



Základy programování v jazyce Python

Instalace a úvod

Kryštof Mrózek (445429@mail.muni.cz)

Kristína Tomanková (kristinatomankova@mail.muni.cz)

Radoslav Brunovský (rbrunovsky@mail.muni.cz)

Instrukce k instalaci (pro Windows)

- Nainstalujte python
 - <https://www.python.org/downloads/>
- Nainstalujte Visual Studio Code
 - <https://code.visualstudio.com/docs/python/python-tutorial>
- Ve VSCode nainstalujte následující rozšíření
 - Python
 - Jupyter
- **Pokud jste zvyklí na jiný nástroj, klidně použijte ten**

Cíle předmětu

- V tomto předmětu se **nenaučíte** programovat
 - Pomůžeme vám s počátečními problémy a představíme základní koncepty
 - Programování ale musíte cvičit
- Ukázat schopnost Pythonu nahradit řadu jiných nástrojů
 - Excel - zpracování dat (NumPy, Pandas)
 - MATLAB - Numerické výpočty (SciPy)
 - GnuPlot, QtiPlot - vykreslování dat (Matplotlib, Plotly)
- Osvojit si základní návyky při psaní vlastního kódu
 - Nevymýšlejte, co už bylo vymyšleno
 - Používejte internet – Google, Stack Overflow, ChatGPT

Osnova

Orientační plán

1. Úvod, instalace
2. Proměnné, datové typy, funkce
3. Podmínky
4. Seznamy
5. Seznamy
6. Slovníky
7. Slovníky
8. Práce se soubory, výjimky
9. Rekurze
10. Třídy, objekty
11. Užitečné balíčky
12. Procvičování

Další Python předměty

- F4500 – Python pro fyziky
 - Předmět zaměřený na praktičtější použití pythonu
 - Lépe ukazuje široké možnosti Pythonu
- F5698 - Ultimátní příručka mladého budovatele v Pythonu
 - Pokročilejší předmět
 - Udržitelný vývoj softwaru
 - Moderní přístupy k programování

Co je Python?

- Z wikipedie

*Python is an interpreted **high-level** general-purpose programming language.*

“High-level” vs. “Low-level” jazyk

- High-level – blíž k uživateli (člověku)
- Low-level – blíž k počítači

Python – high-level jazyk

```
import random as rnd

#ask user for number of dice
nDice = input("How many dice are we playing with? \n")
nSides = input("\nhow many sides each dice has? \n")

msg = "we're rolling "+nDice+" dice. Here are the results:\n"

try:
    for x in range(int(nDice)):
        out = rnd.randint(1,int(nSides))
        print(out)
except:
    print("ops")
```

Assembler – low-level jazyk

```
MAINLINE EQU *
          ST   R14,SAVEML          SAVE REG 14
          SPACE
          MVI  SRDSF,C' '          CLEAR DSF STAN REC AREA
          MVC  SRDSF+1(L'SRDSF-1),SRDSF
          MVI  URRQID,C'1'         SET REQUEST ID TO 1
          MVI  URLVL,C'4'         SET REQUEST LEVEL TO 4
          MVC  URCMND,DEFPATH      MOVE IN DSF COMMAND
          MVC  URQLNM,PATHNAME     MOVE IN PATHNAME
          CALL DDCALL,(URRQAREA,DSFPATH)
          CLC  URRETURN,SPACES     Q-GOOD RETURN CODE?
          BE   GETIT              Y-CONTINUE
```

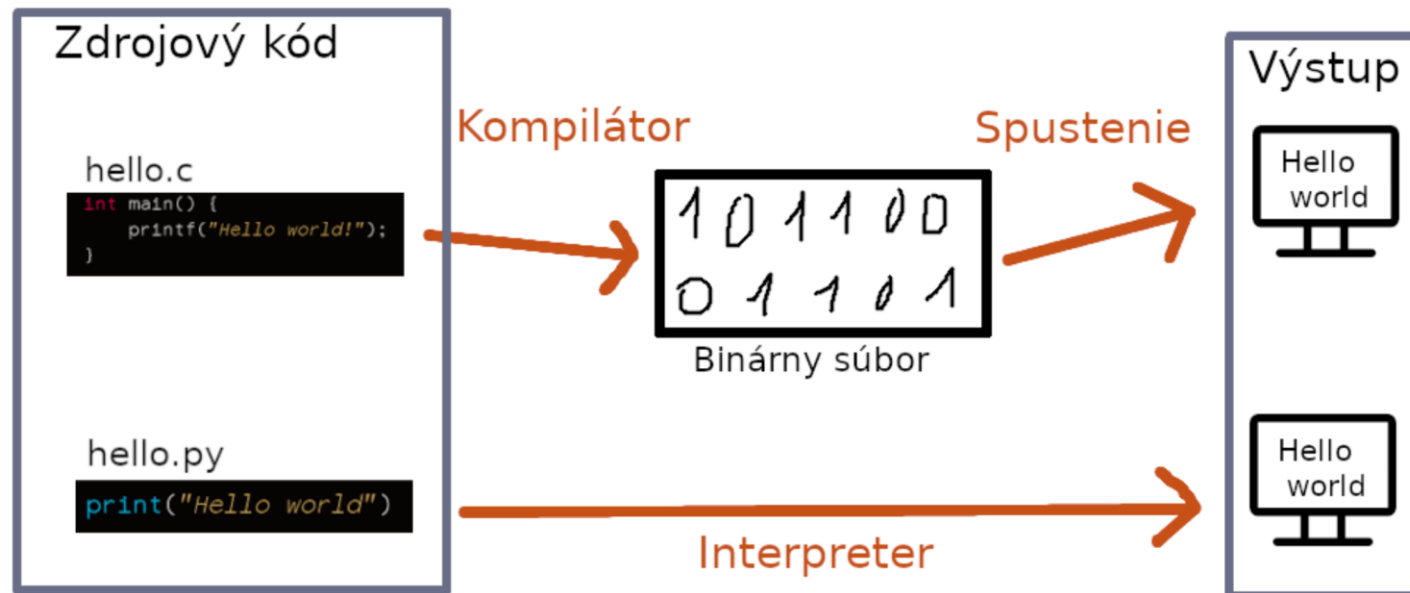
Co je Python?

