

Python

Databázy



Ako Začneme?

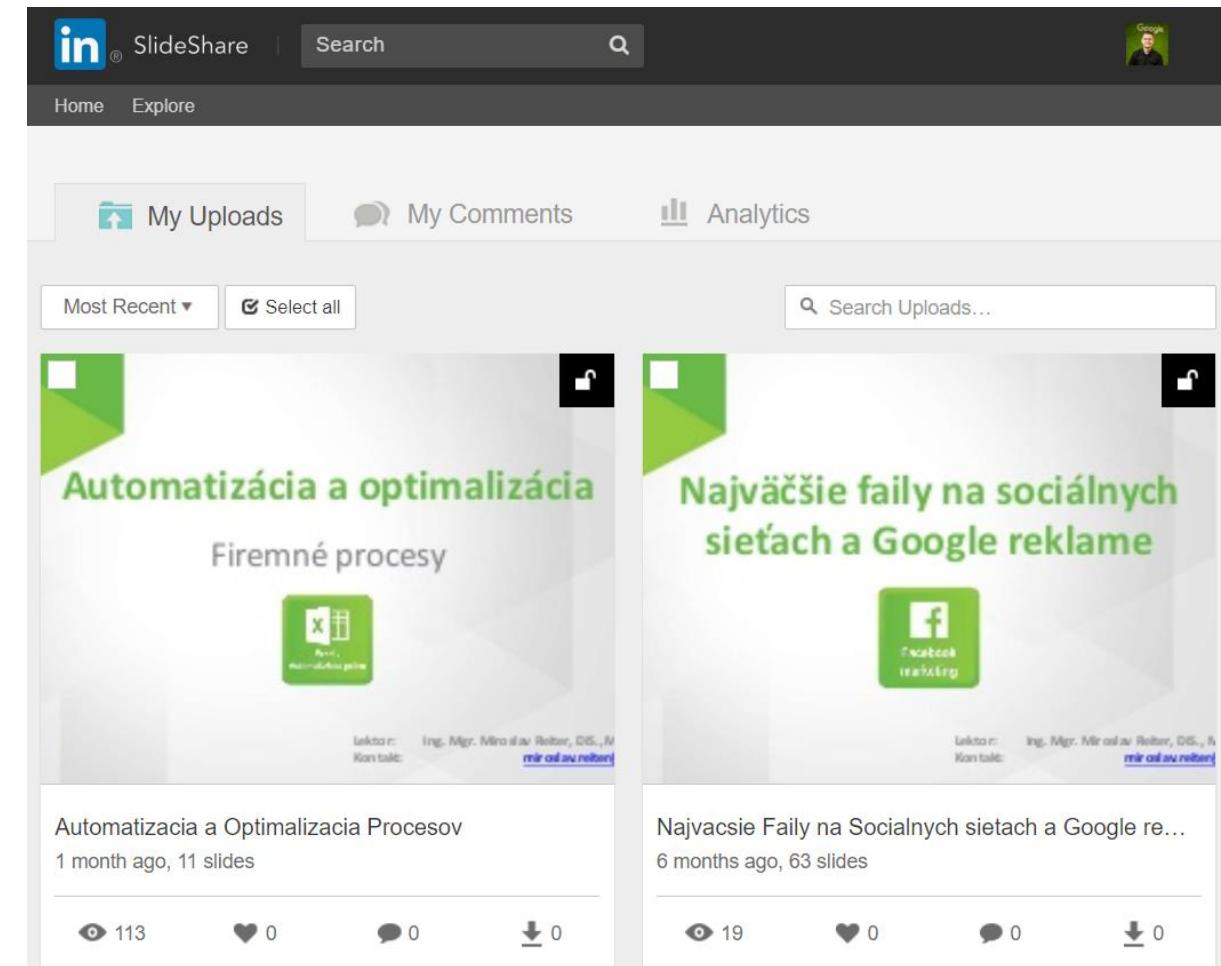
1. Pripojenie na DB/DBMS

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

2. Pridajte si ma na LinkedIn

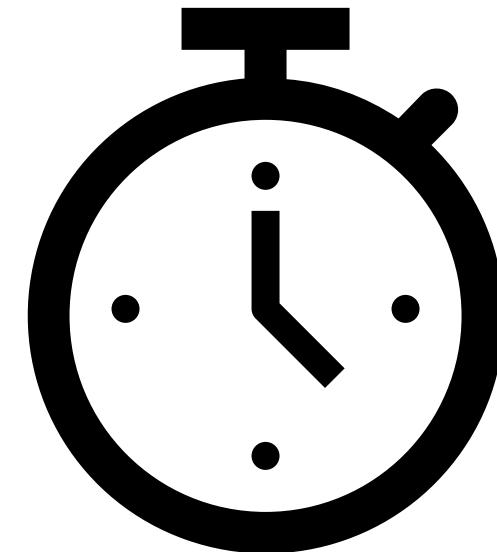
- www.linkedin.com/in/miroslav-reiter/

3. Prezentácia a materiály po prednáške



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



Programovací jazyk Python poskytuje viacero nástrojov a knižníc na prácu s SQL databázami. SQL je programovací jazyk určený na ukladanie, manipulovanie a vyberanie dát v relačných databázach. Počas kurzu preberieme základy databázových systémov a jazyku SQL a ich nasadenie v programoch vytvorených v Pythone.

1. Úvod do Databáz a Databázových Systémov

- Databázy, Tabuľky, Dopyty
- Datalore
- Knižnice a Moduly (pymysql, psycopg2, google-cloud)
- Secrets a Heslá
- Environs
- Pripojenie a odpojenie Databázy

2. Dopytovanie v SQL a Pythone

- Jazyk SQL a Dopyty (Queries)
- Tabuľky a Dataframes
- Výber dát a stĺpcov z tabuľky (Select)
- Dátové typy v DB a v Pythone
- Filtrovanie riadkov (Where)
- Agregačné funkcie (Sum, Min, Max, Avg, Count)
- Limity (Limit, Top, Offset)
- Zoskupenie riadkov (Group by)
- Logické operátory (And, Or, Not, Xor)
- Vytvorenie tabuľky (Create)
- Vymazanie tabuľky (Delete)
- Aktualizácia tabuľky (Update)

O mne - Miroslav Reiter

5

40000+
klientov a
1000+ firiem

IT Architekt
Programátor
Manažér

Microsoft
Google
ISTQB tréner

134
certifikácií

151 príručiek
a publikácií

13 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



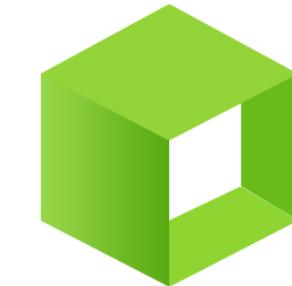
Miroslav Reiter

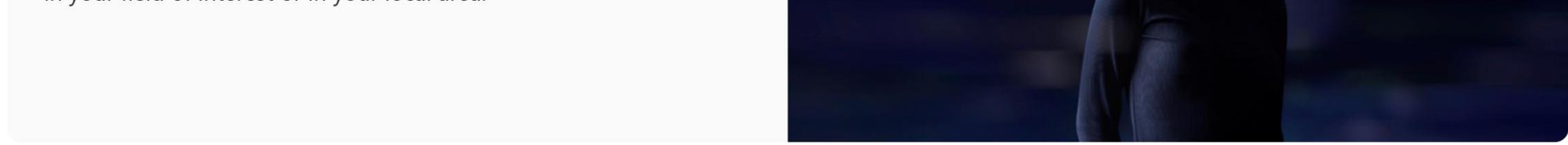


1. PhDr. VŠM (Podnikovný manažment)
2. Ing. STU FEI (Aplikovaná informatika)
3. Mgr. UK FM (Strategický manažment a marketing)
4. Mgr. VŠM (Manažérstvo kvality)
5. Mgr. VŠEMVŠ (Verejná správa)
6. Mgr. DTI (Učiteľstvo ekonomických predmetov)
7. DiS. AMOS (Cestovný ruch)
8. MBA LIGS (Executive management)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



DIGITÁLNA
UNIVERZITA





Filter

[People](#) [Events](#)

[Search by filter](#)

- > Program
- > Country/Region
- > State/Province
- > Language
- > Award Category
- > Technology Area

Search Profiles

[Search Profiles](#)

[Search](#)

[Most Valuable Professionals](#)

[Country/Region: Slovakia](#)

Most Valuable Professionals



[Miroslav Reiter](#)

IT Architect and Programmer Hard worker Lecturer and Certified Trainer

Slovakia

Most Valuable Professionals



[Peter Belko](#)

Content Developer, Technical Writer, Consultant, Trainer for M365 + Office apps

Slovakia

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.

Miroslav Reiter

získava status
Google Certified Trainer

Automation

Google



Microsoft Certified
Trainer

MIROSLAV REITER

Has successfully completed the requirements to be recognized as a Microsoft Certified Trainer

N. S.
Satya Nadella
Chief Executive Officer

Microsoft
CERTIFIED
Trainer

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 €**

Prístup pre Vás do všetkých Aktuálnych aj Pripravovaných Online Kurzov

12 mesačná platnosť

Kúpiť teraz[Zistíť viac](#)

529 kurzov v ponuke



Zábavné online lekcie



Akreditované kurzy



12 rokov skúseností



Certifikovaní profesionálni lektori

Odporučame Kurzy špeciálne pre vás



Online kurz SQL I.

Začiatočník

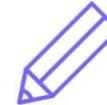
110,00€ 142,00€



Online kurz SAP I.

Začiatočník

224,00€ 292,50€



Online kurz Copywriting I.

Začiatočník

108,00€ 115,00€



Balík SAP Profesionál

998,80€ 998,80€



Online kurz Digitálny Marketing

146,00€ 199,00€



Online kurz Facebook Marketing I. Začiatočník

104,00€ 142,00€

Chat



Luigi, Mário
a Yoshi

Čo Robíte?

1. Študent/učiteľ'

2. Zamestnanec

3. Podnikateľ'

4. Nezamestnaný/materská

5. Dievča pre všetko



Vaše Ciele a Očakávania

16

1. Doplniť si znalosti z jazyka Python

2. Základy analytického myslenia

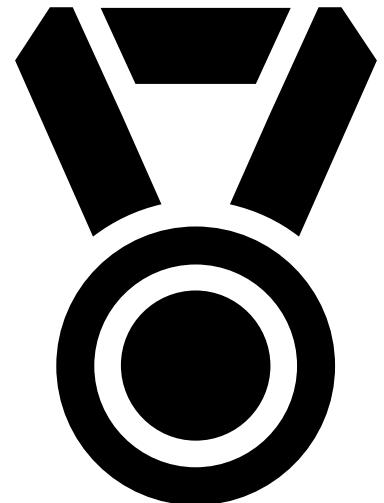
3. Základy spracovania dát v Databázach

4. Doplniť si znalosti z programovania

5. Doplniť si znalosti z vývojových prostredí (IDE)

6. Využitie AI pri programovaní

Zábava je v zaručená v každom bode :-)



Ako sa pripojiť na Databázy v Python?



Databázy Python



Knižnice/Moduly pre DB

1. **sqlite3**
2. **pymysql**
3. **mysql-connector-python**
4. **psycopg2**
5. **cx_Oracle**
6. **pyodbc (MS Access)**
7. **sqlalchemy (ORM)**



Table of Contents

sqlite3 — DB-API 2.0
interface for SQLite
databases

- Tutorial
- Reference
 - Module functions
 - Module constants
 - Connection objects
 - Cursor objects
 - Row objects
 - Blob objects
 - PrepareProtocol objects
 - Exceptions
 - SQLite and Python types
 - Default adapters and converters (deprecated)
 - Command-line interface
- How-to guides
 - How to use placeholders to bind values in SQL queries

sqlite3 — DB-API 2.0 interface for SQLite databases

Source code: [Lib/sqlite3/](#)

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The `sqlite3` module was written by Gerhard Häring. It provides an SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#), and requires SQLite 3.7.15 or newer.

« This document includes four main sections:

- [Tutorial](#) teaches how to use the `sqlite3` module.
- [Reference](#) describes the classes and functions this module defines.
- [How-to guides](#) details how to handle specific tasks.
- [Explanation](#) provides in-depth background on transaction control.

See also:



Search projects



itacademy ▾

mysql-connector-python 8.2.0

pip install mysql-connector-python



[Latest version](#)

Released: Oct 25, 2023

MySQL driver written in Python

Navigation

Project description

Release history

Download files

Project description

MySQL driver written in Python which does not depend on MySQL C client libraries and implements the DB API v2.0 specification (PEP-249).

Project links

[Homepage](#)



Search projects



Help

Sponsors

Log in

Register

pymysql 1.0.3

pip install pymysql



[Latest version](#)

Released: Mar 28, 2023

Pure Python MySQL Driver

Navigation

[Project description](#)

[Release history](#)

[Download files](#)

Project links

[Documentation](#)

[Project](#)

Project description

docs passing coverage 87%

PyMySQL

This package contains a pure-Python MySQL client library, based on [PEP 249](#).

Requirements

- Python -- one of the following:
 - [CPython](#): 3.7 and newer
 - [PyPy](#): Latest 3.x version



Search projects



Help

Sponsors

Log in

Register

psycopg2 2.9.6

pip install psycopg2



[Latest version](#)

Released: Apr 3, 2023

psycopg2 - Python-PostgreSQL Database Adapter

Navigation

Project description

Release history

Download files

Project links

Homepage

Code

Documentation

Project description

Psycopg is the most popular PostgreSQL database adapter for the Python programming language. Its main features are the complete implementation of the Python DB API 2.0 specification and the thread safety (several threads can share the same connection). It was designed for heavily multi-threaded applications that create and destroy lots of cursors and make a large number of concurrent “INSERT”s or “UPDATE”s.

Psycopg 2 is mostly implemented in C as a libpq wrapper, resulting in being both efficient and secure. It features client-side and server-side cursors, asynchronous communication and notifications, “COPY TO/COPY FROM” support. Many Python types are supported out-of-the-box and adapted to matching PostgreSQL data types; adaptation can be extended and customized thanks to a flexible objects adaptation system.

Psycopg 2 is both Unicode and Python 3 friendly.

Documentation



Search projects



Help

Sponsors

Log in

Register

ibm-db 3.2.3

pip install ibm-db



Latest version

Released: Mar 13, 2024

Python DBI driver for DB2 (LUW, zOS, i5) and IDS

Navigation

Project description

Release history

Download files

Project description

Python DBI driver for IBM Db2 for LUW, IBM Informix, IBM Db2 for iSeries(AS400) and IBM Db2 for z/OS servers

Verified details

These details have been verified by PyPI

Maintainers



[opendev](#)

Unverified details



Search projects



Help Sponsors Log in Register

pymongo 4.7.2

pip install pymongo



Latest version

Released: May 7, 2024

Python driver for MongoDB <<http://www.mongodb.org>>

Navigation

Project description

Release history

Download files

Verified details

These details have been verified by PyPI

Maintainers



[10gen](#)

Unverified details

*These details have **not** been verified by PyPI*

Project links

- [Documentation](#)
- [Homepage](#)

Project description

PyMongo

[pypi v4.7.2](#) [python 3.7 | 3.8 | 3.9 | 3.10 | 3.11 | 3.12](#) [downloads/month 27M](#) [docs passing](#)

About

The PyMongo distribution contains tools for interacting with MongoDB database from Python. The `bson` package is an implementation of the [BSON format](#) for Python. The `pymongo` package is a native Python driver for MongoDB. The `gridfs` package is a [gridfs](#) implementation on top of `pymongo`.

PyMongo supports MongoDB 3.6, 4.0, 4.2, 4.4, 5.0, 6.0, and 7.0.

Support / Feedback

For issues with, questions about, or feedback for PyMongo, please look into our [support channels](#). Please do not email any of the PyMongo developers directly with issues or questions - you're more likely to get an answer on [StackOverflow](#) (using a "mongodb" tag).

Bugs / Feature Requests



Search this Manual

 [Documentation Home](#)

MySQL Connector/Python Developer Guide

- [Preface and Legal Notices](#)
- [Introduction to MySQL Connector/Python](#)
- [Guidelines for Python Developers](#)
- [Connector/Python Versions](#)
- [Connector/Python Installation](#)
- [Connector/Python Coding Examples](#)
- [Connector/Python Tutorials](#)
- [Connector/Python Connection Establishment](#)
- [The Connector/Python C Extension](#)
- [Connector/Python Other Topics](#)
- [Connector/Python API Reference](#)
- [Connector/Python C Extension API Reference](#)
- [Index](#)

 [Related Documentation](#)

MySQL Connector/Python Developer Guide

Abstract

This manual describes how to install and configure MySQL Connector/Python, a self-contained Python driver for communicating with MySQL servers, and how to use it to develop database applications.

The latest MySQL Connector/Python version is recommended for use with MySQL Server version 5.7 and higher.

For notes detailing the changes in each release of Connector/Python, see [MySQL Connector/Python Release Notes](#).

For legal information, see the [Legal Notices](#).

For help with using MySQL, please visit the [MySQL Forums](#), where you can discuss your issues with other MySQL users.

Licensing information. This product may include third-party software, used under license. If you are using a Commercial release of MySQL Connector/Python, see the [MySQL Connector/Python 8.2 Commercial License Information User Manual](#) for licensing information, including licensing information relating to third-party software that may be included in this Commercial release. If you are using a Community release of MySQL Connector/Python, see the [MySQL Connector/Python 8.2 Community License Information User Manual](#) for licensing information, including licensing information relating to third-party software that may be included in this Community release.

