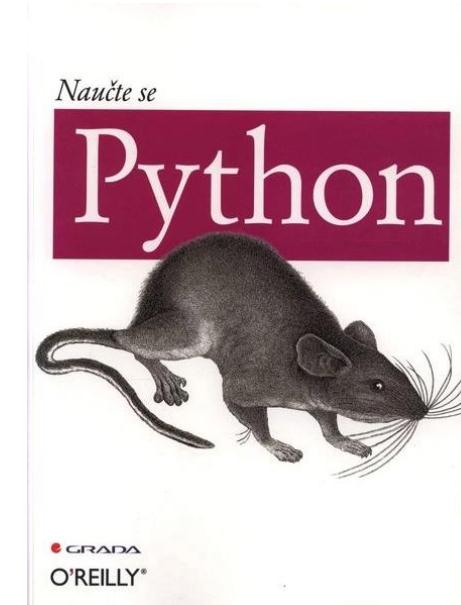
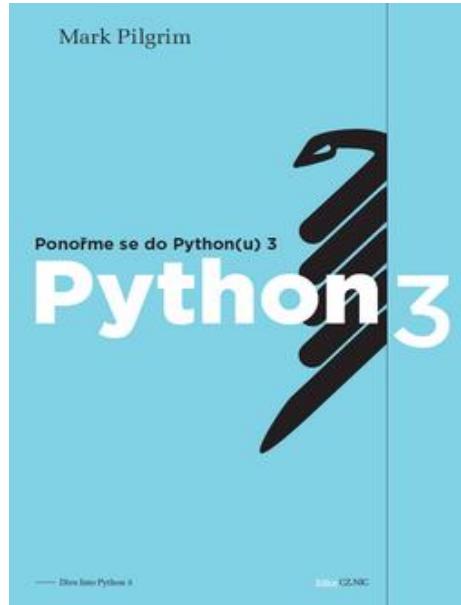
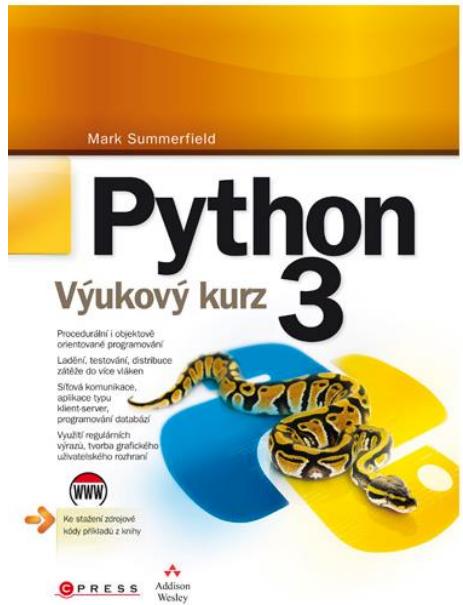
- Albatrosmedia
- Kopp
- Grada
- Wolters Kluwer
- BEN
- Veda

Zahraničie

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

YouTube Tutoriály

- [IT Academy](#)