- Z wikipedie

*Python is an **interpreted** high-level general-purpose programming language.*

“Interpretovaný” vs. “kompilovaný” jazyk

- V kompilovaných jazycích (C, C++, Java) se mezi napsáním kódu a jeho spuštěním musí provést další krok – tzn. kompilace, kdy se kód přeloží do strojového kódu -> spouští se strojový kód
- V interpretovaných jazycích (Python) se spouští přímo zdrojový kód, který je za běhu vyhodnocován „řádek po řádku“



Co je Python?

- Z wikipedie

*Python is an interpreted high-level **general-purpose** programming language.*

“General-purpose” vs. “Domain-specific”

- “General-purpose” jazyky lze využít na širokou škálu aplikací
 - Zpracování dat
 - Strojové učení
 - Web development, software development
- “Domain-specific“ jazyky mají úzké zaměření
 - HTML na tvorbu webových stránek
 - SQL pro komunikaci s databázemi

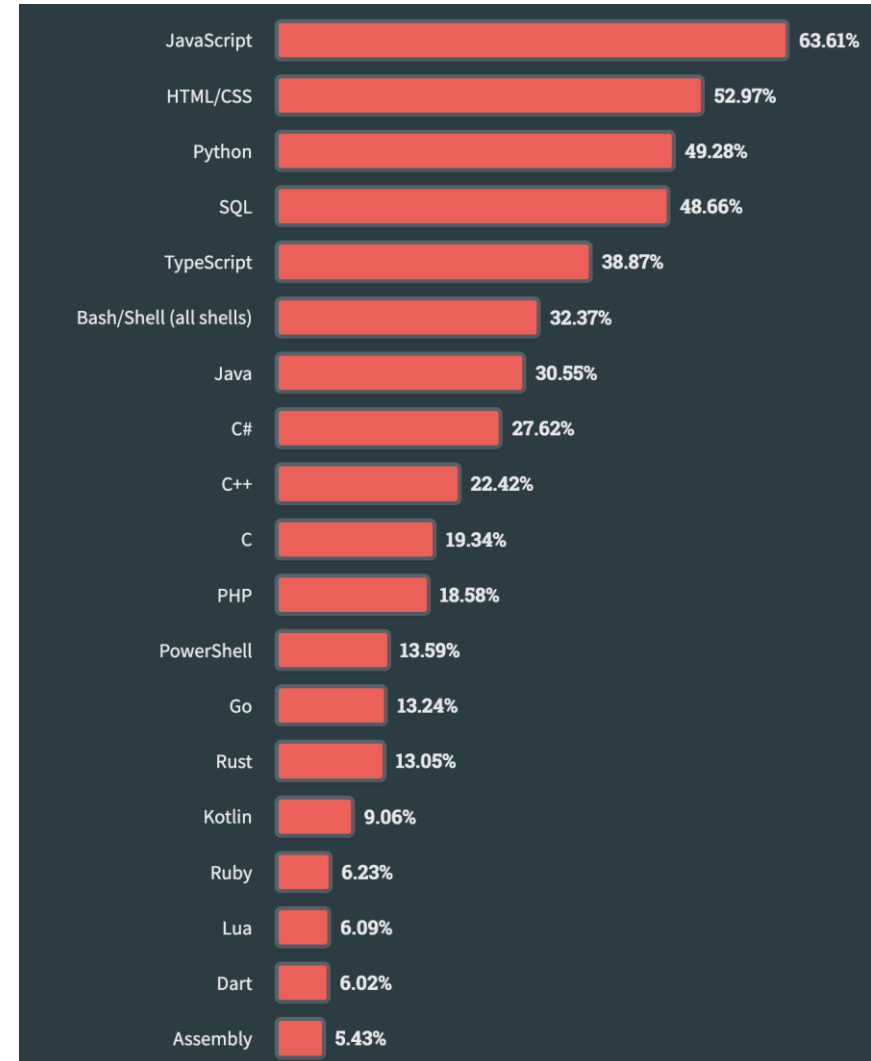
Proč Python?