Document generated on: 2023-11-22 (revision: 77308)

Table of Contents

[Preface and Legal Notices](#)[1 Introduction to MySQL Connector/Python](#)[1.1 MySQL Connector/Python](#)

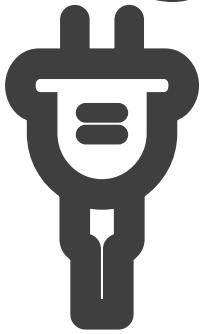
Úlohy Python DB

1. Prehľad knižnice/moduly pre DBMS
2. Pochopenie modulov pre pripojenie sa k DB
3. PIP a import knižnice



Pripojenie

DB



Proces Pripojenie sa na DB v Pythone

1. Inštalácia potrebnej knižnice
2. Importovanie knižnice
3. Vytvorenie pripojenia
4. Vytvorenie kurzora
5. Vykonávanie SQL príkazov
6. Zatvorenia pripojenia a kurzov



Pripojenie na DB

30

MySQL

- DB_MySQL - Ucime-sa-sql-123
- Databázový server (Host): mysql57.r2.websupport.sk
- Name (DB): DB_MySQL
- Login (User): DB_MySQL
- Heslo (Password): Ucime-sa-sql-123
- Port: 3311
- Socket: /tmp/mysql57.sock

PostgreSQL

- DB_PostgreSQL - Ucime-sa-sql-123
- Databázový server: postgresql.r2.websupport.sk
- Meno: DB_PostgreSQL
- Login: DB_PostgreSQL
- Heslo (Password): Ucime-sa-sql-123
- Port: 5432
- Socket: host=localhost



IT_Academy_VITA_U MR

100 GB, 10 domén

Prehľad

Faturačné nastavenia

Balík služieb

Správa domény

vita.sk



Prehľad

Faturačné nastavenia



DNS



Web



FTP účty

CMS



Emaily



Databázy



Zoznam



Informácie o používateľovi

Používateľia



Prihlasovacie údaje pre používateľa DB_MySQL databázy DB_MySQL

Nastaviť poznámku

Prihlasovacie údaje

Meno DB_MySQL

Login DB_MySQL

Heslo Heslo k účtu nie je možné zistíť dodatočne.

[Zabudnuté heslo?](#)

Odporučané pripojenie

(funguje všade)

Databázový server mysql57.r2.websupport.sk

Port 3311

Pripojenie cez socket

(funguje len na našich serveroch)

Socket /tmp/mysql57.sock

localhost

(funguje len na našich serveroch)

Nastavenia predvoleného databázového servera pre localhost.

Môžete to zmeniť v [mysql connect nastavení](#)

Pomoc a podpora

Správa databázy

PhpMyAdmin

- <https://dbadmin.r2.websupport.sk/>
- môžete si vytvoriť vlastnú "skratku" v sekcií Služby v rámci Web nastavení

Adminer

- <https://adminer.r2.websupport.sk/>

 [Príklad použitia v PHP skriptoch s mysqli_connect](#) [Príklad použitia v PHP skriptoch s PDO pripojením](#)



IT_Academy_VITA_U MR

100 GB, 10 domén

Prehľad

Fakturačné nastavenia

Balík služieb

Správa domény

vita.sk



Prehľad

Fakturačné nastavenia



DNS



Web



FTP účty

CMS



Emaily



Databázy



Zoznam



Informácie o používateľovi

Používateľia



Prihlasovacie údaje pre používateľa DB_PostgreSQL databázy DB_PostgreSQL

Nastaviť poznámku

Prihlasovacie údaje

Meno DB_PostgreSQL

Login DB_PostgreSQL

Heslo Heslo k účtu nie je možné zistíť dodatočne.

[Zabudnuté heslo?](#)

Odporučané pripojenie

(funguje všade)

Databázový server postgresql.r2.websupport.sk

Port 5432

Pripojenie cez socket



(funguje len na našich serveroch)

Socket host=localhost

localhost

(funguje len na našich serveroch)

Nastavenia predvoleného databázového servera pre localhost.

Môžete to zmeniť v [mysqli connect nastavení](#)

Pomoc a podpora

Správa databázy

Adminer

- <https://adminer.r2.websupport.sk/>

[Príklad použitia v PHP skriptoch s pg_connect](#) [Príklad použitia v PHP skriptoch s PDO pripojením](#)



Search notebooks



MR

Home ▾

Your notebooks

Recent activity

Shared with you

Your favorites

Trash

Workspace files

Reports

Scheduled notebooks

Databases

Cloud storages

Environment variables

Settings

+ New notebook

Data

< Change type

Display name

MySQL database

General SSH Schemas Advanced

Connection type: default

Host

mysql57.r2.websupport.sk

Port

3311

Authentication

User & Password

User

DB_SQL

Read about how to use database connections in the [documentation](#).
To report a missing database type, leave feedback, or ask any questions, please contact us.

Password

.....

Database

DB_SQL

Save and close

Cancel

✓ Succeeded

Test connection

+ New connection



Attached data

	MySQL Test DB Kurz 2023	...	<
	Connect SQL cell ▾	MySQL	1 of 2
▼	DB MySQL		
▼	tables	9	
>	DEPT		
>	EMP		
>	nobel		
>	Studenti		
>	T1		
>	T10		
>	T100		
>	world		
>	Znamky		
▼	Server Objects		
>	collations	222	
>	users	1	
>	virtual views		

[1] ▶ 8.6s SOURCE MySQL Test DB Kurz 2023 ▾ RESULT SAVED TO df_1 AI + ⚡ ...

1 show tables;

Table Raw Visualize Statistics

	Tables_in_D...
0	DEPT
1	EMP
2	Studenti
3	T1
4	T10
5	T100
6	Znamky
7	nobel
8	world

9 rows x 2 columns ⌂ ⌂ Jump to top ⌂ Jump to bottom ⌂ ⌂ ⌂

Add code cell

Sheet

+

Databázy a Python

python

 Copy code

```
import pymysql

# Parametre pripojenia
host = "meno_hostitela"
port = 3306
user = "vase_meno"
password = "vaso_heslo"
db = "nazov_databazy"

# Vytvorenie pripojenia
conn = pymysql.connect(host=host, port=port, user=user, passwd=password)

# Vytvorenie kurzora
cursor = conn.cursor()

# Vykonávanie SQL príkazu
cursor.execute("SELECT * FROM vase_tabulka")

# Získanie výsledkov
rows = cursor.fetchall()

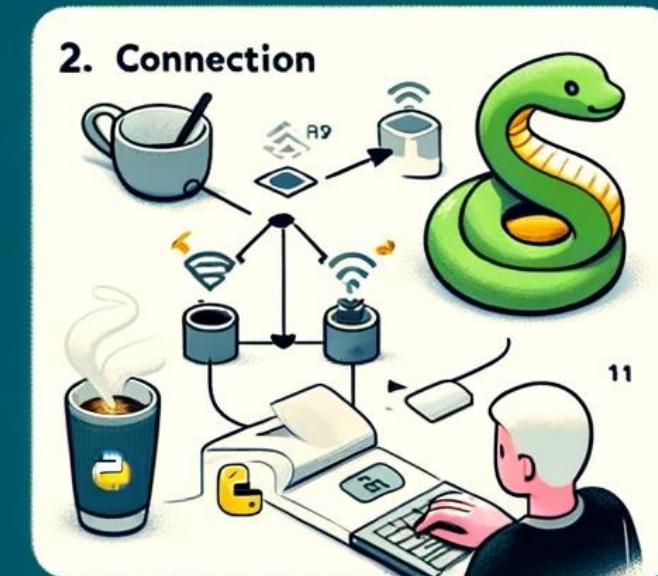
for row in rows:
    print(row)

# Zavretie kurzora a pripojenia
cursor.close()
conn.close()
```

Úlohy Pripojenie



1. Pripojenie sa na rôzne DBMS
2. Prehľad pripojení
3. DEMO: Pripojenia DB MySQL
4. DEMO: Pripojenia DB PostgreSQL
5. DEMO: Pripojenia DB SQLite



Databázy



Table of Contents

sqlite3 — DB-API 2.0
interface for SQLite
databases

- Tutorial
- Reference
 - Module functions
 - Module constants
 - Connection objects
 - Cursor objects
 - Row objects
 - Blob objects
 - PrepareProtocol objects
 - Exceptions
 - SQLite and Python types
 - Default adapters and converters (deprecated)
 - Command-line interface
- How-to guides
 - How to use placeholders to bind values in SQL queries

sqlite3 — DB-API 2.0 interface for SQLite databases

Source code: [Lib/sqlite3/](#)

SQLite is a C library that provides a lightweight disk-based database that doesn't require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It's also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The `sqlite3` module was written by Gerhard Häring. It provides an SQL interface compliant with the DB-API 2.0 specification described by [PEP 249](#), and requires SQLite 3.7.15 or newer.

« This document includes four main sections:

- [Tutorial](#) teaches how to use the `sqlite3` module.
- [Reference](#) describes the classes and functions this module defines.
- [How-to guides](#) details how to handle specific tasks.
- [Explanation](#) provides in-depth background on transaction control.

See also:

SQLite a Python Typy

Python type	SQLite type
None	NULL
<u>int</u>	INTEGER
<u>float</u>	REAL
<u>str</u>	TEXT
<u>bytes</u>	BLOB

SQLite type	Python type
NULL	None
INTEGER	<u>int</u>
REAL	<u>float</u>
TEXT	depends on <u>text_factory</u> , <u>str</u> by default
BLOB	<u>bytes</u>

Table of Contents

`sqlite3` — DB-API 2.0 interface for SQLite databases

- Tutorial
- Reference
- Module functions
- Module constants
- Connection objects
- Cursor objects
- Row objects
- Blob objects
- PrepareProtocol objects
- Exceptions
- SQLite and Python types
- Default adapters and converters (deprecated)
- Command-line interface

- How-to guides
- How to use placeholders to bind values in SQL queries

- How to adapt custom Python types to SQLite values

- How to write adaptable objects
- How to register adapter callables

- How to convert SQLite values to custom Python types

- Adapter and converter recipes

- How to use connection shortcut methods

- How to use the connection context manager

- How to work with SQLite URLs

- How to create

Tutorial

In this tutorial, you will create a database of Monty Python movies using basic `sqlite3` functionality. It assumes a fundamental understanding of database concepts, including [cursors](#) and [transactions](#).

First, we need to create a new database and open a database connection to allow `sqlite3` to work with it. Call [`sqlite3.connect\(\)`](#) to create a connection to the database `tutorial.db` in the current working directory, implicitly creating it if it does not exist:

```
import sqlite3
con = sqlite3.connect("tutorial.db")
```

The returned [Connection](#) object `con` represents the connection to the on-disk database.

In order to execute SQL statements and fetch results from SQL queries, we will need to use a database cursor.

Call [`con.cursor\(\)`](#) to create the [Cursor](#):

```
cur = con.cursor()
```

Now that we've got a database connection and a cursor, we can create a database table `movie` with columns for title, release year, and review score. For simplicity, we can just use column names in the table declaration – thanks to the [flexible typing](#) feature of SQLite, specifying the data types is optional. Execute the `CREATE TABLE` statement by calling [`cur.execute\(...\)`](#):

```
cur.execute("CREATE TABLE movie(title, year, score)")
```

We can verify that the new table has been created by querying the `sqlite_master` table built-in to SQLite, which should now contain an entry for the `movie` table definition (see [The Schema Table](#) for details). Execute that query by calling [`cur.execute\(...\)`](#), assign the result to `res`, and call [`res.fetchone\(\)`](#) to fetch the resulting row:

```
>>> res = cur.execute("SELECT name FROM sqlite_master")
>>> res.fetchone()
('movie',)
```

We can see that the table has been created, as the query returns a [tuple](#) containing the table's name. If we query `sqlite_master` for a non-existent table `spam`, `res.fetchone()` will return `None`:

```
>>> res = cur.execute("SELECT name FROM sqlite_master WHERE name='spam'")
>>> res.fetchone() is None
True
```

Now, add two rows of data supplied as SQL literals by executing an `INSERT` statement, once again by calling [`cur.execute\(...\)`](#):

DL Variable viewer

File Tools Kernel View Code Run Help

Build report Run all MR

Python Databazy_2024

Úvod do Databáz a ... SQL Pripojenie DB Kurzory DB a Pandas Environs a Secrets SQLAlchemy + ▾

[2]

```
1 import sqlite3
2 pripojenie = sqlite3.connect("nova.db")
3
4 kurzor = pripojenie.cursor()
5 kurzor.execute("CREATE TABLE IF NOT EXISTS
6 filmy(nazov, rok, hodnotenie)")
7
```

<sqlite3.Cursor at 0x7f79c80c4c00>

[3] ▶ 0.0s

```
1 zdroj = kurzor.execute("SELECT name FROM
2 sqlite_master")
3 zdroj.fetchone()
```

('filmy',)

Idle CPU RAM 0% 2 GB

Calculated In process Errors 2 0 0

CPU S

Insert code cell More

...

Objekt sqlite3.Connection

- 1. `in_transaction`** je atribút typu boolean, ktorý indikuje, či je spojenie aktuálne v transakcii
- 2. `isolation_level`** určuje úroveň izolácie transakcií pre DB spojenie
- 3. `row_factory`** umožňuje nastaviť funkciu alebo triedu, ktorá transformuje riadky výsledkov získané z dopytu
- 4. `total_changes`** vráti celkový počet riadkov databázy, ktoré boli zmenené, vložené alebo vymazané od momentu, keď bolo DB spojenie otvorené



Desktop Version control

Database Explorer

console sqlite_master

WHERE ORDER BY

type name tbl_name rootpage sql

1	table	novi_zamestnanci	novi_zamestnanci		2 CREATE TABLE novi_zamestnanci (id_zamestnanec INTEGER PRIMARY KEY, cele_meno TEXT NOT NULL, vek INTEGER NOT NULL, mzda REAL, datum_narodenia DATE, oddelenie TEXT)
2	table	projekty	projekty		5 CREATE TABLE projekty (id_projekt INTEGER PRIMARY KEY, nazov TEXT NOT NULL, ...)
3	view	vyssie_mzdy	vyssie_mzdy		6 CREATE VIEW vyssie_mzdy AS SELECT cele_meno, mzda, oddelenie FROM novi_zamestnanci WHERE ...
4	view	mladi_zamestnanci	mladi_zamestnanci		7 CREATE VIEW mladi_zamestnanci AS SELECT cele_meno, vek, oddelenie FROM novi_zamestnanci ...
5	index	idx_cele_meno	novi_zamestnanci		8 CREATE INDEX idx_cele_meno ON novi_zamestnanci (cele_meno)
6	index	idx_vek	novi_zamestnanci		9 CREATE INDEX idx_vek ON novi_zamestnanci (vek)
7	index	idx_mzda	novi_zamestnanci		10 CREATE INDEX idx_mzda ON novi_zamestnanci (mzda)
8	index	idx_oddelenie	novi_zamestnanci		11 CREATE INDEX idx_oddelenie ON novi_zamestnanci (oddelenie)
9	index	idx_nazov	projekty		
10	index	idx_id_zamestnanec	projekty		

Files

Desktop C:\Users\miros\Desktop

Dokumenty

!IES Databázy.Ink

!Podpora Rozvoja Zručnosti Zamestnancov.Ink

!RNDR.Ink

!SAV AI, Python, SQL a R.Ink

!UK FM PhD.Ink

!Zručnosti pre Trh Práce.Ink

2024-05-21 20_14_17-Greenshot.prn

2024-05-21 20_16_35-Greenshot.prn

2024-05-21 20_32_04-Greenshot.prn

DB_Projekty_Zamestnanci.db

DB_Projekty_Zamestnanci_Indexy.db

DB_Skola.db

desktop.ini

Docker Desktop.Ink

Dropbox.Ink

Effect House.Ink

Financie.Ink

GitHub Desktop.Ink

IKT Architekt.Ink

IntelliJ IDEA Ultimate.Ink

Marketing.Ink

music_for_programming.Ink

Nmap - Zenmap GUI.Ink

OFFEO.Ink

OneDrive.Ink

OneDrive IT Academy.Ink

Python Databázy_2024.ipynb

Stiahnuté súbory.Ink

TODO.Ink

VITA.Ink

~\$ičenia - Umelá Inteligencia OK.docx

~\$Meranie-Výkonu.xlsx

Šablóny, Odpovede, FAQ.Ink

Škola.Ink

Scratches and Consoles

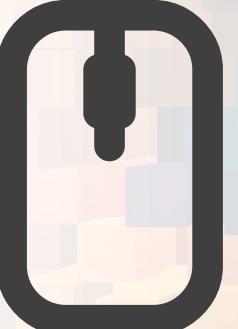
Database > DB_Projekty_Zamestnanci_Indexy > main > tables > sqlite_master

Úlohy DB SQLite

1. Pochopenie čo je DBMS SQLite
2. Používanie databázy SQLite
3. Používanie knižnice sqlite3



Databázové Kurzory



Fetching



DB Kurzory

- Nástroje používané pri práci s DB
- Je v podstate **ukazovateľ**, ktorý umožňuje **programátorovi vykonávať operácie na dátach v DB** ako sú **čítanie, aktualizácia, vkladanie alebo mazanie záznamov**
- Pri používaní Pythonu na prácu s DBMS ako sú napríklad SQLite, MySQL, PostgreSQL alebo iné, programátori najprv vytvoria spojenie s DB pomocou vhodného DB adaptéra
- Po vytvorení spojenia s DB môžu pomocou metódy **cursor()** vytvoriť kurzor
- Tento **kurzor** potom môžu používať na vykonávanie **SQL príkazov**

Table of Contents

[sqlite3 — DB-API 2.0 interface for SQLite databases](#)

- Tutorial
- Reference
 - Module functions
 - Module constants
 - Connection objects
 - Cursor objects
 - Row objects
 - Blob objects
 - PrepareProtocol objects
 - Exceptions
 - SQLite and Python types
 - Default adapters and converters (deprecated)
 - Command-line interface
- How-to guides
 - How to use placeholders to bind values in SQL queries
 - How to adapt custom Python types to SQLite values
 - How to write adaptable objects
 - How to register adapter callables

class sqlite3.Cursor

A [Cursor](#) instance has the following attributes and methods.

`execute(sql, parameters=(), /)`

Execute a single SQL statement, optionally binding Python values using [placeholders](#).

Parameters:

- **sql** ([str](#)) – A single SQL statement.
- **parameters** ([dict](#) | [sequence](#)) – Python values to bind to placeholders in *sql*. A dict if named placeholders are used. A sequence if unnamed placeholders are used. See [How to use placeholders to bind values in SQL queries](#).

Raises:

[ProgrammingError](#) – If *sql* contains more than one SQL statement.

If [autocommit](#) is [LEGACY_TRANSACTION_CONTROL](#), [isolation_level](#) is not [None](#), *sql* is an [INSERT](#), [UPDATE](#), [DELETE](#), or [REPLACE](#) statement, and there is no open transaction, a transaction is implicitly opened before executing *sql*.

Deprecated since version 3.12, will be removed in version 3.14: [DeprecationWarning](#) is emitted if [named placeholders](#) are used and *parameters* is a sequence instead of a [dict](#). Starting with Python 3.14, [ProgrammingError](#) will be raised instead.

Use [executescript\(\)](#) to execute multiple SQL statements.

`executemany(sql, parameters, /)`

For every item in *parameters*, repeatedly execute the [parameterized DML](#) SQL statement *sql*.

Uses the same implicit transaction handling as [execute\(\)](#).