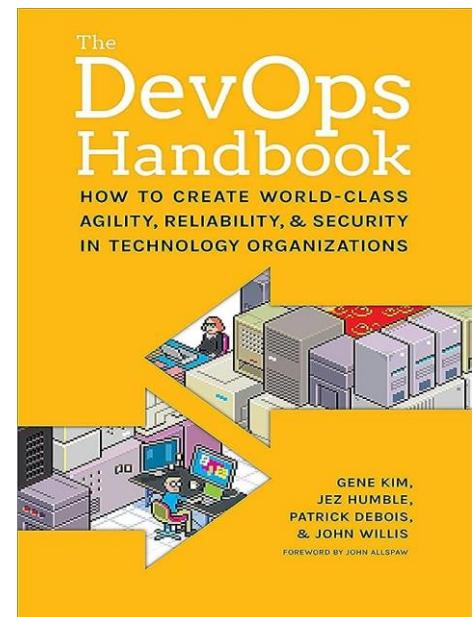
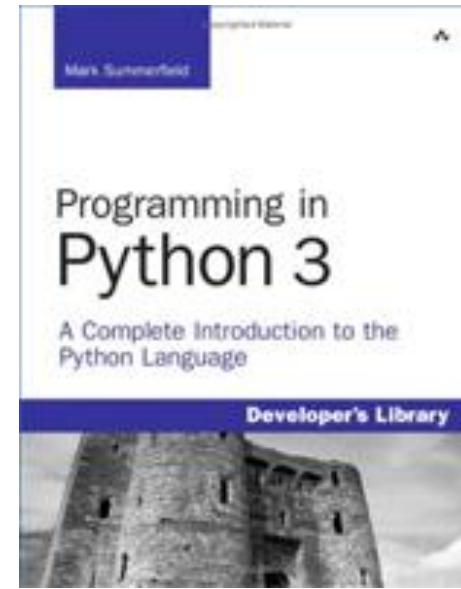
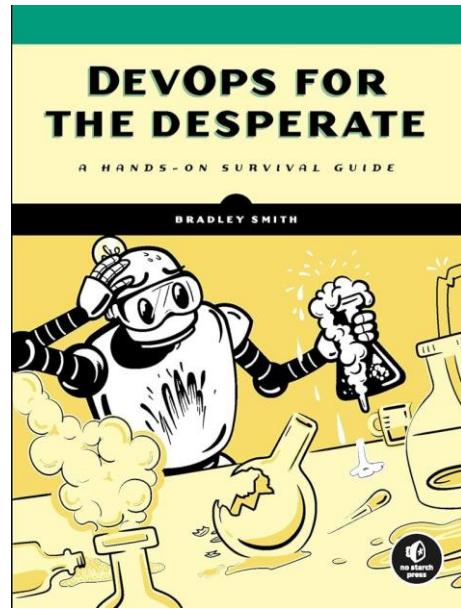
Čo sa oplatí/neoplatí prečítať SK/CZ?



Effective DevOps

BUILDING A CULTURE OF COLLABORATION,
AFFINITY, AND TOOLING AT SCALE

Jennifer Davis & Ryn Daniels



Čo sa oplatí/neoplatí prečítať EN?

Čo Odporúčam si Pozriet’?

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>



TOP

Klávesové Skratky



TOP 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

Efektívne Používanie Klávesnice

Špeciálne znaky, kde ich nájst' na klávesnici

The diagram shows a standard QWERTY keyboard with various keys highlighted in different colors to represent different categories of special characters:

