



1. Úvod

C2184 Úvod do programování v Pythonu
podzim 2022

Ondřej Schindler

Proč neabsolvovat C2184?

2. Uvedte číslem, kolik hodin celkem jste s tímto předmětem strávil(a) za tento semestr (účast na přednáškách a seminářích + cvičení a seminární práce + domácí příprava + učení na zápočet nebo zkoušku).

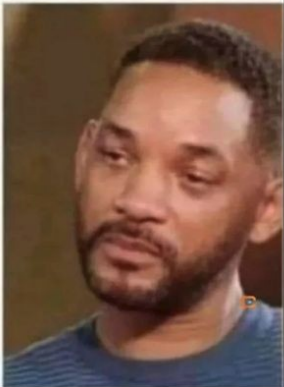
max	350
min	6
medián	77.5
průměr	92.3
fakultní medián	50
fakultní medián stejně kreditově hodnocených předmětů	45
počet kreditů tohoto předmětu	2

Coworker: How long did you sleep for?
Me: 8
Coworker: Hours?
Me:



First year in programming

Fourth year in programming

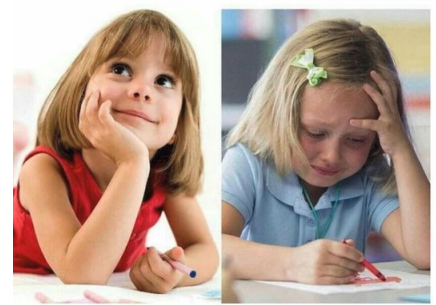


The only way to learn how to code is to spend time coding.



Thinking about a new program

Writing a new program



What people think programming is like



What programming is actually like



Proč absolvovat C2184?



Proč absolvovat C2184?

Python Programmer Information Technology

Gross monthly salary in Czech Republic is

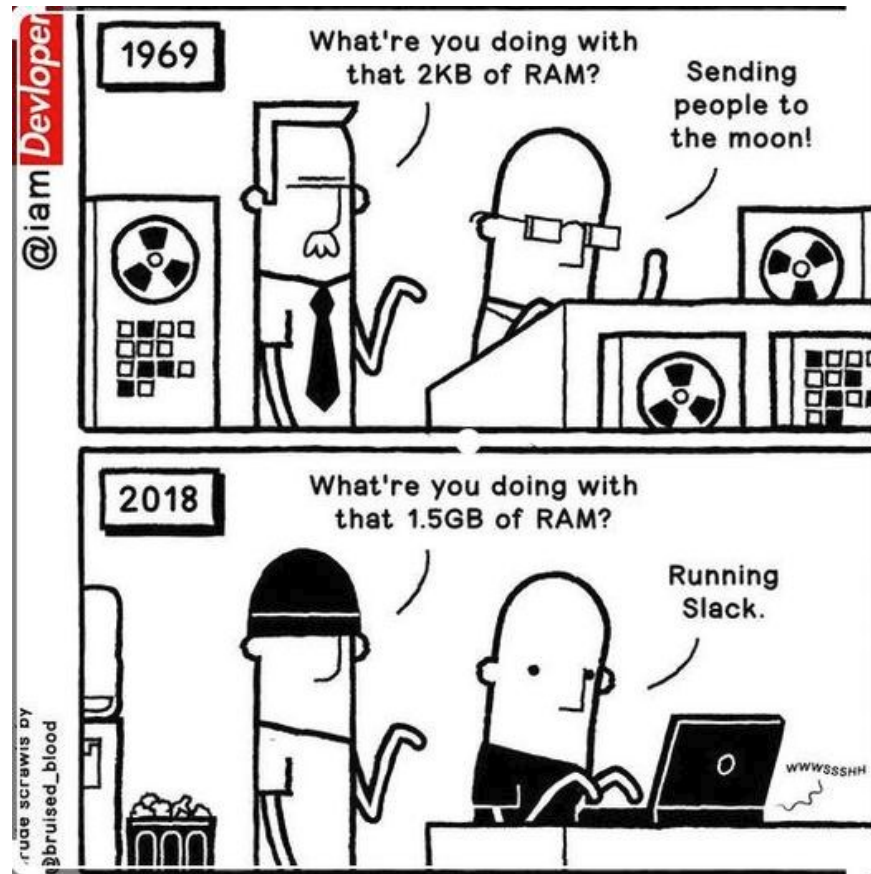
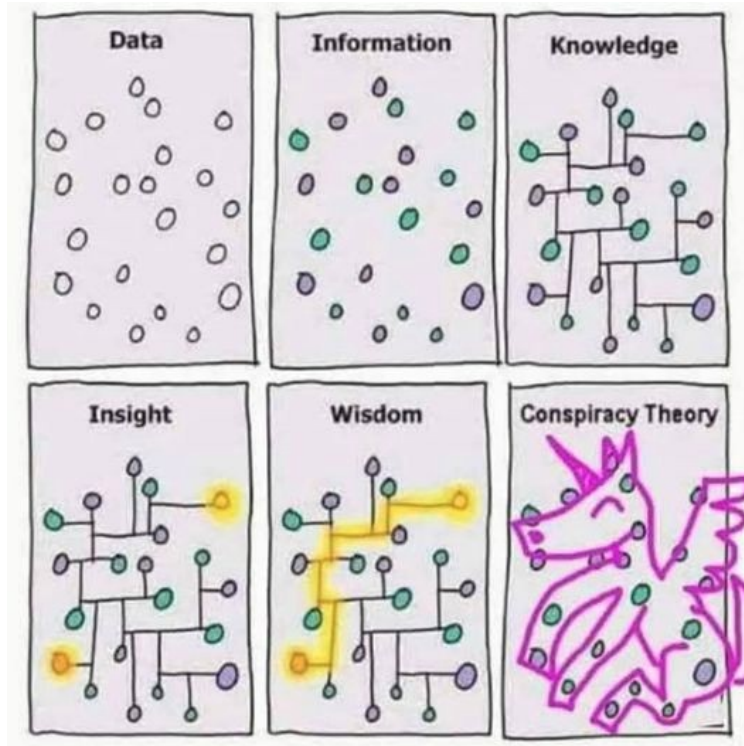