Parameters:

- **sql** ([str](#)) – A single SQL DML statement.
- **parameters** ([iterable](#)) – An iterable of parameters to bind with the placeholders in *sql*. See [How to use placeholders to bind values in SQL queries](#)

Použitie Kurzora

1

2

3

4

5

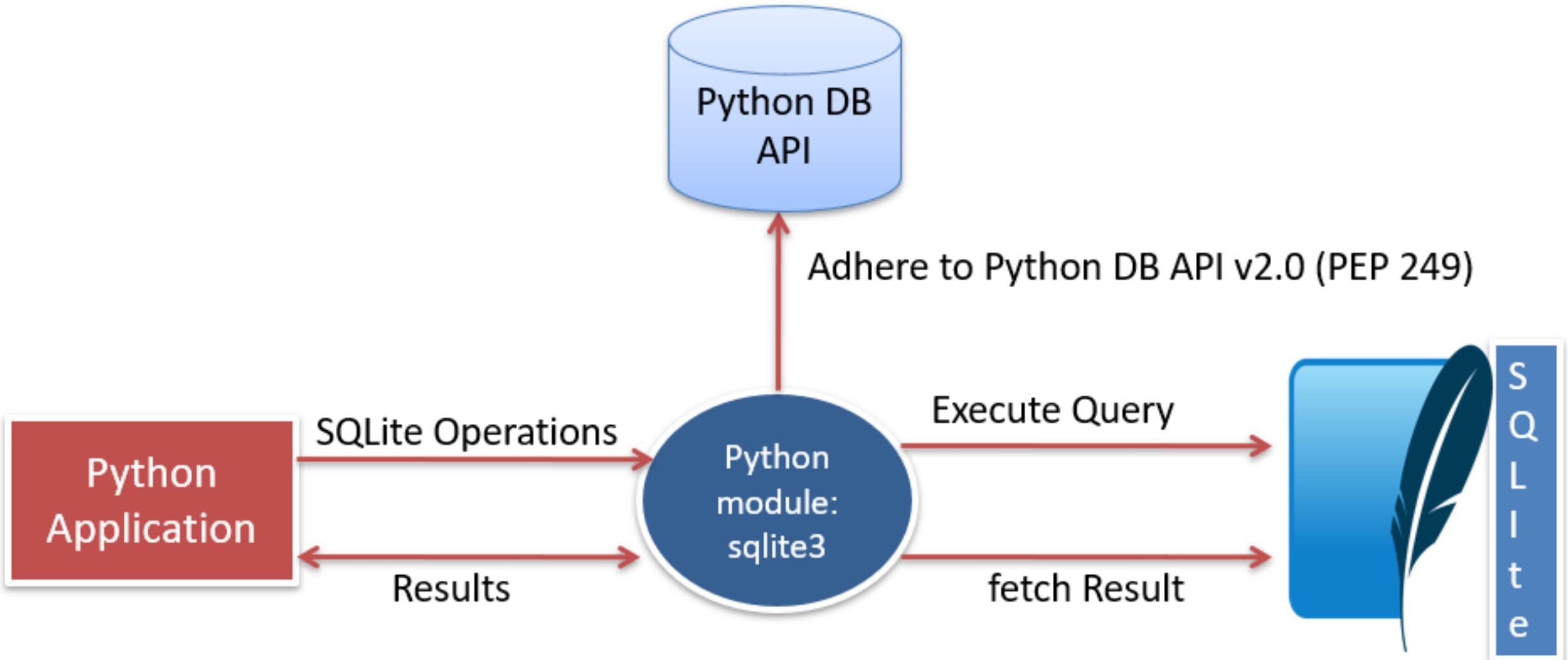
Navigácia cez výsledky:
Kurzor sa používa na postupné prechádzanie riadkami dát vrátenými SQL dopytov.

Manipulácia s dátami:
Kurzory umožňujú nielen čítanie dát, ale aj ich úpravy.

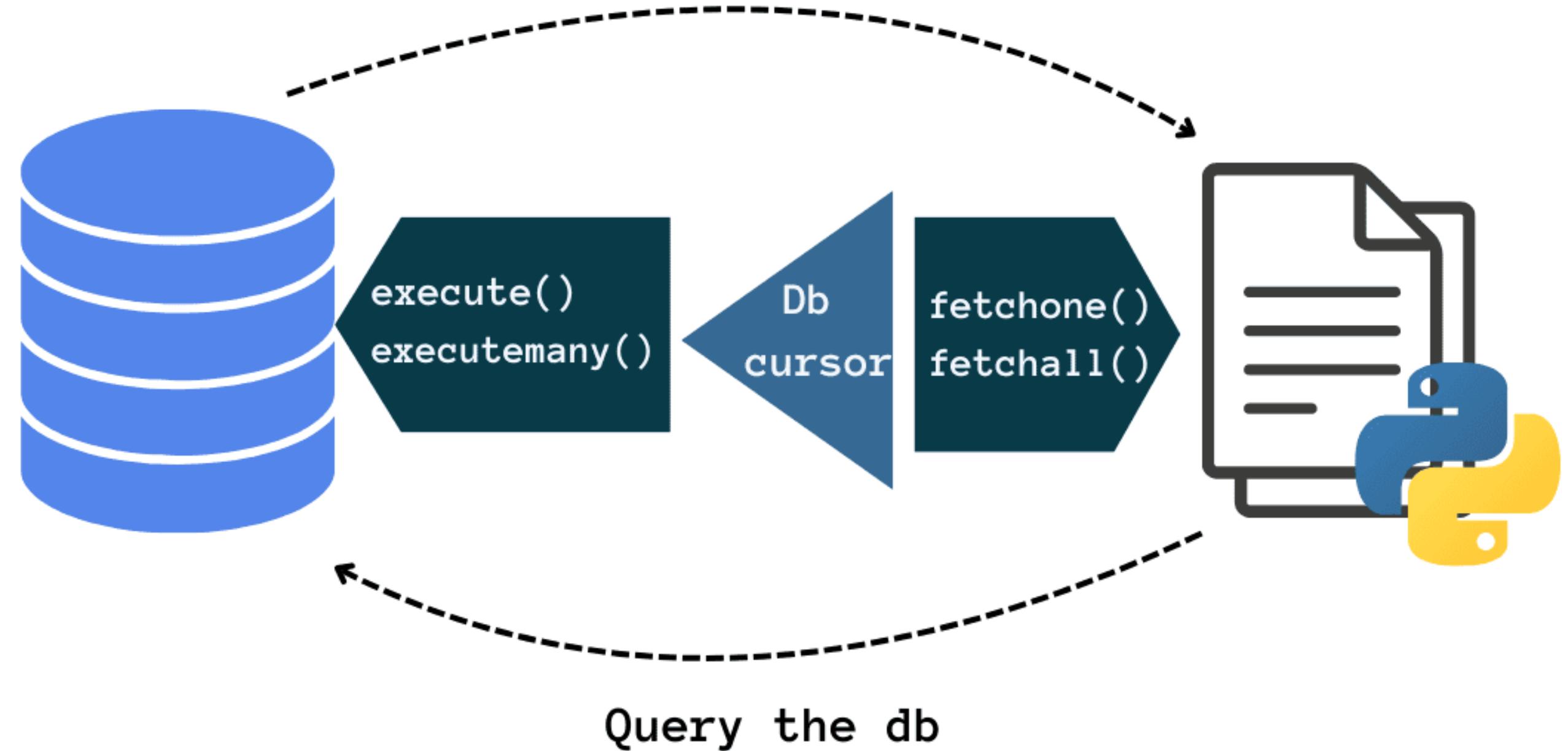
Transakčná kontrola:
Využitím kurzorov môže byť ľahšie riadiť transakcie.

Efektivita: V niektorých prípadoch je použitie efekt. z hľadiska pamäti a výkonu pri práci s obrovskými dátovými setmi.

SQL dopyt:
Kurzor sa často používa v kontexte vykonávania SQL príkazov.

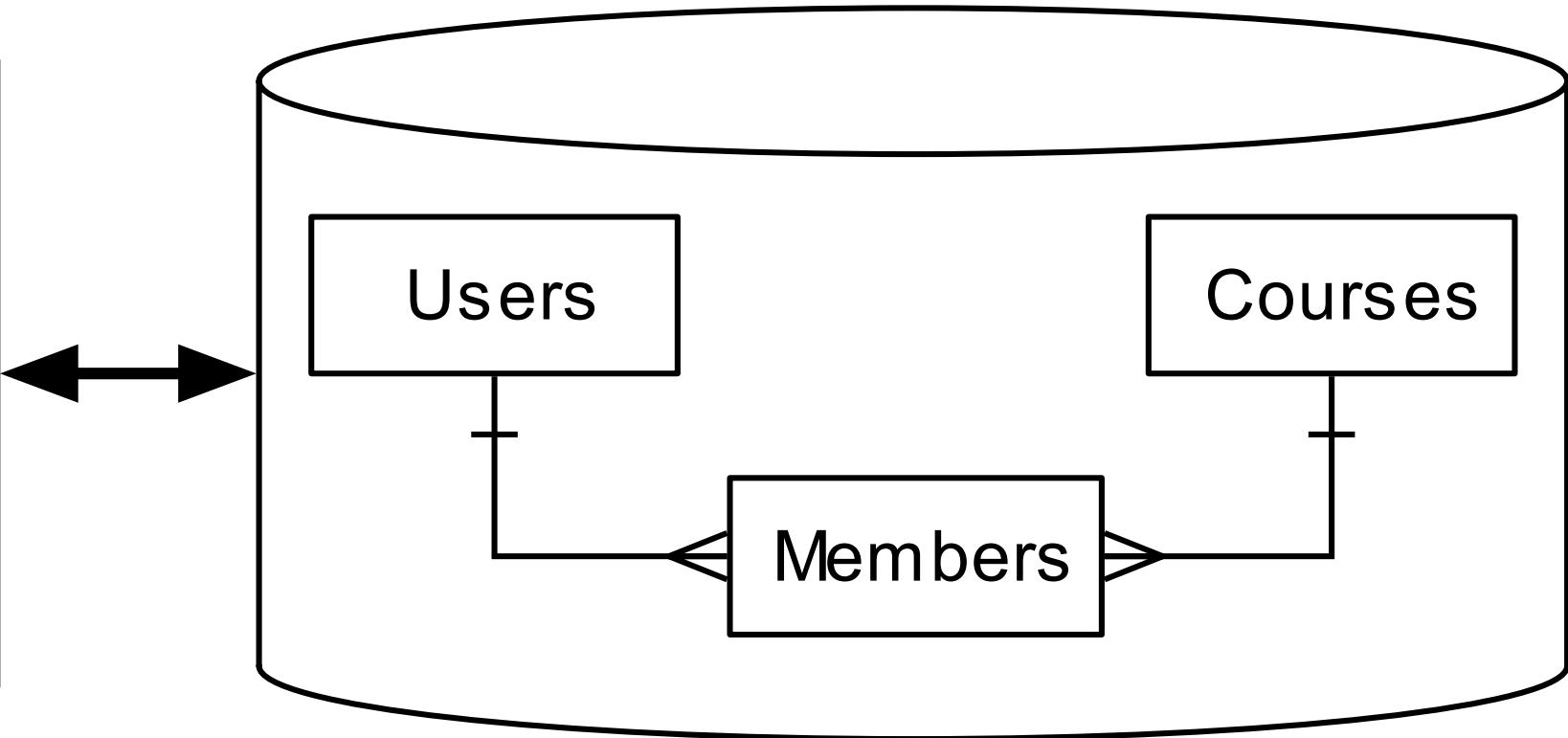


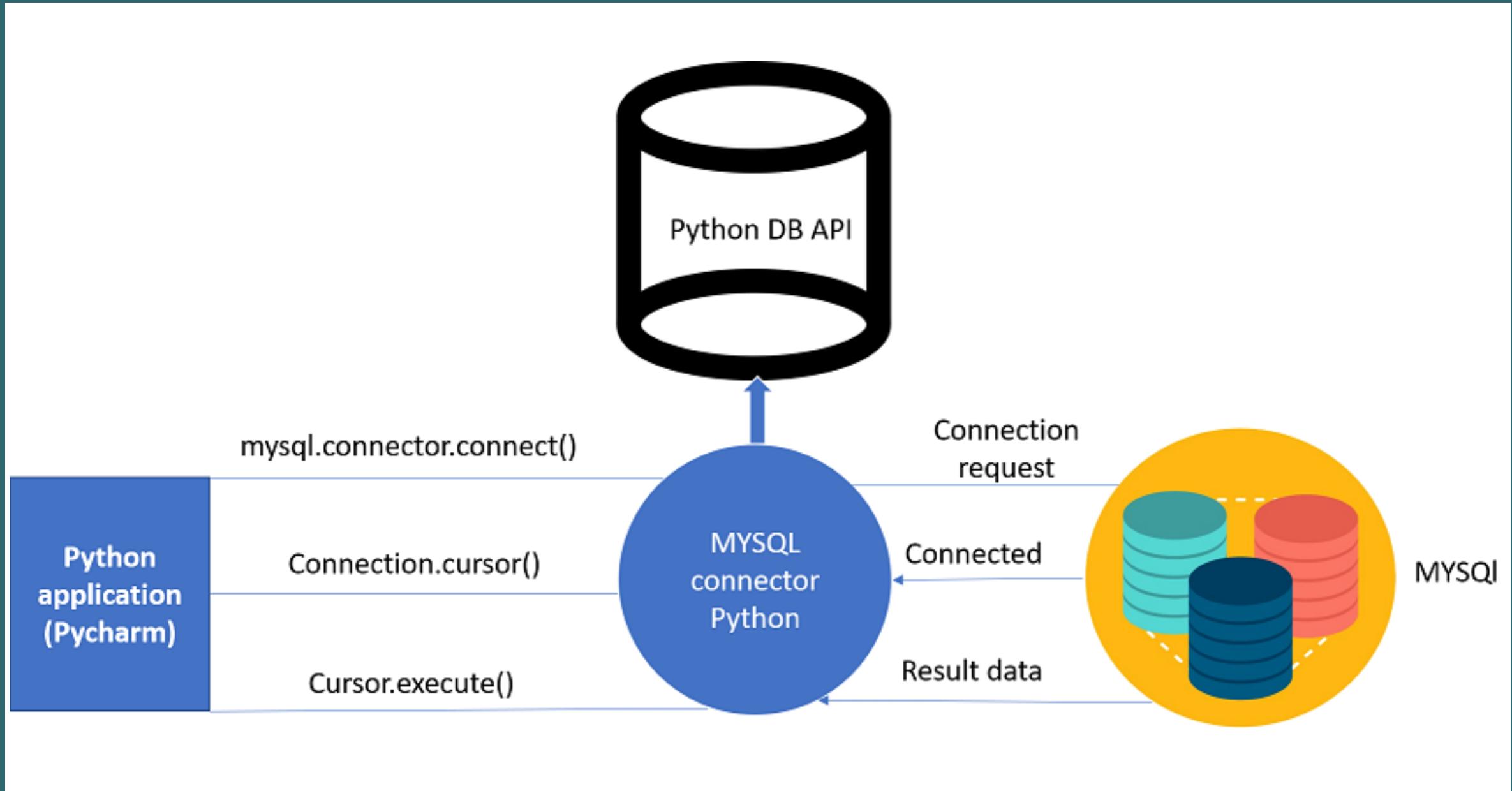
Fetch the results of the query

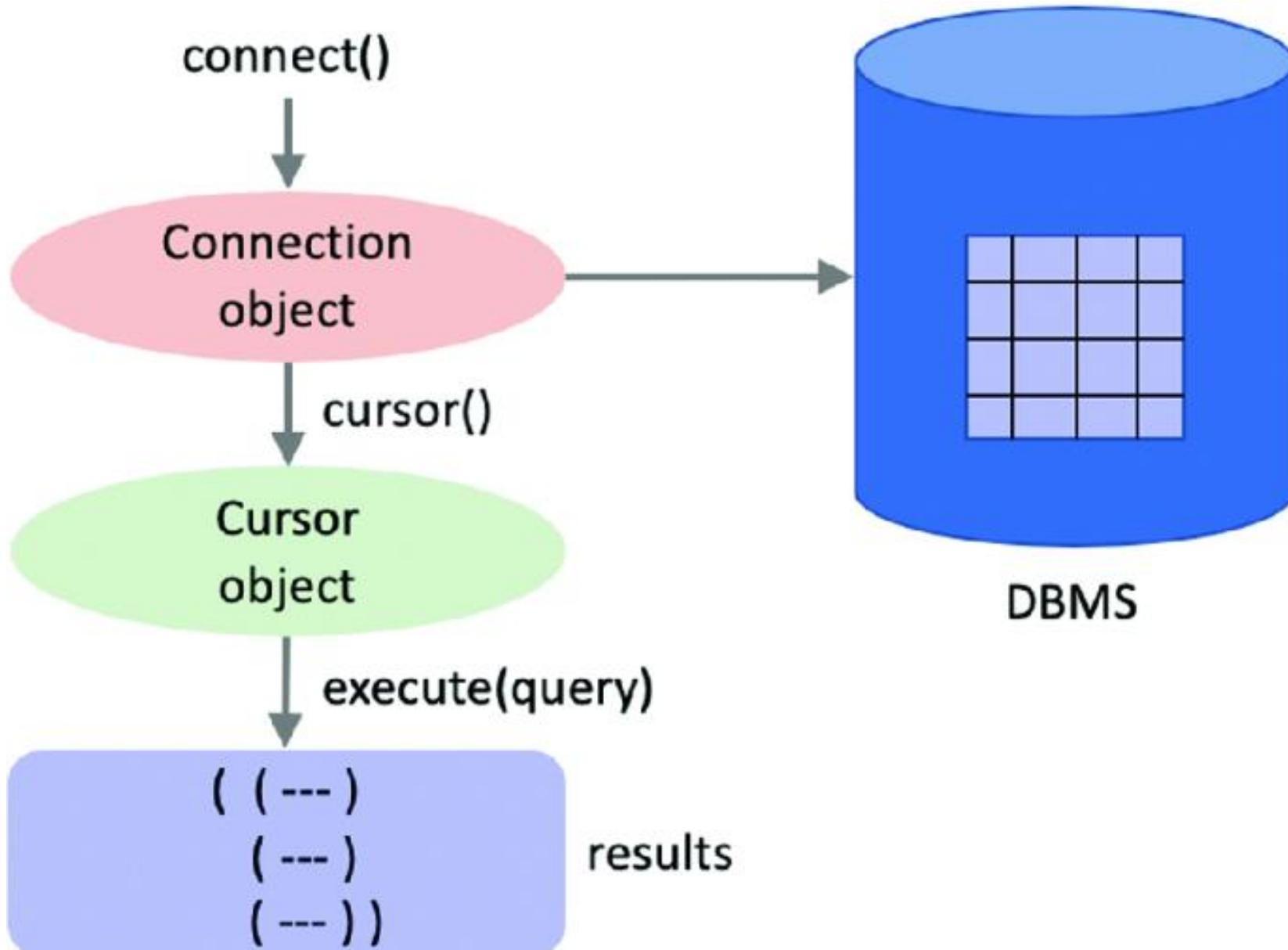


execute	CURSOR
fetchone	
fetchall	
close	

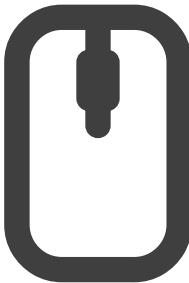
Your
Program







Použitie Kurzora



1. Vytvorenie kurzora:

python

```
connection = some_database_adapter.connect('database_name')
cursor = connection.cursor()
```