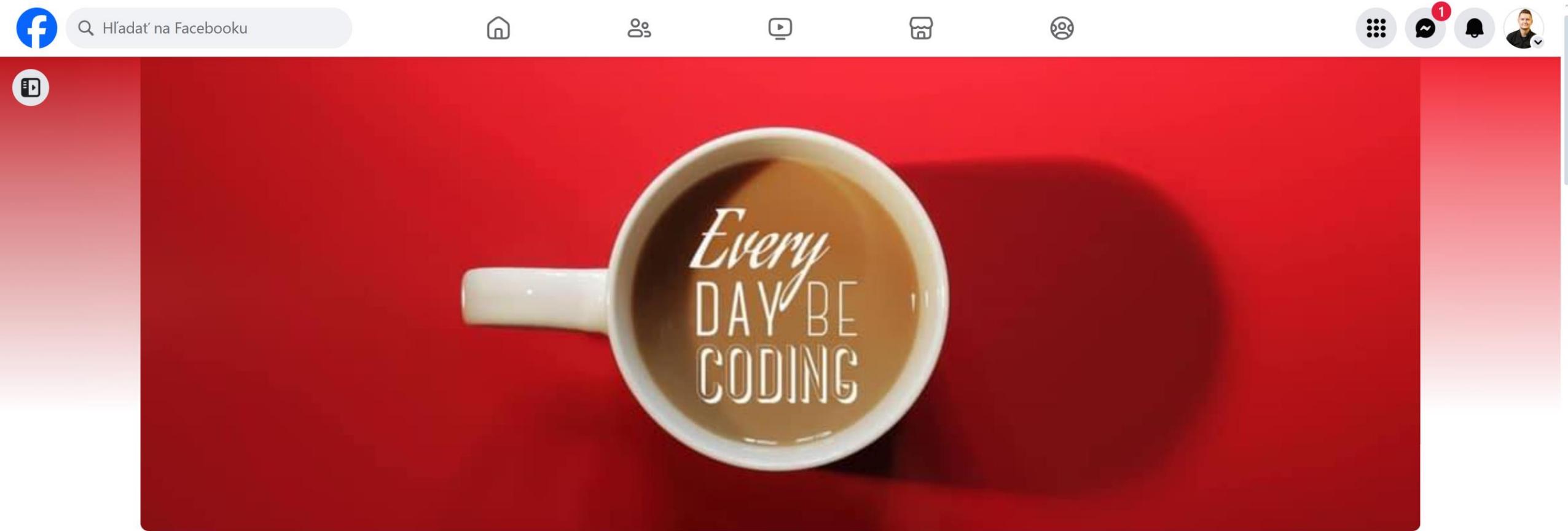
- Operátory (Operators):** +, *, -, /, %, @.
- Porovnávanie (Comparison):** <, =, !.
- Retazce (Text):** " (double quotes), \ (backslash).
- Oddelovače (Separators):**, ;, :.
- Bitové operácie (Bitwise Operations):** &, |, ^, ~, ?.
- Zátvorky (Brackets):** (,), {}, [].
- Poznámky (Annotations):** #, !, €.
- Komentár (Comment):** /*, */.
- Pomocník (Helper):** %, ^, _.

Legend:

- Grey:** Standard characters (e.g., A-Z, 0-9, QWERTY keys).
- Green:** Operators (+, *, /, %, @), Comparison operators (<, =, !), and Separators (., ;, :).
- Blue:** Bitwise operations (&, |, ^, ~, ?) and Brackets ((), {}, []).
- Red:** Text operators (" and \), Annotations (#, !, €), and Komentár (/* and */).
- Yellow:** Special characters (., ;, :, ?, !, €).
- Purple:** Poznámky (%, ^, _).

Large symbols on the right side of the diagram correspond to the highlighted characters:

- #
- &
- !
- €



Vývojári

Verejná skupina · 7,4 tis. členov

+ Pozvať

↗ Zdieľať

👤 Člen ▾

▼

Diskusia

Ludia

Podujatia

Médiá

Súbory



...



Napíšte niečo...



Reel



Fotka/video



Anketa

Informácie

Skupina softvérových vývojárov. Táto skupina by mala byť miestom, kde sa môžu českoslovení vývojári vzájomne spoznať, vyžiadať si konštrukívnu... [Zobrazit viac](#)

👤 Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.





Hľadať na Facebooku



```
52 </div>
53 </body>
54 <script type="text/javascript">
55 <!--
56 var currentImage = "bigImage1";
57 var pages = Math.ceil.photos.length / 9);
58 updatePages();
59 updateAllImages();
60 // document.getElementById('bigImage0').src = 'images/wieksza' + photos[page] + '1';
61 // document.getElementById('bigImage0').style.display = '';
62 changePhotoDescription('1');
63
64
65
66
67
68
69
70
71
72
73
```

Programátori

Verejná skupina · 11,1 tis. členov

+ Pozvat

Člen ▾



Diskusia

Vybrané

Ludia

Podujatia

Média

Súbory



Napíšte niečo...



Reel



Fotka/video



Anketa

Informácie

Táto skupina slúži na dohadzovanie si kšeftíkov a pre hľadačov programátorov / vývojárov.

Verejná

Členov skupiny a ich príspevky bude vidieť ktokoľvek.

Viditeľná

Príspevky sú viditeľní iba členom skupiny.



I am programmer



I have Life



I have
stackoverflow



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

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

[Popular](#) [Name](#) [New](#)

Ako Skončíme?