- Uživatelsky přívětivý
 - Dynamické typování, “duck-typing“
- Vhodný pro začátečníky
 - Už po relativně krátké době budete schopni psát jednoduché **užitečné** programy
 - Zároveň ale v Pythonu budovat i složitější aplikace
- Volně dostupný, široce rozšířený
 - Obrovské množství různých balíčků (rozšíření)
 - Zpracování dat, machine learning, numerické výpočty, kvantové programování

Proč ne Python?

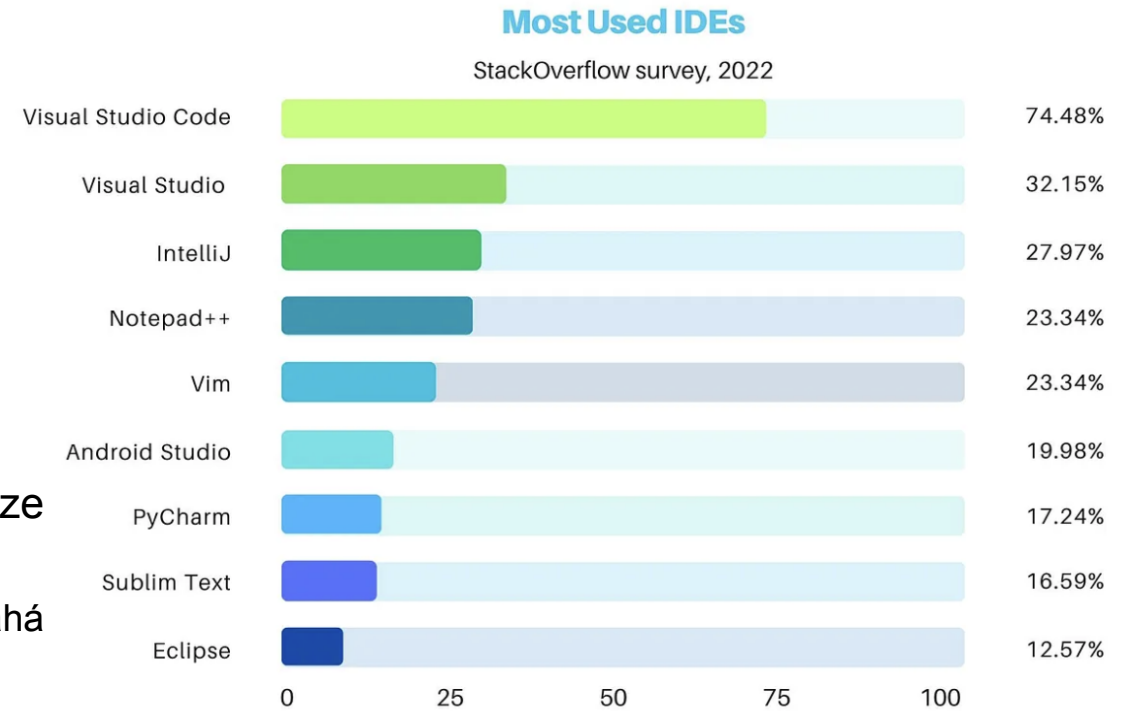
- Pomalý ve srovnání s kompilovanými, staticky typovanými jazyky
- Náchylný ke kritickým, nepředvídatelným chybám

Jazyky používané uživateli Stack OverFlow



Proč VSCode?

- Několik možností, jak Python používat
 - Příkazová řádka + textový editor (vim)
 - Jupyter notebook
 - **IDE (Integrated Development Environment)**
- VSCode je rychle rostoucí open-source vývojové prostředí
- V mnoha oblastech se stává standardem
- Vhodná pro začátečníky, rychlá instalace
- Obsahuje množství komplexních funkcí (další lze doinstalovat)
 - GitHub Copilot – AI integrovaná do IDE, která vám pomáhá programovat
- Umožňuje nainstalovat a používat Python bez nutnosti znát příkazovou řádku (terminál)
 - Příkazová řádka – přímý způsob komunikace s počítačem skrz textové příkazy
 - V tomto předmětu se jí nebudeme zabývat
 - Nicméně je dobré, vědět o co jde



Kombinace Python + VSCode vás dostane celkem daleko

Instalace balíčků ve VSCode

- Otevřete terminál ve VSCode (View > Terminal)
- Napište

```
py -m pip install <package>
```

Nebo

```
python -m pip install <package>
```

Varování:

- Na Unix systémech (Linux, MacOS) je správný (bezpečný) způsob instalace balíčků je do tzv. virtuálních prostředí (virtual environments)
 - pyenv, virtualenv
- V tomto předmětu je ignorujeme
- Instalace základních balíčků (NumPy, SciPy, Pandas, Plotly) by neměla způsobit žádné problémy
- Pokud budete instalovat exotičtější balíčky, nastudujte si virtuální prostředí