Copy code

2. Použitie kurzora na vykonávanie SQL príkazov:

python

```
cursor.execute("SELECT * FROM some_table")
```

Copy code

3. Získanie výsledkov zo SQL dotazu:

python

```
rows = cursor.fetchall()
for row in rows:
    print(row)
```

Copy code

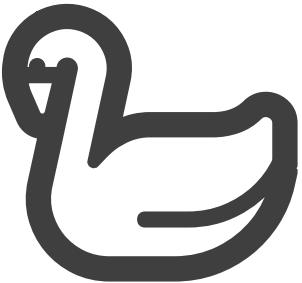
4. Zatvorenie kurzora a spojenia:

python

```
cursor.close()
connection.close()
```

Copy code

DB Kurzor a sqlite



python

```
import sqlite3

# Pripojenie k databáze SQLite
conn = sqlite3.connect('example.db')

# Vytvorenie kurzora
cursor = conn.cursor()

# Vytvorenie tabuľky
cursor.execute('''CREATE TABLE IF NOT EXISTS employees
                  (id INTEGER PRIMARY KEY, name TEXT, position TEXT)''')

# Vloženie údajov do tabuľky
cursor.execute("INSERT INTO employees (name, position) VALUES ('John Doe', 'Software Engineer')")
cursor.execute("INSERT INTO employees (name, position) VALUES ('Jane Smith', 'Project Manager')")

# Uloženie zmien
conn.commit()

# Načítanie údajov z tabuľky
cursor.execute("SELECT * FROM employees")

# Vypísanie načítaných údajov
for row in cursor.fetchall():
    print(row)

# Zatvorenie pripojenia
conn.close()
```

Copy code

DB Kurzor a MySQL



bash

```
pip install mysql-connector-python
```

Copy code

python

```
import mysql.connector

# Pripojenie k MySQL databáze
conn = mysql.connector.connect(
    host="localhost",
    user="yourusername",
    password="yourpassword",
    database="mydatabase"
)

# Vytvorenie kurzora
cursor = conn.cursor()

# Vytvorenie tabuľky
cursor.execute("CREATE TABLE IF NOT EXISTS employees (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), position VARCHAR(255))")

# Vloženie údajov do tabuľky
cursor.execute("INSERT INTO employees (name, position) VALUES ('Alice Blackwell', 'Software Engineer')")
cursor.execute("INSERT INTO employees (name, position) VALUES ('Bob White', 'Project Manager')")

# Uloženie zmien
conn.commit()

# Načítanie údajov z tabuľky
cursor.execute("SELECT * FROM employees")

# Vypísanie načítaných údajov
for row in cursor.fetchall():
    print(row)

# Zatvorenie pripojenia
conn.close()
```

Copy code

Objekt sqlite3.Cursor

1. **arraysize** určuje, koľko riadkov bude načítaných z databázy v jednej operácii pri použití metódy `fetchmany()`
2. **description** poskytuje informácie o stĺpcoch vrátených posledným vykonaným SQL dopytom
3. **lastrowid** poskytuje ID posledného riadku, ktorý bol vložený pomocou operácie `INSERT`
4. **row_factory** umožňuje nastaviť spôsob, akým sa riadky budú vracať z dopytu
5. **rowcount** indikuje počet riadkov ovplyvnených posledným vykonaným SQL príkazom (napr. `INSERT`, `UPDATE`, `DELETE`)

Úlohy Kurzory

1. Osvojenie si pojmu Kurzor
(DB Cursor)
2. Prehľad medzi typmi kurzorov
3. Používanie kurzorov
4. Pochopenie atribútorov kurzov



Úlohy Fetch



1. Osvojenie si pojmu Fetch (Fetching) v DBMS
2. Pochopenie fetch funkcií
3. Zorientovanie v základných typoch fetchov
4. DEMO: Fetch výsledkov z tabuľky DB



[Datalore documentation home page](#)[Quick start tutorial](#)[Overview of Datalore features](#)[▶ Before you start](#)[▶ Notebook editor](#)[▶ Data](#)[▼ Environment](#)[Customize notebook environment](#)

Environment variables

[Manage packages in the editor](#)[Work with Git in Datalore](#)[Switch Python versions](#)[▶ Kernels and computation](#)[▶ Sharing and collaboration](#)[▶ Account settings](#)[Plans](#)[▶ Datalore Enterprise installation guide](#)[▶ Administer Datalore Enterprise](#)[▶ Datalore reference](#)

Environment variables

Last modified: 07 April 2023

[Environment variables](#)[About environment variables](#)[Procedures](#)

About environment variables

Create sets of environment variables for your notebooks. Such variables can store passwords, API tokens, and other kinds of sensitive data for your notebooks to integrate with other services.

You can set up an environment variable for just one specific notebook or create one in a specific workspace and add it to any associated notebook.

Collaboration and permissions

This is how collaborators can manage environment variables:

- Environment variable owners (creators) can manage them on the workspace and notebook level
- Workspace collaborators with editor rights can attach already existing environment variables to, and detach them from, notebooks
- Workspace collaborators with viewer right can view environment variables

Migrate secrets to environment variables

Environment variables are the new alternative to Secrets (soon deprecated). To preserve

Fetching Získanie Dát

z DB 

 python



Úlohy Fetch



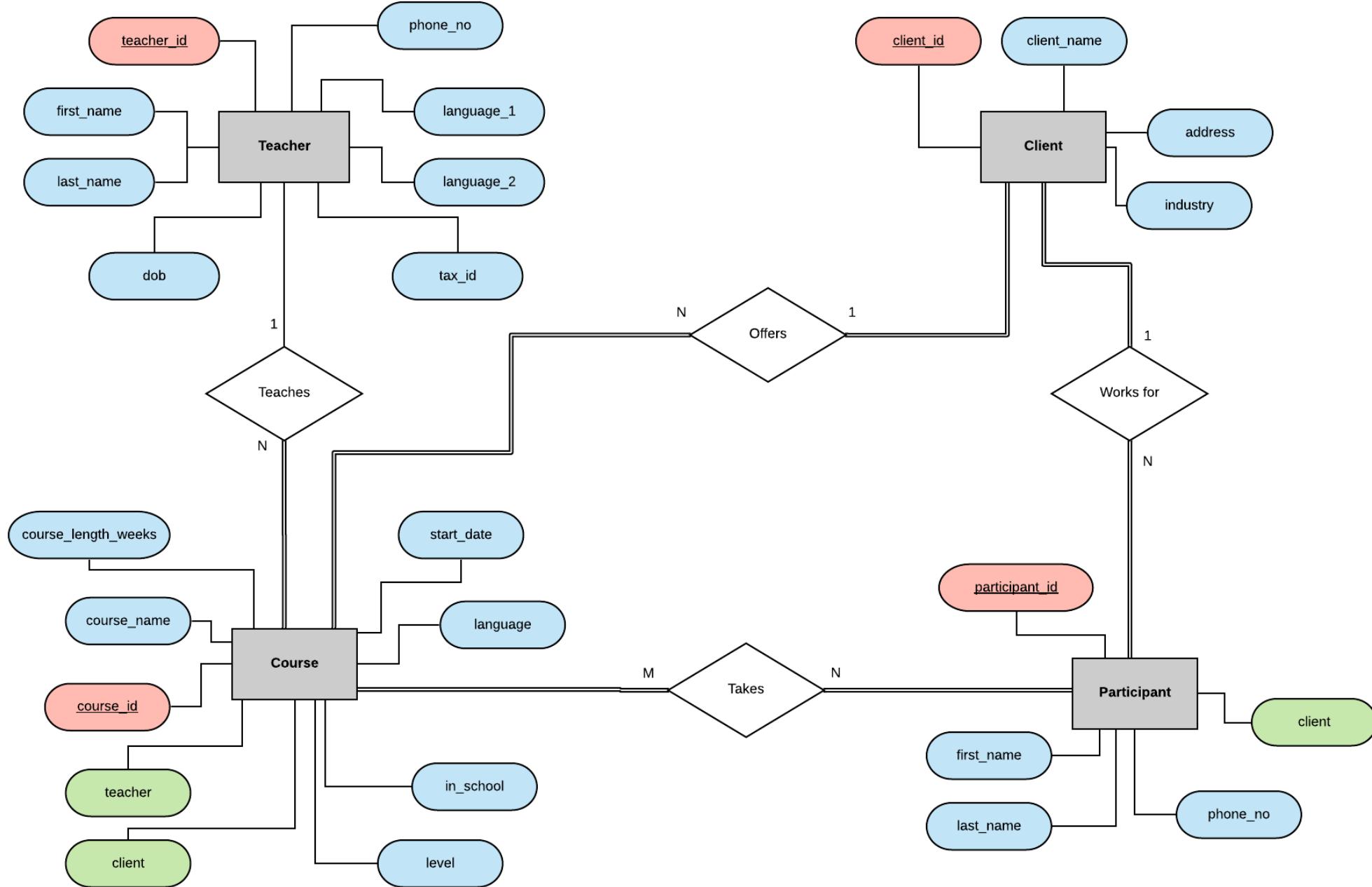
1. Osvojenie si pojmu Fetch (Fetching) v DBMS.
2. Pochopenie fetch funkcií.
3. Zorientovanie v základných typoch fetchov.
4. DEMO: Fetch výsledkov z tabuľky DB .



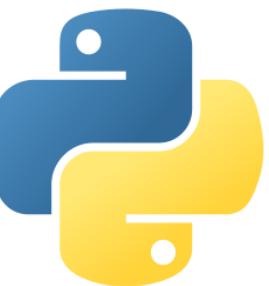
International Language School

Entity Relationship Diagram (ERD)

64

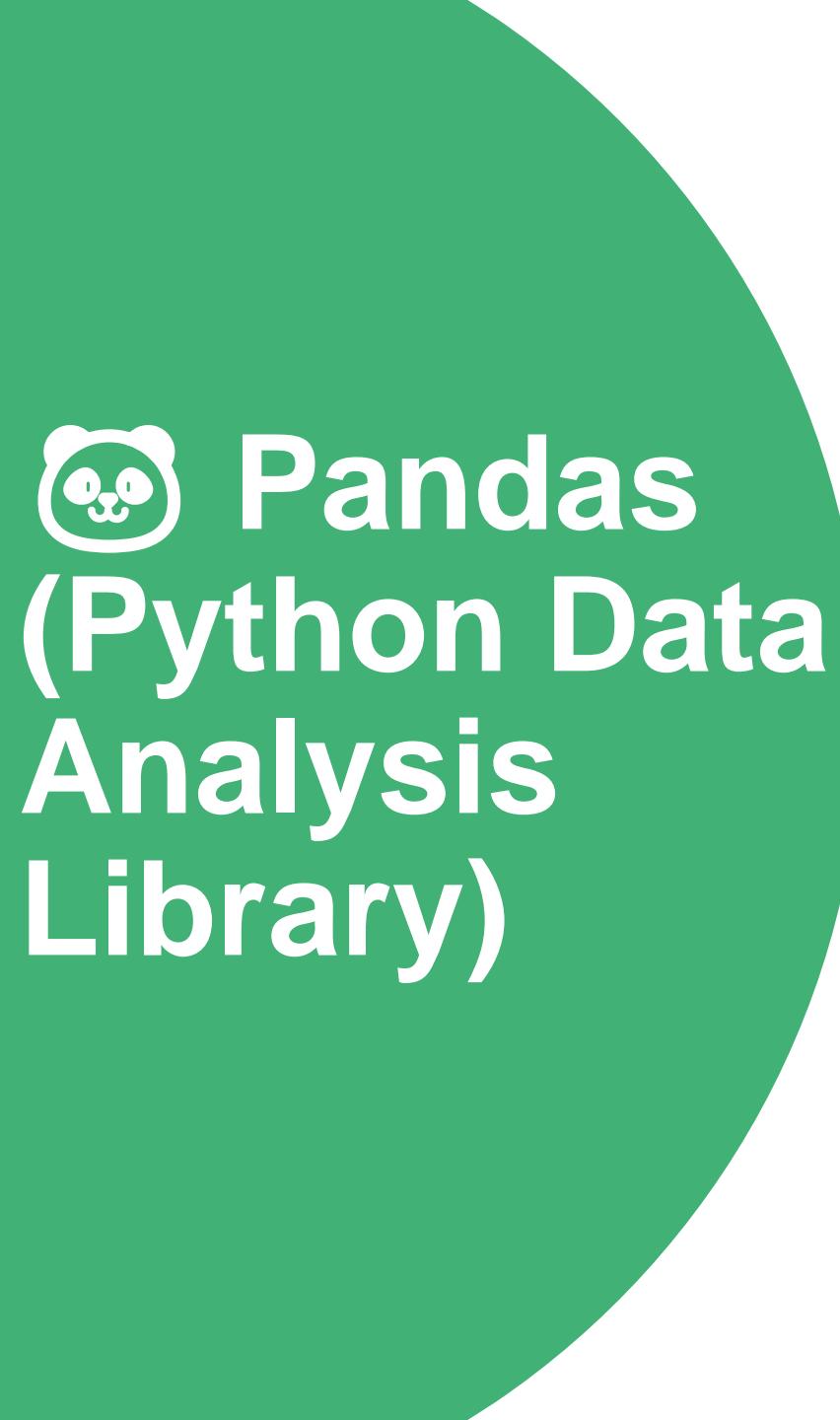


Pandas a DB



python





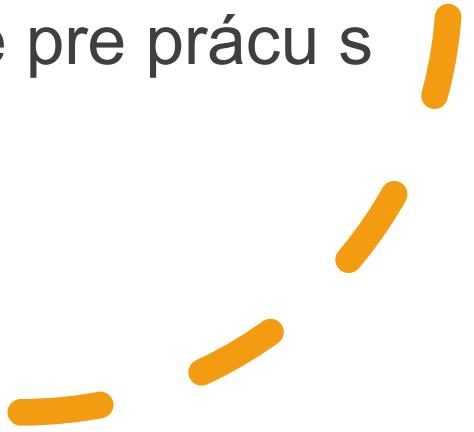
Pandas (Python Data Analysis Library)

- Knižnica programovacieho jazyka Python, ktorá sa používa pre prácu s dátami.
- Je špeciálne navrhnutá na manipuláciu a analýzu dátových štruktúr.
- Je veľmi populárna knižnica v oblasti analýzy dát a vede o dátach v Python, pretože veľmi zjednodušuje proces spracovania a analýzy dát.
- Hlavné dátové štruktúry v Pandas sú:
 1. **DataFrame**: Je to tabuľková dátová štruktúra, ktorá obsahuje riadky a stĺpce. Každý stĺpec môže mať rôzny dátový typ (napríklad čísla, reťazce, dátumy). DataFrame je vhodný na reprezentáciu skutočných dát ako sú tabuľky v Exceli alebo SQL tabuľky.
 2. **Series**: Je jednorozmerná dátová štruktúra v Pandas. Môžete si ju predstaviť ako jeden stĺpec dát z DataFrame.

Pandas Funkcie



- 1. **Čítanie dát**: Pandas podporuje čítanie rôznych formátov dát ako CSV, Excel, JSON a SQL databázy.
- 2. **Základné manipulácie**: Zmena hodnôt, pridávanie a mazanie stĺpcov a riadkov, filtrovanie dát a pod.
- 3. **Zlučovanie a spojovanie**: Pandas umožňuje zlučovať dáta z rôznych zdrojov.
- 4. **Skupinové operácie**: Ako súčet, priemer, maximálna hodnota pre skupiny dát.
- 5. **Časové rady**: Špeciálne funkcie pre prácu s časovými radmi.