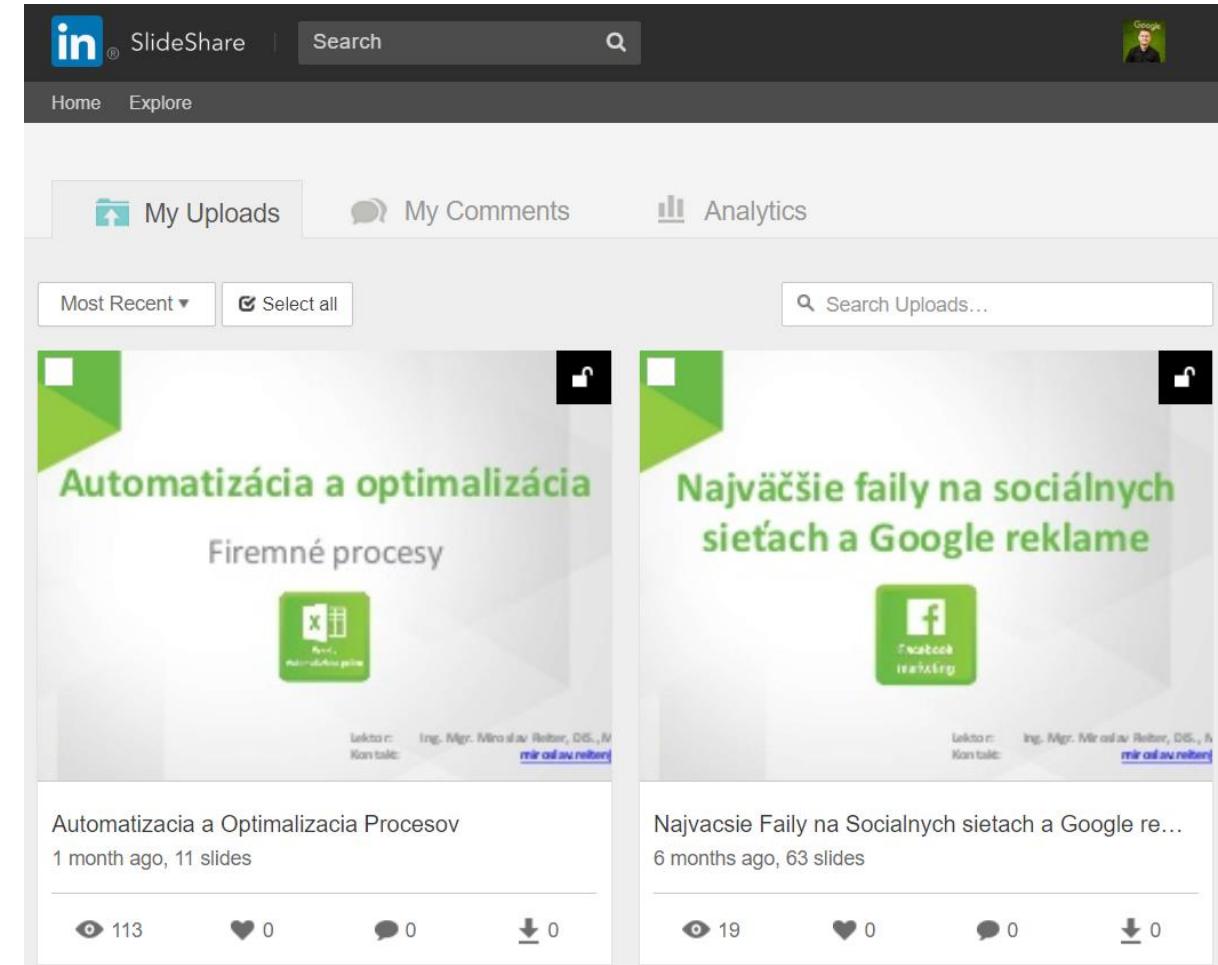
1. Stiahnite si Súbory

- https://github.com/miroslav-reiter/Kurzy_SAV_Analytika_Python_R

2. Pridajte si ma na LinkedIn

- www.linkedin.com/in/miroslav-reiter/

3. Prezentácia a materiály po prednáške



Vyber si online kurz

Nauč sa programovať, tvoriť webstránky a grafiku, manažovať alebo sa zameraj na osobný rozvoj. Všetko jednoducho vďaka našim online kurzom z pohodlia tvojho domova.