Compare your salary with others

The displayed value is just for information. Compare your salary with others to see more detailed information.

[Share](#) [Tweet](#) [Share](#) [Save](#) [Get link](#)

Proč absolvovat C2184?



Proč absolvovat C2184?

When you're a programmer and see a hacking scene in a movie



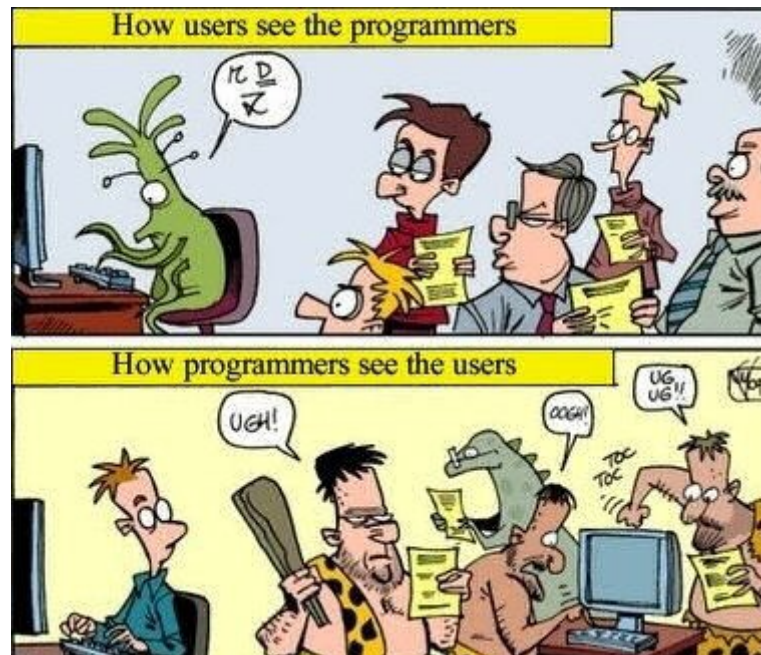
**LEARN PROGRAMMING
FOR
FUTUREWORK**

**LEARN PROGRAMMING
TO UNDERSTAND
R/PROGRAMMERHUMOR
JOKES**

Co je to programování?

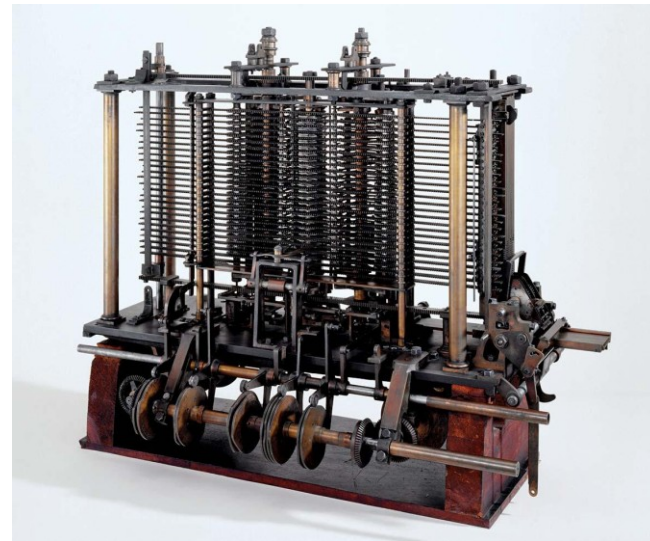
Programming is when you're describing to the machine how to do something.