Pandas

VS

SQL

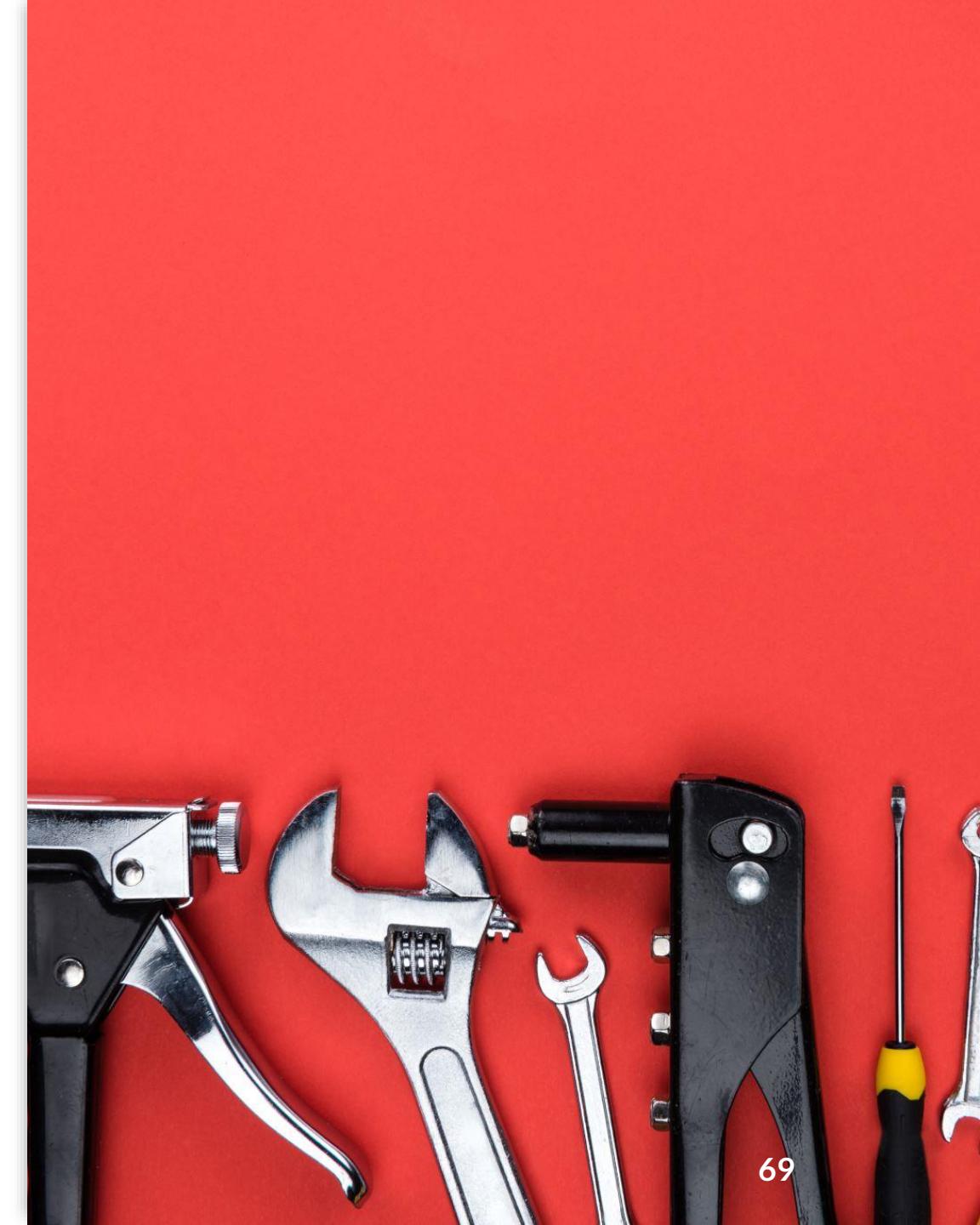


Operation	Pandas	SQL
Read CSV	<code>pd.read_csv(file)</code>	<code>LOAD DATA INFILE 'data.csv' INTO TABLE table FIELDS TERMINATED BY '' LINES TERMINATED BY '\n' IGNORE 1 ROWS;</code>
Print first 10 (or k) rows	<code>df.head(10)</code>	<code>SELECT * FROM table LIMIT 10;</code>
Dimensions	<code>df.shape</code>	<code>SELECT count(*) FROM table;</code>
Datatype	<code>df.dtypes</code>	<code>DESCRIBE table;</code>
Filter Data	<code>df[df.column>10]</code>	<code>SELECT * FROM table where column>10;</code>
Select column(s)	<code>df.column</code>	<code>SELECT column FROM table;</code>
Sort	<code>df.sort_values("column")</code>	<code>SELECT * FROM table ORDER BY column;</code>
Fill NaN	<code>df.column.fillna(0)</code>	<code>UPDATE table SET column=0 WHERE column IS NULL;</code>
Join	<code>pd.merge(df1, df2, on ="col", how = "inner")</code>	<code>SELECT * FROM table1 JOIN table2 ON (table1.col = table2.col);</code>
Concatenate	<code>pd.concat((df1, df2))</code>	<code>SELECT * FROM table1 UNION ALL table2;</code>
Group	<code>df.groupby("column"). agg_col.mean()</code>	<code>SELECT column, avg(agg_col) FROM table GROUP BY column;</code>
Unique values	<code>df.column.unique()</code>	<code>SELECT DISTINCT column FROM table;</code>
Rename column	<code>df.rename(columns = {"old_name": "new_name"})</code>	<code>ALTER TABLE table RENAME COLUMN old_name TO new_name;</code>
Delete column	<code>df.drop(columns = ["column"])</code>	<code>ALTER TABLE table DROP COLUMN column;</code>

Výber dát (Select) v Pandas



- Pri **výbere podmnožín údajov** sa používajú **hranaté zátvorky []**
- V týchto **zátvorkách môžete použiť 1 štítok stĺpca/riadku**, zoznam štítkov stĺpca/riadku, časť štítkov, podmienený výraz alebo dvojbodku
- Ak **používate názvy riadkov a stĺpcov**, vyberte **konkrétnе riadky a/alebo stĺpce pomocou loc**
- Pri použití **pozícíí v tabuľke** vyberte **konkrétnе riadky a/alebo stĺpce pomocou iloc**
- **Výberu** môžete **priradiť nové hodnoty** na základe **loc/iloc**



Účel a Použitie

Knižnica Pandas



- Knižnica v jazyku Python na manipuláciu s dátami a analýzu, hlavne používaná pre spracovanie a analýzu dát v pamäti

Jazyk SQL



- Jazyk na spravovanie a manipuláciu s dátami v databázových systémoch, zameraný na vytváranie, spravovanie a získavanie dát z relačných databáz.

Typy Dát a Spracovanie

71

Knižnica Pandas



- Efektívne pracuje s tabuľkovými dátami, hlavne v pamäti.
- Vhodné pre komplexné transformácie a analýzy, ako sú zlučovanie datasetov, pivotovanie, komplexné filtrovanie, agregácie a pod.

Jazyk SQL



- Navrhnuté na prácu s veľkými objemami dát uložených v databázach.
- Optimalizované pre efektívne vyhľadávanie, filtrovanie, a agregáciu dát priamo v databáze.

Výkon (Performance)

Knižnica Pandas

- Obmedzené kapacitou pamäte počítača.
- Výkon môže byť problém pri veľmi veľkých dátových súboroch.

Jazyk SQL

- Môže efektívne spracovať veľké množstvá dát, výkon závisí na výkone databázového servera.

Syntax a Učenie sa

Knižnica Pandas



- Vyžaduje znalosť Pythonu.
- Syntax môže byť pre nováčikov menej intuitívna, ale je veľmi silná pre dátovú analýzu.

Jazyk SQL



- Dobrá integrácia s rôznymi databázovými systémami a aplikáciami.
- Mnohé databázové systémy poskytujú vlastné rozšírenia a optimalizácie.

Integrácia a Rozšíriteľnosť

74

Knižnica Pandas



- Dobre integrovateľná s inými Python knižnicami a nástrojmi, ako sú NumPy, SciPy, Matplotlib atď.

Jazyk SQL



- Relatívne jednoduchý a deklaratívny jazyk.
- Široko používaný a štandardizovaný, so zameraním na správu dát.

Škálovateľnosť

Knižnica Pandas



- Menej skalovateľná, vhodná pre stredne veľké datasety.

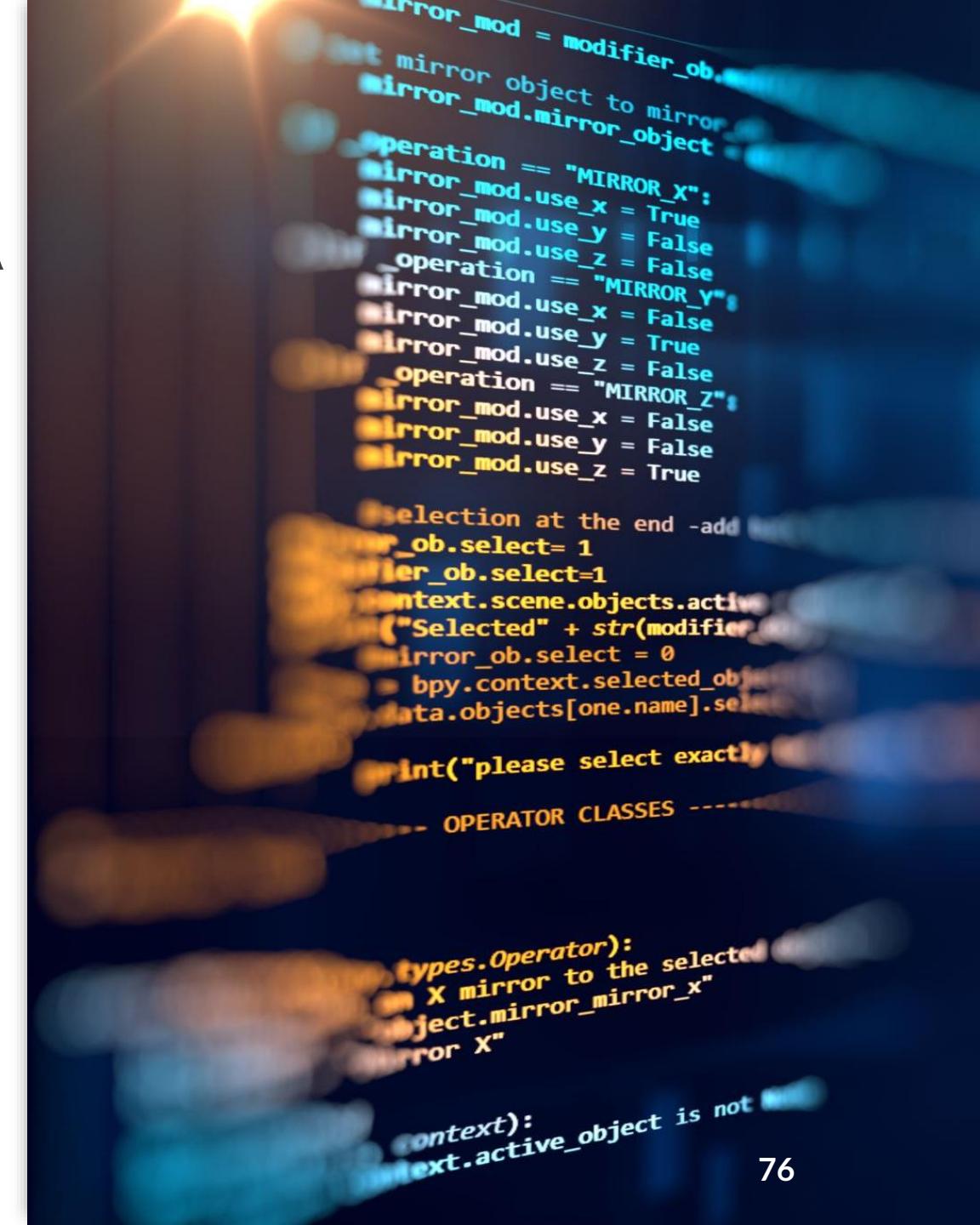
Jazyk SQL



- Vysoko skalovateľný, ideálny pre prácu s veľmi veľkými dátovými súbormi a databázami.

Sumarizácia

- **V závislosti od konkrétej úlohy** a dátových potrieb môže byť vhodnejšie použiť Pandas, SQL, **alebo dokonca kombináciu oboch.**
- Napríklad, môžete použiť SQL na vyfiltrovanie a získanie dát z databázy a následne použiť Pandas na ďalšiu analýzu a vizualizáciu týchto dát v Python prostredí.



Úlohy Pandas



1. Osvojenie si pojmu Dataframes a Series.
2. Použitie funkcie `read_sql_query`.
3. Pochopenie Pandas funkcií.
4. DEMO: Načítanie výsledkov z tabuľky EMP v DB.

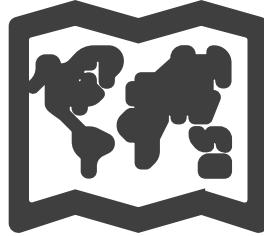


Objektovo- Relačné Mapovanie (ORM)

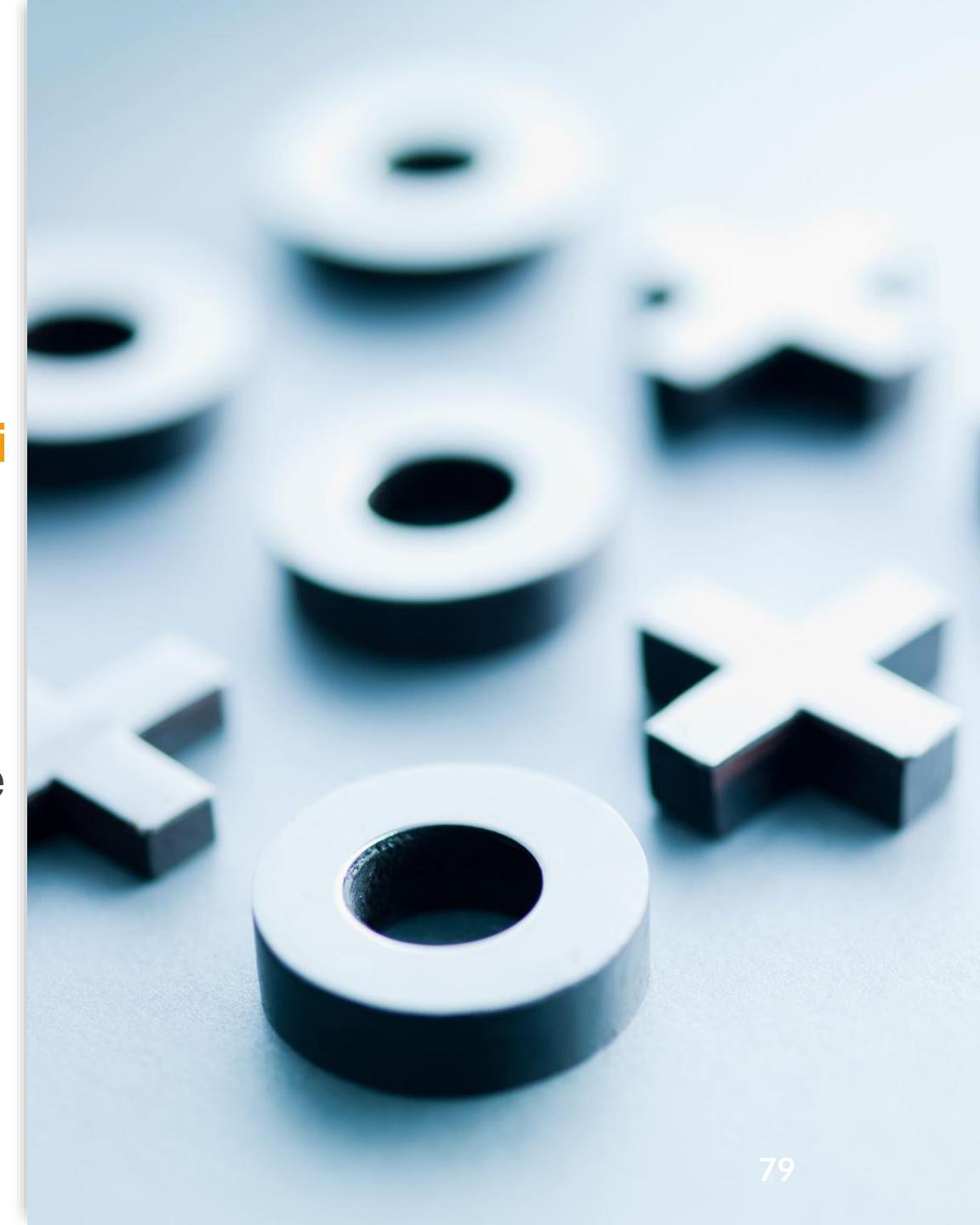
 python



ORM



- Technika programovania, ktorá umožňuje konverziu dát medzi nekompatibilnými typmi systémov používanými v objektovo orientovanom programovaní a relačných databázach
- V kontexte objektovo orientovaného programovania, ORM pomáha prekladat' dáta z objektov (ktoré sú typické pre programacie jazyky) na relačné tabuľky (typické pre databázové systémy), a naopak
- Toto mapovanie umožňuje programátorom pracovať s dátami na vyššej úrovni abstrakcie, bez toho, aby museli priamo interagovať s databázovými tabuľkami alebo SQL dopytmi



web framework	None	Flask	Flask	Django
ORM	SQLAlchemy	SQLAlchemy	SQLAlchemy	Django ORM
database connector	(built into Python stdlib)	MySQL-python	psycopg	psycopg
relational database	 SQLite	 MySQL	 PostgreSQL	 PostgreSQL



- Programovacia technika v softvérovom inžinierstve, ktorá zaistuje automatickú konverziu dát medzi **relačnou databázou** a **objektovo orientovaným programovacím jazykom**
- Pri **modelovaní** a **vývoji aplikácií** je snaha čo **najvernejšie zachytiť realitu**
- Objekty reálneho sveta sú v aplikácii reprezentované **ako entity**
- Zatiaľ čo je **v relačnej databáze entita reprezentovaná ako riadok**, resp. **množina riadkov** v **databázových tabuľkách**, ako aj v **objektovo orientovanom jazyku** je **entita** spravidla reprezentovaná ako **inštancia** nejakej **tryedy**

ORM

Objektovo Relačné Mapovanie (ORM)

- Stará sa o **automatickú konverziu rozdielnych dátových typov** medzi databázovým systémom a programovacím jazykom
- Pokročilé techniky ORM tiež riešia napríklad možnosť **využitia objektovej dedičnosti**, ktorú štandardné relačné databázy nepodporujú
- Hlavným cieľom ORM je **synchronizácia medzi používanými objektmi** v aplikácii a ich **reprezentáciou v databázovom systéme** tak, aby bola zaistená **perzistencia dát**
- Vývojár potrebuje persistentne uchovávať objekty, ale nepotrebuje sa starať, ako sa táto perzistencia vykoná



Going
outside
and getting
a gf

import life

def girlfriend:
personality.id = kind,
caring
gender.id = ("female",

Čo je ORM?

83

Relational database (such as PostgreSQL or MySQL)

ID	FIRST_NAME	LAST_NAME	PHONE
1	John	Connor	+16105551234
2	Matt	Makai	+12025555689
3	Sarah	Smith	+19735554512
...

Python objects

```
class Person:  
    first_name = "John"  
    last_name = "Connor"  
    phone_number = "+16105551234"
```

```
class Person:  
    first_name = "Matt"  
    last_name = "Makai"  
    phone_number = "+12025555689"
```

```
class Person:  
    first_name = "Sarah"  
    last_name = "Smith"  
    phone_number = "+19735554512"
```

ORMs provide a bridge between
**relational database tables, relationships
and fields and Python objects**

Objektovo Relačné Mapovanie (ORM)

84

The Django
ORM

SQLAlchemy

Peewee

Tortoise

SQLObject

Storm



ORM v Python

- SQLAlchemy:** Toto je jedna z najrobustnejších a najflexibilnejších ORM knižníc v Pythone. Umožňuje prácu s databázami na vysokej ako aj nízkej úrovni abstrakcie. SQLAlchemy podporuje mnoho rôznych databázových backendov a ponúka silný systém pre prácu s transakciami a databázovými modelmi.
- Django ORM:** Súčasťou populárneho Django web frameworku, Django ORM poskytuje vysoko abstraktný, ale mocný spôsob práce s databázami. Je navrhnutý tak, aby sa ľahko používal s Django modelmi a zabezpečil efektívnu prácu s databázami v kontexte webových aplikácií.
- Peewee:** Jedná sa o menší, jednoduchší ORM nástroj vhodný pre menšie aplikácie a projekty, kde nie je potrebná plná funkčnosť SQLAlchemy alebo Django ORM. Peewee je známy svojou jednoduchosťou a ľahkosťou použitia.
- SQLObject:** Táto knižnica ponúka objektovo-relačné mapovanie a je známa svojou jednoduchosťou a priamym prístupom k mapovaniu. Je vhodná pre malé a stredne veľké projekty.
- Tortoise ORM:** Ide o asynchrónny ORM, ktorý je navrhnutý špeciálne pre použitie s asynchronnými Python frameworkmi ako je FastAPI a Sanic. Ponúka jednoduché API a podporuje viaceré databázové backendy.