Ročné
predplatné na
všetky online
kurzy

~~2299.99€~~

399.99€

Prístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

🛒 Kúpiť teraz

Zadarmo

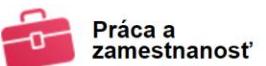
1. Kurzy NSCC/SAV

2. YouTube kanál VITA Academy

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

Platené

Moje kurzy na www.vita.sk



Práca a
zamestnanosť



Rodina a sociálna
pomoc



Sociálne poistenie
a dôchodkový
systém



Ministerstvo



[Úvod](#) > [Úvodná stránka](#) > [Informácie pre médiá](#) > [Aktuality](#) > Úrady práce sociálnych vecí a rodiny podporia rozvoj zručnosti

INŠPEKCIJA V SOCIÁLNYCH
VECIACH

Aktívne starnutie

Pomoc pre odídencov z Ukrajiny

Informácie pre médiá

Aktuality

Informácie pre verejnosť

Európsky orgán práce ELA

Pracovná mobilita cudzincov

Elektronické služby, formuláre a
žiadosti

Legislatíva

Organizácie ministerstva

Programy Európskej únie

Plán obnovy a odolnosti

Potravinová a základná
materiálna pomoc

Národné kompetenčné centrum
pre sociálne inovácie

Neprehliadnite

Podcasty o vývoji na trhu práce

Novačník milionára

19.03.2024

Úrady práce sociálnych vecí a rodiny podporia rozvoj zručnosti

Nový národný projekt Zručnosti pre trh práce prinesie už od 8.apríla možnosť bezplatne získať finančnú podporu na vzdelávanie či rekvalifikáciu. Je to prvý projekt svojho druhu, ktorý je určený pre ľudí bez práce, ale aj pre zamestnaných.

„Podobné projekty boli v minulosti obľúbené a fungovali, no vlane skončili. Jednou z našich priorit hned po nástupe na ministerstvo bolo opäťovne a v čo najkratšom čase podobný projekt opäť rozbehnuť. A tak sme pripravili národný projekt Zručnosti pre trh práce. Vzdelávanie je klúčovým nástrojom na podporu nezamestnaných pri ich ceste späť na trh práce, ako aj pre ľudí v ohrození stratou zamestnania. Tak ako pri ostatných nástrojoch, aj v tomto prípade sme urobili viac a priniesli inováciu s cieľom zvýšiť efektivitu a kvalitu služieb pre občanov a v jednom projekte spájame podporu pre ľudí bez práce aj zamestnaných. Okrem toho majú ľudia možnosť využiť tento projekt opakovane,“ povedal minister práce, sociálnych vecí a rodiny Erik Tomáš.

Cieľom nového národného projektu Zručnosti pre trh práce je podporiť uchádzačov o zamestnanie v snahе nájsť si prácu prostredníctvom rekvalifikácie. Projekt bude pôsobiť aj preventívne. Možnosť získať bezplatné vzdelávanie ponúkne aj zamestnaným záujemcom o zamestnanie, a to s cieľom ochrániť a zlepšiť ich postavenie na trhu práce. Projekt nie je určený pre samostatne zárobkovo činné osoby.

„Tento projekt sme nastavovali citlivо, na základe potrieb ľudí, ale aj údajov o trhu práce tak, aby bola pomoc čo najúčinnejšia. Väčší priestor a viac podpory sme v rámci neho dali ľuďom, ktorí čelia komplikovanejšej životnej situácii a lepšie vzdelanie u nich má väčší potenciál podstatne im zlepšiť kvalitu života. Aj preto sme kládli dôraz na znevýhodnených uchádzačov o zamestnanie a mladých ľudí do 30 rokov bez práce,“ dodal generálny riaditeľ a generálny tajomník služobného úradu Ústredia práce, sociálnych vecí a rodiny Peter Ormandy.

Žiadatelia si sami vyberú vzdelávanie a aj jeho poskytovateľa. Úrady práce, sociálnych vecí a rodiny budú akceptovať iba poskytovateľov pôsobiacich v oblasti vzdelávania najmenej jeden rok. O príspevok na vzdelávanie bude možné žiadať od 8.apríla na príslušnom úrade práce, sociálnych vecí a rodiny, ktorý každú žiadosť individuálne posúdi. Žiadosť o poskytnutie príspevku je potrebné podať najneskôr 30 kalendárnych dní pred začiatkom kurzu. Pri scénariu, že žiadost o poskytnutie príspevku je schvázená, poskytovateľ sa vzdelenie zaplatí vzdelávací kurz. V rámci tohto projektu je možné žiadať o poskytnutie príspevku už počas vzdelenia.

repas@itacademy.sk

Ako sa s nami Spojit'?



ADRESA: IT Academy, s. r. o.

Budova KOLOSEO prízemie
Tomášikova 50/A
831 04 Bratislava



WEB: www.it-academy.sk



E-MAIL: info@it-academy.sk



TELEFÓN: +421 917 095 406



Ako Vieme Pomôcť?

#Copywriting

#Školenia

#Zamestnanci

#Pomáhame

#Rast

#Projekty

#Certifikácie

#Kurzy

#Tréningy

#Vzdelávanie

#PPC Kampane

#Elearning

#Mentoring

#Konzultácie

#Online

#Programovanie

#Vývoj

#Marketing

#Reklama

#Prenájom Techniky

- Domov
- Shorts
- Odbory
- Moje
- História

Ak chcete k videám pridať označenie páci sa, komentovať alebo sa prihlásiť na odber, musíte sa prihlásiť.

[Prihlásiť sa](#)

Preskúmať

- Populárne
- Hudba
- Filmy
- Hry
- Sport

Viac zo služby YouTube

- YouTube Premium
- YouTube Music
- YouTube Kids
- Nastavenia
- História nahlasova...
- Pomocník
- Odoslať spätnú väz...










Naštartuj sa s nami v IT!



Online kurzy
www.vita.sk



Prezenčné kurzy
www.itacademy.sk



Daj Odber



Miroslav Reiter - VITA Academy

@VITA-Academy · 9,37 tis. odberateľov · 1,6 tis. videí

 Prednášajúci a garant kurzov: RNDr. PhDr. Ing. Mgr. et Mgr. Miroslav Reiter, DiS., MBA, ...viac

vita.sk a 10 ďalších odkazov

Odoberať

Pripojiť sa

Domov Videá Shorts Zoznamy Príspevky



Najnovšie Populárne Najstaršie



Online kurz Testovanie Softvéru s Katalon - Ako používať Katalon Recorder a exportova...

3 zhliadnutí · pred 30 minútami



Online kurz Microsoft Excel - Ako používať Finančné a Účtovné Funkcie PV, FV, NPV,...

97 zhliadnutí · pred 1 dňom



Online kurz Google Analytics 4 - Úvod a Predstavanie Kurzu

56 zhliadnutí · pred 2 dňami



Online kurz Jazyk Julia - Ako na HPC a Paralelizmus v Julia? Čo sú Vlákna...

31 zhliadnutí · pred 3 dňami



Online Kurz PostgreSQL - Ako používať Dátové Typy a Štandardné Funkcie v...

77 zhliadnutí · pred 4 dňami



Online kurz Manažerske Účtovníctvo a Podnikanie - Ako na Biznis Plán a Model?...

85 zhliadnutí · pred 5 dňami



Online kurz Python Bokeh - Čiarové grafy (Line charts) a Interaktívne Vizualizácie Dát

37 zhliadnutí · pred 6 dňami



Online kurz Testovanie Softvéru - Čo je BDD (Behavior-driven development), Cucumber,...

47 zhliadnutí · pred 7 dňami

Dajte odber na VITA Academy

[www.YOUTUBE.COM/@VITA-ACADEMY](https://www.youtube.com/@VITA-ACADEMY)