Vzor ORM Django

python

 Copy code

```
from django.db import models

class User(models.Model):
    username = models.CharField(max_length=100)
    email = models.EmailField()
    join_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.username
```

ORM Django Databázové Operácie

- Vytváranie nového záznamu:

python

 Copy code

```
new_user = User(username='novy_pouzivatel', email='email@priklad.com')
new_user.save()
```

- Vyhľadávanie záznamov:

python

 Copy code

```
users = User.objects.filter(username='novy_pouzivatel')
for user in users:
    print(user.email)
```

- Aktualizácia záznamu:

python

 Copy code

```
user = User.objects.get(username='novy_pouzivatel')
user.email = 'novy_email@priklad.com'
user.save()
```

- Mazanie záznamu:

python

 Copy code

```
user = User.objects.get(username='novy_pouzivatel')
user.delete()
```

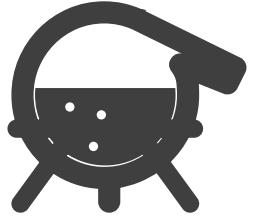
Úlohy ORM



1. Osvojenie si pojmu ORM.
2. Osvojenie si pojmu DBMS.
3. Pochopenie funkcií DBMS.
4. Zorientovanie v základných typoch databáz.
5. DEMO: Vytvorenie DB (+WS).



SQLAlchemy

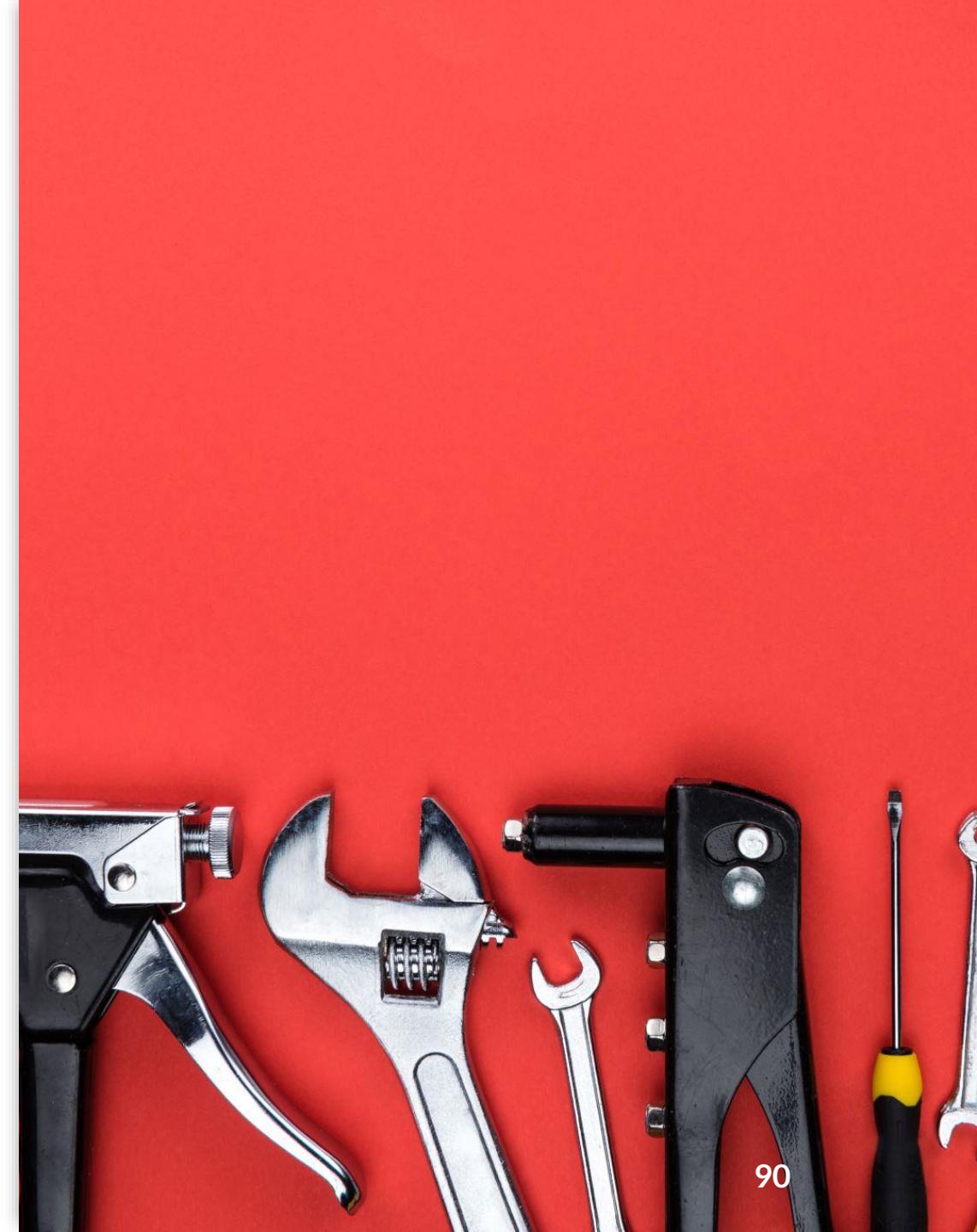


SQLA



SQLAlchemy

- **Knižnica** jazyka Python, ktorá sa používa pre **prácu s databázami**
- Je to **ORM** (Object-Relational Mapping) **nástroj**, ktorý umožňuje **programátorom pracovať s databázami pomocou objektovo orientovaného prístupu** namiesto tradičných SQL dopytov
- Poskytuje **abstraktnú vrstvu**, ktorá umožňuje programátorom **komunikovať s rôznymi typmi databáz**, ako napríklad MySQL, PostgreSQL, SQLite a ďalšie, **bez toho, aby museli meniť svoj kód**



Výhody SQLAlchemy

- 1. Abstrakcia databázy:** Umožňuje programátorom písat' databázový kód nezávisle od konkrétneho typu databázy.
- 2. Objektovo-relačná transformácia:** Umožňuje mapovať databázové tabuľky na Python triedy a objekty na riadky v tabuľkách, čo zjednodušuje manipuláciu s dátami.
- 3. Flexibilita a mocnosť:** Ponúka pokročilé funkcie ako transakčný manažment, lazy loading, automatické migrácie a iné.
- 4. Podpora pre komplexné SQL dotazy:** Aj keď je ORM mocný, SQLAlchemy stále umožňuje vykonávať surové SQL dotazy pre komplexné alebo špecifické operácie, ktoré nie sú priamo podporované ORM.



The Python SQL Toolkit and Object Relational Mapper

SQLAlchemy is the Python SQL toolkit and Object Relational Mapper that gives application developers the full power and flexibility of SQL.

It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

Documentation

- [Current Documentation \(version 2.0\)](#) - learn SQLAlchemy here
 - [Documentation Overview](#)
 - [Installation Guide](#)
 - [ORM Quickstart](#)
 - [Comprehensive Tutorial](#)
 - [Reference Guides](#)
 - [Object Relational Mapping \(ORM\)](#)
 - [Core \(Connections, Schema Management, SQL\)](#)
 - [Dialects \(specific backends\)](#)
- [Documentation by Version](#)
 - [Version 2.0](#)
 - [Version 1.4](#)
 - [Version 1.3](#)

Learn More

- [Front Matter](#)
 - [SQLAlchemy's Philosophy](#)

CURRENT RELEASES

2.0.10 - 2023-04-21 - [announce](#)
[changes](#) | [migration notes](#) | [docs](#)

[pypi v2.0.10](#) [downloads 69M/month](#)
[python 3.7 | 3.8 | 3.9 | 3.10 | 3.11](#)

1.4.47 - 2023-03-18 - [announce](#)
[changes](#) | [migration notes](#) | [docs](#)

SPONSOR SQLALCHEMY!

[Donate](#)

Donate to SQLAlchemy
through [PayPal](#)



Sponsor SQLAlchemy
through the Tidelift
Subscription

LATEST NEWS

SQLAlchemy 2.0.10 Released
Fri, 21 Apr 2023

SQLAlchemy 2.0.9 Released
Thu, 06 Apr 2023

SQLAlchemy 2.0.8 Released
Fri, 31 Mar 2023

SQLAlchemy



```
from sqlalchemy import *
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relation, sessionmaker

Base = declarative_base()

class Movie(Base):
    __tablename__ = "movies"

    id = Column(Integer, primary_key=True)
    title = Column(String(255), nullable=False)
    year = Column(Integer)
    directed_by = Column(Integer, ForeignKey("directors.id"))

    director = relation("Director", backref="movies", lazy=False)

    def __init__(self, title=None, year=None):
        self.title = title
        self.year = year

    def __repr__(self):
        return "Movie(%r, %r, %r)" % (self.title, self.year, self.director)

class Director(Base):
    __tablename__ = "directors"

    id = Column(Integer, primary_key=True)
    name = Column(String(50), nullable=False, unique=True)

    def __init__(self, name=None):
        self.name = name

    def __repr__(self):
        return "Director(%r)" % (self.name)

engine = create_engine("dbms://user:pwd@host/dbname")
Base.metadata.create_all(engine)
```

Storm

```
from storm.locals import *

class Foo:
    __storm_table__ = 'foos'
    id = Int(primary=True)

class Thing:
    __storm_table__ = 'things'
    id = Int(primary=True)
    name = Unicode()
    description = Unicode()
    foo_id = Int()
    foo = Reference(foo_id, Foo.id)

db = create_database('sqlite:')
store = Store(db)

foo = Foo()
store.add(foo)
thing = Thing()
thing.foo = foo
store.add(thing)
store.commit()
```

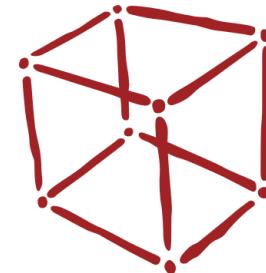
Úlohy SQLAlchemy



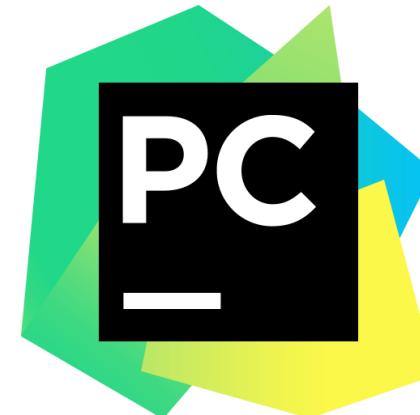
1. Osvojenie si pojmu Databáza (DB).
2. Osvojenie si pojmu DBMS.
3. Pochopenie funkcií DBMS.
4. Zorientovanie v základných typoch databáz.
5. DEMO: Vytvorenie DB (+WS).



Aké IDE mám použiť?



NetBeans



wxPython



Visual Studio



Chceme úplne všetko!



 ANACONDA NAVIGATOR[Sign in to Anaconda Cloud](#) Home

Applications on

base (root)

Channels

Refresh

 Environments Learning Community

Documentation

Developer Blog



Applications on base (root)		Channels
 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated	 JupyterLab 1.2.6 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.	 Notebook 6.0.3 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch	Launch	Launch
 VS Code 1.52.1 Streamlined code editor with support for development operations like debugging, task running and version control.	 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets.	 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.
Launch	Install	Install
 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated	 Qt Console 4.6.0 PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.	 Spyder 4.0.1 Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch	Launch	Launch

"python"

Filtre

Aplikácie

Hranie

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 €



Python

考试题库

计算计算机二级 Python 考试
库

Aplikácie • Education

Bezplatné

Python
DATA SCIENCE

Data Science with Python

Aplikácie • Education



Python

Python Programs

Aplikácie • Education

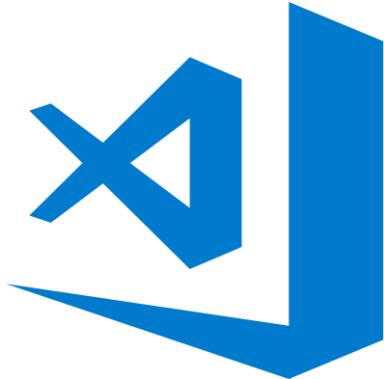
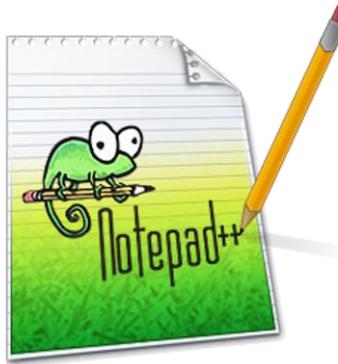
Learn
Python

Learn Python

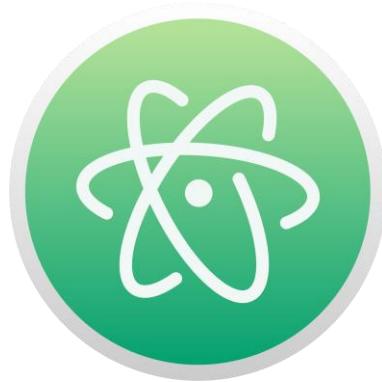
Aplikácie • Books & reference

Introduction to Python
Programming by...
Aplikácie • Books & referencePython Playground
Aplikácie • Developer toolsPython Programming
Manual For Absolute...
Aplikácie • Utilities & toolsPy
TutorialPython User Tutorial
Aplikácie • Utilities & tools

Aký editor mám použiť?



```
:::  
iLE880j. :jD888880j:  
.LGitE888D.f8GjjjL8888E;  
iE :888Et. .G888.  
;i E888, ,8888,  
D888, :8888:  
D888, :8888:  
D888, :8888:  
D888, :8888:  
888W, :8888:  
W88W, :8888:  
W88W: :8888:  
DGGD: :8888:  
:8888:  
:W888:  
:8888:  
E888i  
tW88D
```



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

[+ 5 -](#)*You have no recent consoles.*

New console:

[\\$ Bash](#)[>>> Python ▾](#)[More...](#)

Version

[2.7](#)[3.7](#)[3.8](#)

Recent Files

[+ 5 -](#)

/home/itacademysk/.bashrc
/home/itacademysk/.gitconfig
/home/itacademysk/.profile
/home/itacademysk/.pythonstartup.py
/home/itacademysk/.vimrc

[+ Open another file](#)[Browse files](#)

Recent Notebooks

[+ 5 -](#)

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:

A password input field with a small circular icon containing a question mark in the top right corner.

Confirm password:

A password input field with a small circular icon containing a question mark in the top right corner.

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.



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

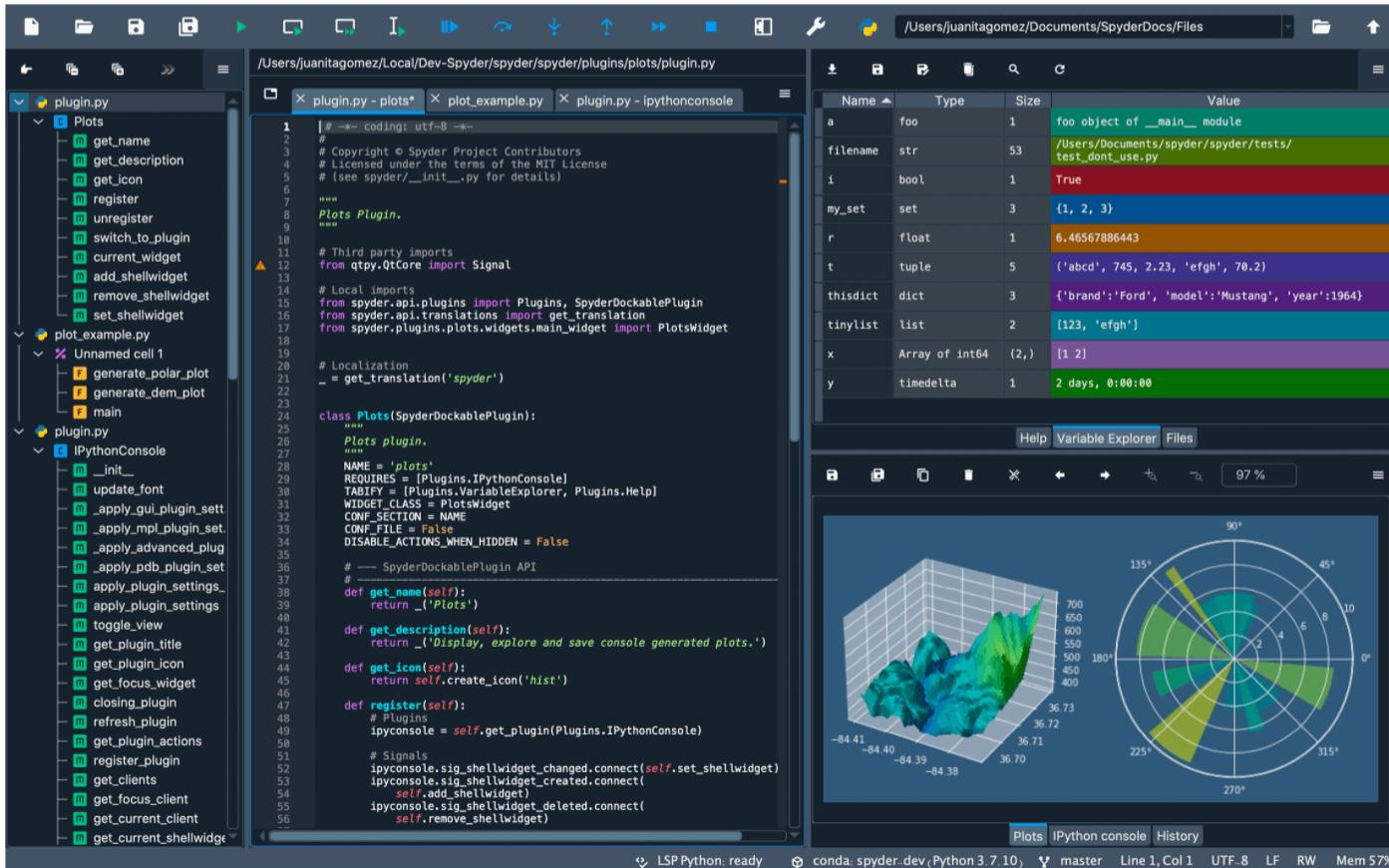
TODO

Plugins

> Version Control



The Scientific Python Development Environment



The screenshot displays the Spyder IDE interface with the following components:

- Code Editor:** Shows the file `/Users/juanitagomez/Local/Dev-Spyder/spyder/spyder/plugins/plots/plugin.py`. The code defines a `Plots` plugin for Spyder, which includes methods for generating polar plots and displaying them in the IPython console.
- Variable Explorer:** A table showing the current state of variables:

Name	Type	Size	Value
a	foo	1	foo object of __main__ module
filename	str	53	/Users/Documents/spyder/spyder/tests/test_dont_use.py
i	bool	1	True
my_set	set	3	{1, 2, 3}
r	float	1	6.46567886443
t	tuple	5	('abcd', 745, 2.23, 'efgh', 78.2)
thisdict	dict	3	{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
tinylist	list	2	[123, 'efgh']
x	Array of int64	(2,)	[1 2]
y	timedelta	1	2 days, 0:00:00
- Plots:** Two plots are displayed: a 3D surface plot of a mountain-like function and a polar plot showing concentric rings with radial labels.
- Bottom Status Bar:** Shows the Python version (LSP Python: ready), conda environment (conda: spyder.dev Python 3.7.10), and system status (master Line 1, Col 1 UTF-8 LF RW Mem 57%).

VERSION

Spyder 5

Search ...



WELCOME

QUICKSTART

INSTALLATION GUIDE

▶ INTRO VIDEOS

▶ PANES IN DEPTH

▶ SPYDER PLUGINS

▶ TROUBLESHOOTING

▶ WORKSHOPS

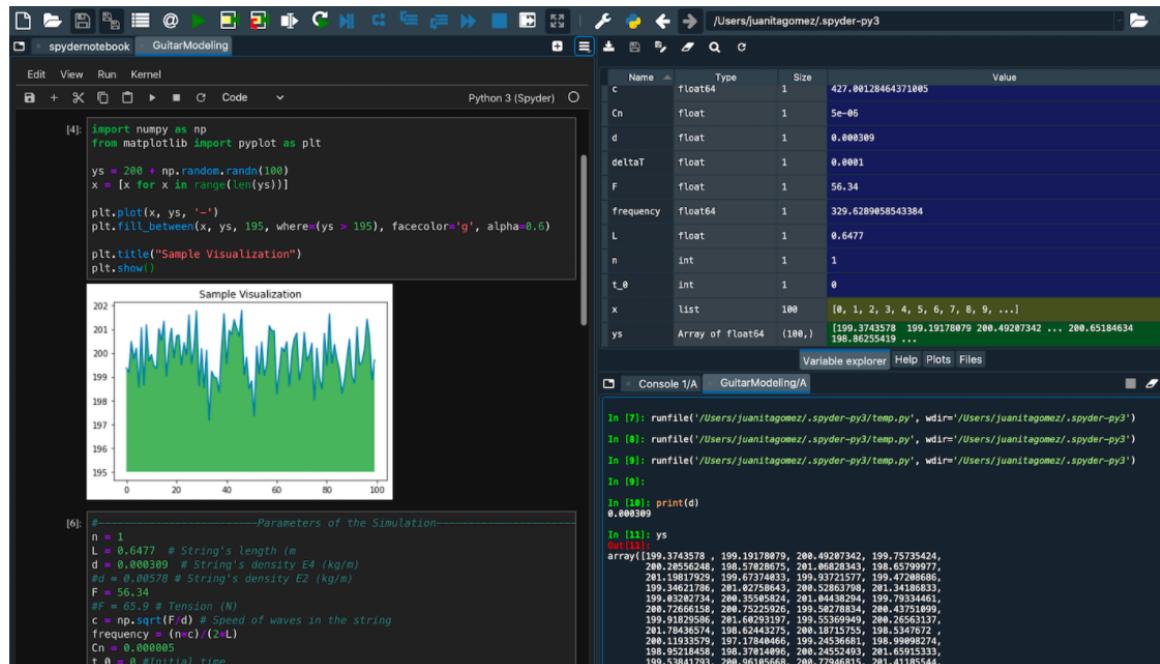
FAQ

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](#)[OPEN CHAT](#)

Útržky kódu

Filtrovať útržky kódu

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

Adding form fields Vložit'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 types of fields with type checking including sliders, date pickers, input fields, dropdown menus, and dropdown menus that allow input.

no_type_checking = '' #@param
string_type = 'example' #@param
slider_value = 142 #@param {type: "number", min: 0, max: 200}
number = 102 #@param {type: "number", min: 0, max: 200}
date = '2010-11-05' #@param {type: "date", min: '2010-01-01', max: '2020-12-31'}
pick_me = "monday" #@param [ 'monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday' ]
select_or_input = "apples" #@param {type: "select", options: [ "apples", "oranges", "bananas" ]}
```

[Zobrazit zdrojový zápisník](#)

+ Kód + Text Kopírovať na Disk

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

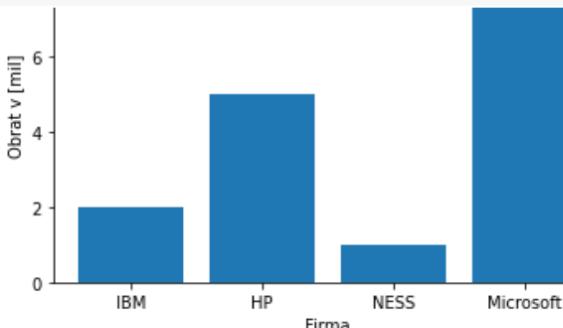
print(type(y))
print(y)

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]
```

C:\Users\miros

▼ ↑

Python_I_Za...

Edit View Run Kernel

Python 3 (Spyder)

Name Type Size Value

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
Uroková sadzba je: 1.5 % p.a.

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]:

Editor Notebook

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



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

temp.py kite_tutorial.py

```

1 # Welcome to...
2 #
3 #      `hmy+.      :::
4 #      .mMMMMMNho: ` NMMm
5 #      :NMMMMMMMMMdS/.` NMMm      :ss:
6 #      +NMMMMMMMMMMMMmy+ NMMm      -MMM-   ---
7 #      `oMMMMMMMMMMMMMMMo NMMm      /ss/   :MM+
8 #      `yMMMMMMMMNshmNMMMN` NMMm      /MM+
9 #      .dMMMMMMMMm/hmhssydmMM+ NMMm      `/yhy. shhy ohmMMMhhhh. ..ydmmdho-
10 #     omMMMMMd/mMMMMMhsosy` NMMm      .omMMmo. mMMN odmMMMdsss. omMNsoshNMNy
11 #     .+dMMMy/mMMMMMMMMMd- NMMm-yNMMy/` mMMN /MM+ sMMN: `:NMMy
12 #     `ymo/NMMMMMMMMMd NMMmNMNMMN` mMMN :MM+ MMNdddNNMMN
13 #     `hMMMMMMMMMM: NMMm+mMMNs. mMMN :MM+ MMN//////////////:
14 #     `:yNMMMMMMMMh NMMm `/dMMNy- mMMN :MM+ `sMMNo` `:
15 #     .+mMMMMMd- NMMm `/dMMNy- mMMN .MMNddNN/ +NMNdhydNNMs
16 #     `:yMMMy yhhs   `/hhh shhs :ydmmdho: `/sdmmmmhs:`
17 #     `om.
18
19 """
20 Kite is your Python programming copilot. Kite will try to show you the
21 right information at the right time as you code to prevent you from context
22 switching out of your current line of thought.
23
24 This tutorial will teach you how to use all of Kite's core features. You
25 should be able to learn everything in 5 minutes.
26
27 If you get stuck at any point, please visit https://help.kite.com/ or file
28 an issue at https://github.com/kiteco/issue-tracker.
29 """
30
31
32
33 """
34 PART 0: BEFORE WE START =====
35
36 Spyder will by default try to start the Kite backend when the editor first
37 starts. You can change this behavior by opening settings, clicking on
38 "Completion and linting", "Advanced", and then changing Kite's "Start Kite
39 Engine on editor startup" setting.
40
41 Look for the Kite indicator in the bottom left corner of Spyder's status
42 bar – It will tell you if Kite is ready and working. If the indicator reads
43 "not running", then you'll have to start the Kite Engine manually before
44 proceeding with the rest of this tutorial.
45 """
46
47
48
49

```

↓ ☰ 🔍 C

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]:



For Teams Download

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     if os.path.exists(os.path.join(dirname, '.git')):
6         count = 0
7         for root, dirs, files in os.walk(dirname):
8             count += len([f for f in files if f.endswith('.py')])
9         print('{} has {} Python files'.format(dirname, count))
10        for name in os.listdir(di)
```



dirname
dirs
dict

kite.com

```
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



Seems like I've
installed wrong version
of Python...

sys Variables		String Methods		Datetime Methods					
argv	Command line args	capitalize() *	lstrip()	today()	fromordinal(ordinal)				
builtin_module_names	Linked C modules	center(width)	partition(sep)	now(timezoneinfo)	combine(date, time)				
byteorder	Native byte order	count(sub, start, end)	replace(old, new)	utcnow()	strftime(date, format)				
check_interval	Signal check frequency	decode()	rfind(sub, start ,end)	fromtimestamp(timestamp)	utcfromtimestamp(timestamp)				
exec_prefix	Root directory	encode()	rindex(sub, start, end)						
executable	Name of executable	endswith(sub)	rjust(width)						
exitfunc	Exit function name	expandtabs()	rpartition(sep)						
modules	Loaded modules	find(sub, start, end)	rsplit(sep)						
path	Search path	index(sub, start, end)	rstrip()	replace()	utcoffset()				
platform	Current platform	isalnum() *	split(sep)	isoformat()	dst()				
stdin, stdout, stderr	File objects for I/O	isalpha() *	splines()	__str__()	tzname()				
version_info	Python version info	isdigit() *	startswith(sub)	strftime(format)					
winver	Version number	islower() *	strip()						
sys.argv for \$ python foo.py bar -c qux --h		isspace() *	swapcase() *	Date Formatting (strftime and strptime)					
sys.argv[0]	foo.py	istitle() *	title() *	%a	Abbreviated weekday (Sun)				
sys.argv[1]	bar	isupper() *	translate(table)	%A	Weekday (Sunday)				
sys.argv[2]	-c	join()	upper() *	%b	Abbreviated month name (Jan)				
sys.argv[3]	qux	ljust(width)	zfill(width)	%B	Month name (January)				
sys.argv[4]	--h	lower()		%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 number ¹ (00 to 53)				
				%W	Week number ² (0 to 6)				
				%x	Date				
				%X	Time				
				%y	Year without century (00 to 99)				
				%Y	Year (2008)				
				%Z	Time zone (GMT)				
				%%	A literal "%" character (%)				
os Variables		List Methods		File Methods					
altsep	Alternative sep	append(item)	pop(position)	close()	readlines(size)				
curdir	Current dir string	count(item)	remove(item)	flush()	seek(offset)				
defpath	Default search path	extend(list)	reverse()	fileno()	tell()				
devnull	Path of null device	index(item)	sort()	isatty()	truncate(size)				
extsep	Extension separator	insert(position, item)		next()	write(string)				
linesep	Line separator			read(size)	writelines(list)				
name	Name of OS			readline(size)					
pardir	Parent dir string								
pathsep	Patch separator								
sep	Path separator								
Note		Registered OS names: "posix", "nt", "mac", "os2", "ce", "java", "riscos"		Class Special Methods					
		<u>__new__(cls)</u>		<u>__lt__(self, other)</u>					
		<u>__init__(self, args)</u>		<u>__le__(self, other)</u>					
		<u>__del__(self)</u>		<u>__gt__(self, other)</u>					
		<u>__repr__(self)</u>		<u>__ge__(self, other)</u>					
		<u>__str__(self)</u>		<u>__eq__(self, other)</u>					
		<u>__cmp__(self, other)</u>		<u>__ne__(self, other)</u>					
		<u>__index__(self)</u>		<u>__nonzero__(self)</u>					
		<u>__hash__(self)</u>							
		<u>__getattr__(self, name)</u>							
		<u>__getattribute__(self, name)</u>							
		<u>__setattr__(self, name, attr)</u>							
		<u>__delattr__(self, name)</u>							
		<u>__call__(self, args, kwargs)</u>							
				Indexes and Slices (of a=[0,1,2,3,4,5])					
		<u>len(a)</u>		6					
		<u>a[0]</u>		0					
		<u>a[5]</u>		5					
		<u>a[-1]</u>		5					
		<u>a[-2]</u>		4					
		<u>a[1:]</u>		[1,2,3,4,5]					
		<u>a[:5]</u>		[0,1,2,3,4]					
		<u>a[:-2]</u>		[0,1,2,3]					
		<u>a[1:3]</u>		[1,2]					
		<u>a[1:-1]</u>		[1,2,3,4]					
		<u>b=a[:]</u>		Shallow copy of a					
Note									
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.									

PC Settings X

Search

> 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

	Cross Platform	Commercial/Fre e	Auto Code Completion	Multi-Langua ge Support	Integrat ed Python Debugging	Error Markup	Source Control Integrati on	Smart Indent	Bracket Matchin g	Line Numbering	UML Editing / Viewing	Code Folding	Code Templat es	Unit Testing	GUI Designe r (Qt, Eric, etc)	Integrat ed DB Support	Rapid Application	Development
Atom	Y	F			Y	Y	Y	Y	Y	Y		Y	Y					
BlackAdder	Y	C							Y			Y						
BlueFish	L																	
ConTEXT	W	C																
DABO	Y																	
DreamPie		F	Y				Y											
Dr.Python		F			Y													
Editra	Y	F	Y	Y			Y	Y	Y	Y		Y						
Emacs	Y	F	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y				
Eric Ide	Y	F	Y		Y	Y		Y		Y		Y			Y			
E-Texteditor	W																	
Geany	Y	F	Y*	Y				Y	Y	Y		Y						*very limited
Gedit	Y	F	Y ¹	Y				Y	Y	Y			Y ²					1with plugin; 2sort of
Idle	Y	F	Y		Y			Y	Y									
JEdit	Y	F		Y					Y	Y		Y						
KDevelop	Y	F		Y			Y	Y	Y	Y		Y						
Komodo	Y	CF	Y	Y	Y	Y	Y	Y	Y	Y		Y	Y	Y	Y	Y		
NetBeans*	Y	F	Y	Y	Y		Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	*pre-v7.0
NotePad++	W	F		Y				Y*		Y								*with plugin
Pfaide	W	C	Y	Y				Y	Y	Y		Y	Y					
PIDA	LW	F	Y	Y				Y	Y	Y		Y						VIM based
PTVS	W	F	Y	Y	Y	Y	Y	Y	Y	Y		Y			Y*		Y	*WPF bsed
PyCharm	Y	CF	Y	Y*	Y		Y	Y	Y	Y		Y		Y				*JavaScript
PyDev(Eclipse)	Y	F	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y		
Pyscripter	W	F	Y		Y	Y		Y		Y			Y	Y				
PythonWin	W	F	Y		Y			Y	Y			Y						
SciTE	Y	F ¹		Y		Y		Y	Y	Y		Y	Y					1Mac version is commercial
ScriptDev	W	C	Y	Y	Y	Y	Y		Y	Y	Y		Y	Y				
SPE		F	Y								Y							
Spyder	Y	F	Y		Y	Y		Y	Y	Y								
Sublime Text	Y	CF	Y	Y				Y	Y	Y		Y	Y	Y*				extensible w/Python,

Ktoré GUI a prečo?



wxPython



Dôležité Zdroje

Slovensko a Česko

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

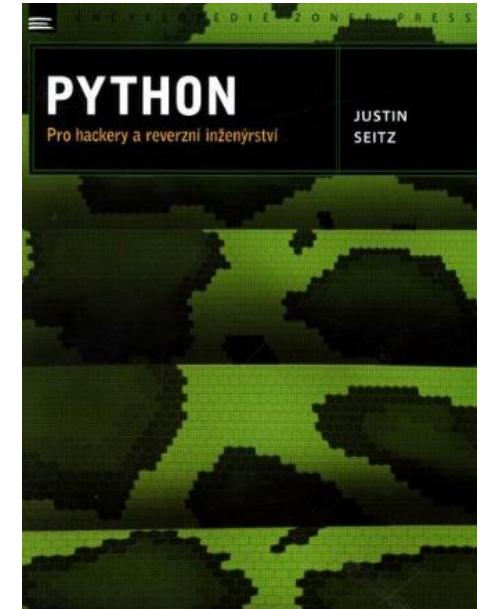
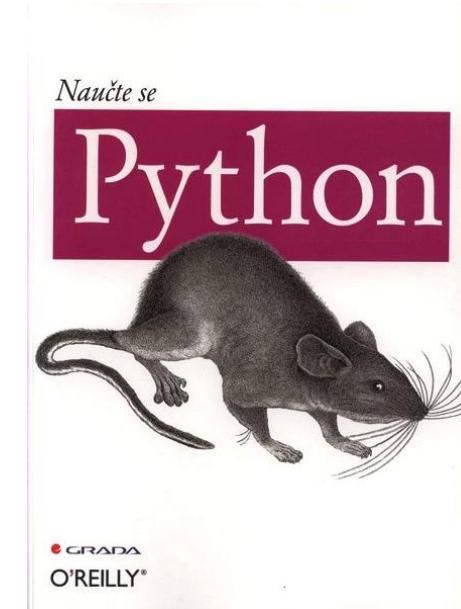
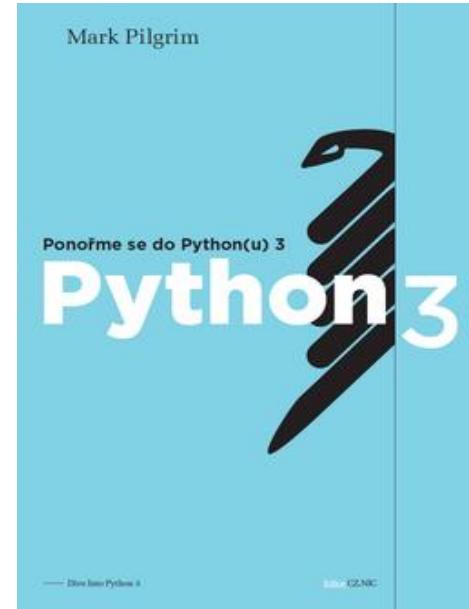
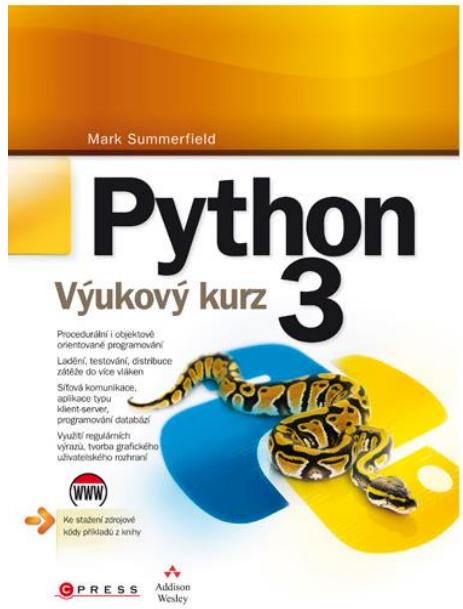
Zahraničie

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

YouTube Tutoriály

- [IT Academy](#)

IT Academy



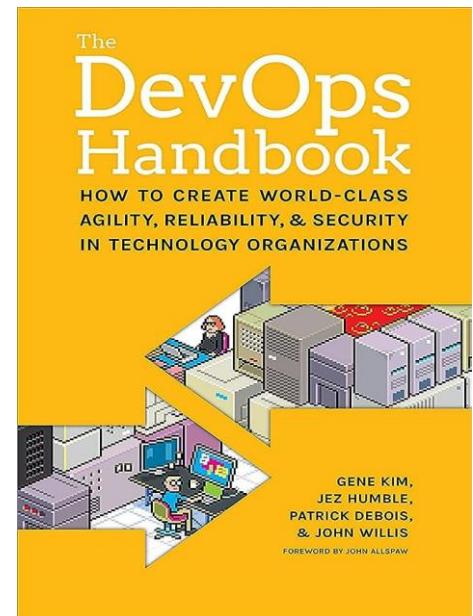
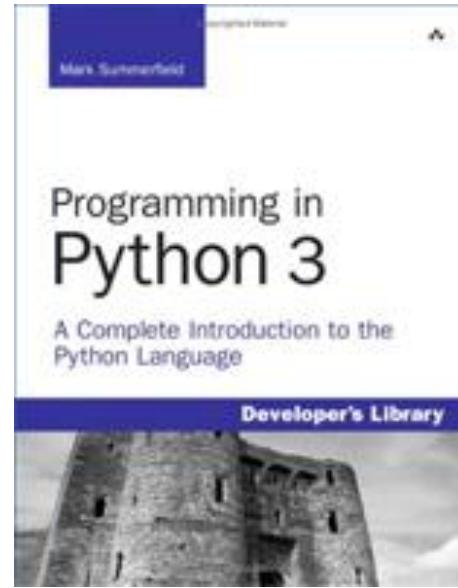
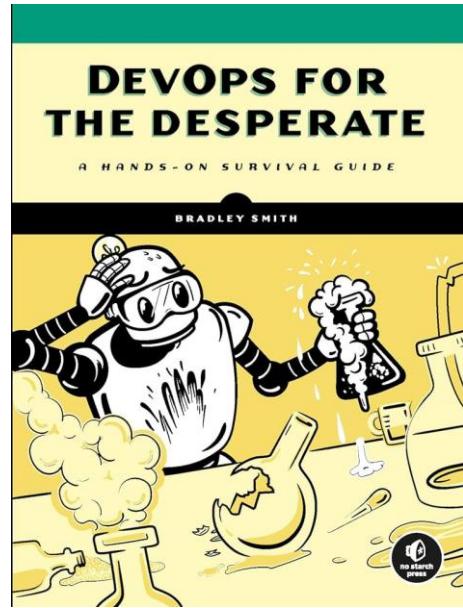
Čo sa oplatí/neoplatí prečítať SK/CZ?



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

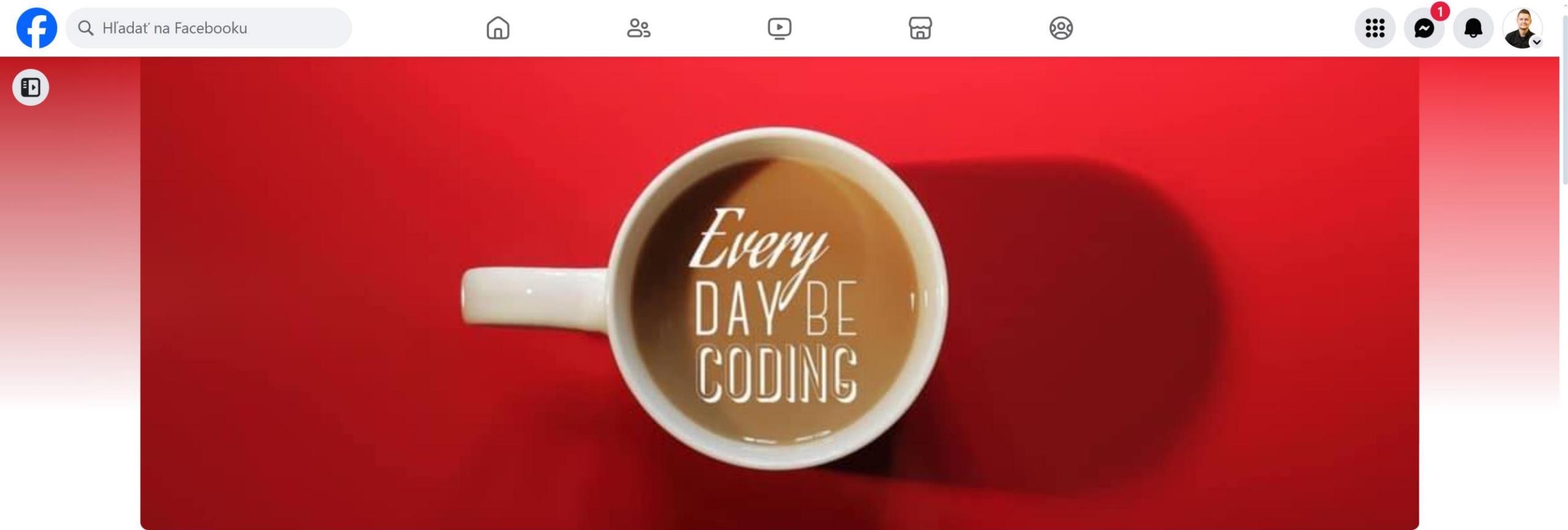
The diagram shows a standard QWERTY keyboard with various keys highlighted in different colors to represent different categories of special characters:

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

Surrounding the keyboard are large, semi-transparent symbols representing common uses of these characters:

- #
- &
- !
- €

IT ACADEMY



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

Popular**Name****New****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

[PROGRAM](#)[SPONZORI](#)

2%

[O NÁS](#)[KÚP SI LÍSTOK](#)

PYCON SK 2024

15.3.2024 - 17.3.2024 | Bratislava

23 : 12 : 54 : 57

days

hours

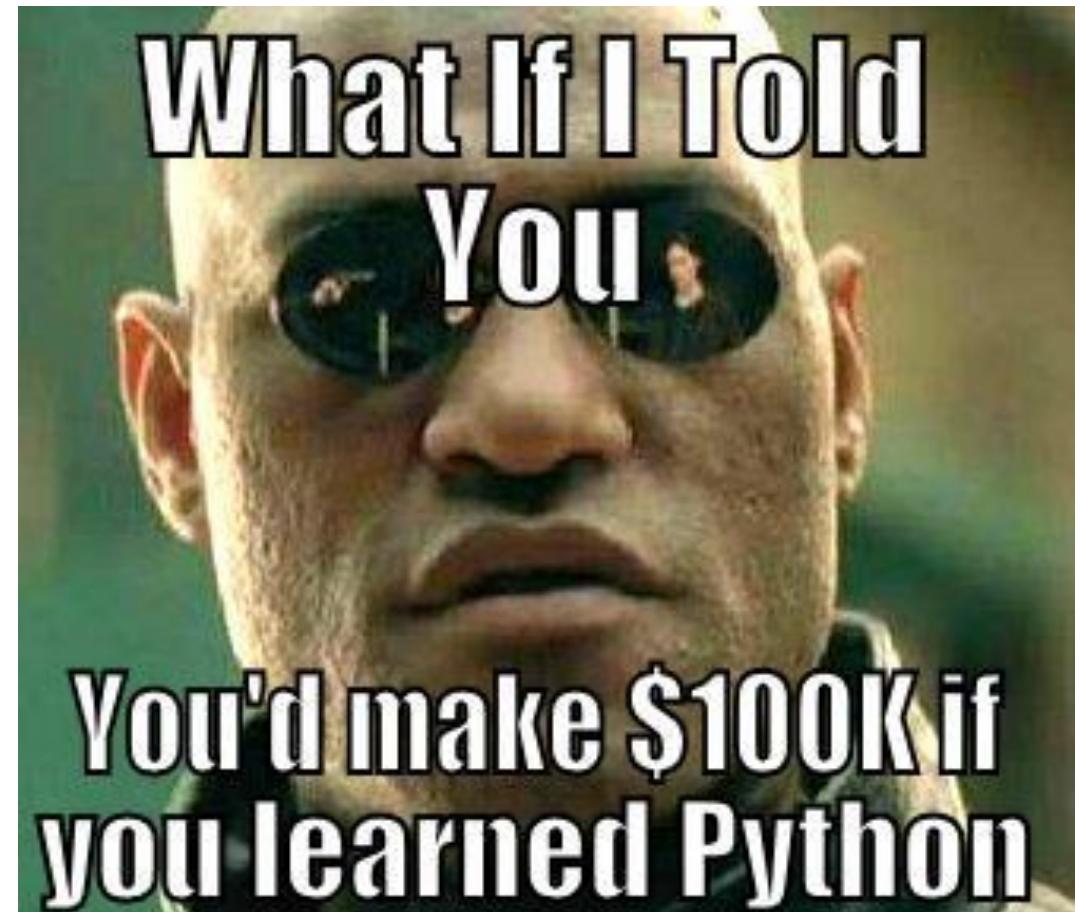
minutes

seconds

[KÚP SI LÍSTOK](#)

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)  
>>> |
```

No future {} a ;



Inšpirácia projekty

Python Project Ideas

Easy



Quote Gener.



Number guessing



Dice Simulation



YT downloader

Mid



Password Manag.



Mario Party

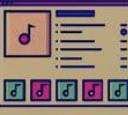


Web Crawler



Email Autom.

Pro



Music Player



Face Detection



Twitter Clone



Twitter Bot

@TheInsaneApp

Ako Skončíme?

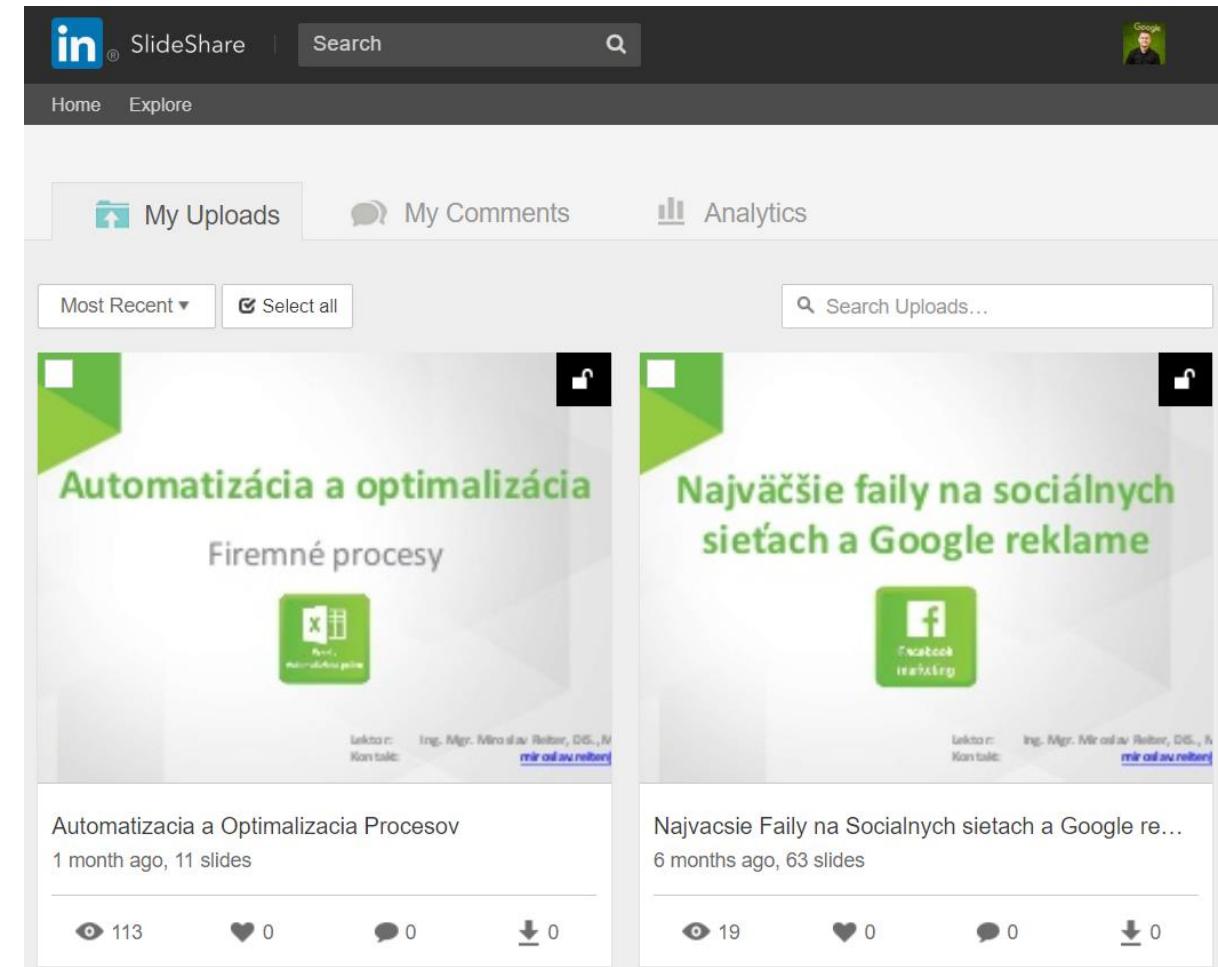
1. Stiahnite si Súbory

- 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€

Priístup pre Teba do všetkých aktuálnych aj pripravovaných online kurzov

12 mesačná platnosť

🛒 Kúpiť teraz

Zadarmo

1. Kurzy SAV/NSC

2. YouTube kanál IT Academy

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

Platené

Moje kurzy na www.vita.sk

Online Kurzy Zdarma



Úrady práce,
sociálnych
vecí a rodiny



info@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

Naučné kurzy

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 scievaní žiadostí o poskytnutie príspevku sa žiadateľ a jeho rodina prihládajú na individuálne predpoklady a poskytnutie príspevku nie je právny nárok, môže byť odmietnuté. Úrady práce, sociálnych vecí a rodiny žiadost schvália, keďž žiadateľ a jeho rodina zaplatí vzdelávací kurz. V rámci tohto projektu bude možné žiadať o poskytnutie príspevku iba jedným žiadostí.

info@vita.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 páči sa mi, komentovať alebo sa prihlásiť na odber, musíte sa prihlásiť.

Prihlásiť sa

Preskúmať

- Trendy
- Hudba
- Filmy
- Hry
- Šport

Prehliadat kanály

Viac zo služby YouTube

- YouTube Premium
- YouTube Music
- YouTube Kids

Naštartuj sa s nami v IT!

IT ACADEMY PREZENČNÉ KURZY www.it-academy.sk

VITA ONLINE KURZY www.vita.sk

Daj Odber

IT Academy



Odoberať

Domov Videá Shorts Naživo Zoznamy Komunita



Videá ► Prehrať všetko



Online kurz Data Science a Jazyk R - Jemný Úvod do...

7 zhliadnutí • pred 1 hodinou



Ako programuje Google v Python? Toto je docstring Dokumentačny retazec Toto je nasa dokumentacia ku zdrojovemu kodu... Autor: Larry Page Verzia: 1.0

8:38

Online kurz Python - Ako programuje Google v Python?

51 zhliadnutí • pred 1 dňom



Typy Grafov



Online Kurz Microsoft Excel Grafy a Typy Grafov

61 zhliadnutí • pred 2 dňami



Protokol
HTTPs
(v1/2/3)

http://

Online Kurz Počítačové Siete - Protokol HTTP(s), Stavové...

166 zhliadnutí • pred 3 dňami



Názory
Odporúčania
Hodnotenia



Online Kurz Efektívna Komunikácia - Názory a...

60 zhliadnutí • pred 3 dňami



MS Outlook

1. Duplicítne Emaily
2. Doplnky
3. Tipy a Triky

AKREDITOVANÝ KURZ

Microsoft Visual Basic

30:24

Online kurz Microsoft Outlook - Ako na...

142 zhliadnutí • pred 5 dňami

Oblíbené videá ► Prehrať všetko



SQL I. Začiatčník a. Databáza, Tabuľky, Select, Limit, Where, In, Between, And, Or, Komentár 11-2019

IT ACADEMY

1:51:23



Microsoft Excel - Kontingenčné Tabuľky a. Hárik, Duplicita, Zoradenie, Polia, Funkcie, Agregovanie, Stĺpce, Riadky, Zbalenie 2019

IT ACADEMY

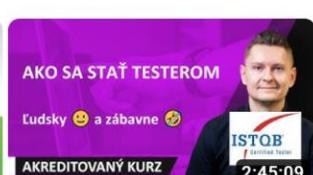
1:51:58



Siete I. Základy Siete a. Konzola, Ping, Ipconfig, Netstat, Sysinternals, TcpView, Traceroute 2018

IT ACADEMY

1:40:33



AKO SA STAŤ TESTEROM a. Ľudsky 😊 a zábavne 😊 ISTQB Internationale Testgruppe AKREDITOVANÝ KURZ

IT ACADEMY

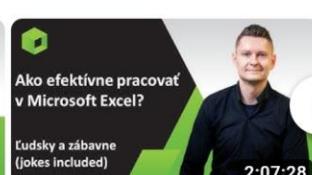
2:45:09



Microsoft Access I. Začiatčník a. Backstage, Databázy, Tabuľky, Porovnanie, Súťaž 03-2019

IT ACADEMY

1:46:27



Ako efektívne pracovať v Microsoft Excel? Ľudsky a zábavne (jokes included)

IT ACADEMY

2:07:28

Dajte odber na IT Academy



[www.YOUTUBE.COM/C/IT-ACADEMYSK](https://www.youtube.com/c/IT-ACADEMYSK)