Bill Gates



Historie

- 1837 – Charles Babbage navrhl Analytický stroj (Analytical Engine)
 - Všestranný mechanický počítač
 - Nikdy nebyl celý sestrojen :(



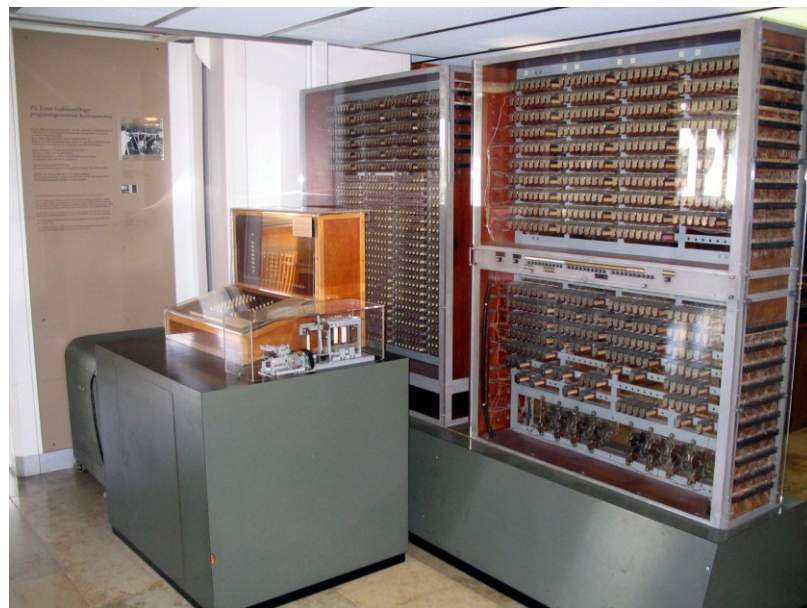
Část Analytického stroje

- Ada Lovelace
 - Popsala metodu, jak pomocí Analytického stroje spočítat Bernoulliho čísla
 - Považuje se za autorku prvního počítačového programu

Historie

40. léta 20. století

- Vznikají první počítače
- Používají se nízkoúrovňové programovací jazyky
 - Např. strojový kód, Assembly...
 - Složité pro psaní, specifické pro konkrétní typ procesoru



Replika počítače Z3 (1941)

Strojový kód

```
b8 21 0a 00 00
a3 0c 10 00 06
b8 6f 72 6c 64
a3 08 10 00 06
b8 6f 2c 20 57
a3 04 10 00 06
b8 48 65 6c 6c
a3 00 10 00 06
b9 00 10 00 06
ba 10 00 00 00
bb 01 00 00 00
b8 04 00 00 00
cd 80
b8 01 00 00 00
cd 80
```



Takhle vypadají soubory .exe
Tomuto rozumí procesor

Jazyk Assembly pro procesor IBM-PC (i386)

```
dosseg
.model small
.stack 100h

.data
hello_message db 'Hello, World!',0dh,0ah,'$'

.code
main proc
mov ax,@data
mov ds,ax

mov ah,9
mov dx,offset hello_message
int 21h

mov ax,4C00h
int 21h
main endp
end main
```

Historie

40.–50. léta

- První vysokoúrovňové jazyky
 - Umožňují vyšší míru abstrakce
 - Přeloží se do strojového kódu pomocí překladače (nemusí to dělat člověk)
- 1954 – jazyk Fortran (IBM)
 - První rozšířený vysokoúrovňový jazyk
 - Stále používán pro high-performance computing
- Další jazyky: LISP, COBOL...

Fortran

```
PROGRAM HELLOWORLD  
10 FORMAT (1X,11HHELLO WORLD)  
WRITE(6,10)  
END
```

Historie

60.–80. léta:

- Pascal, C, C++, MATLAB...

90. léta:

Python

```
print('Hello World!')
```

- Python, R, Java...
- Internet: PHP, JavaScript...

Současnost – stále nové jazyky:

- C# (2001), Go (2009), TypeScript (2012)...

Me: I wrote my first Hello World program!
People: Haha another naive beginne...
Me:



Vývoj Pythonu

- 1989 – Guido van Rossum začal pracovať na jazyce Python
 - Výzkumné centrum CWI v Holandsku
 - Jazyk pojmenován podle komické skupiny Monty Python



• 1994 – Python 1.0

• 2000 – Python 2.0



• 2008 – **Python 3.0**

- Cíl: napravit chyby v návrhu Pythonu
- Není zpětně kompatibilní v Pythonem 2!
- Python 2 stále přežívá (kvůli množství existujícího kódu)
- ... Konec podpory – 1.1.2020

• 6.9.2023 – Python 3.12



Proč Python?



Python

The Ent

DIFFICULTY



Help little Hobbits (beginners) to understand programming concepts

Help Wizards (computer scientists) to conduct researches

Widely regarded as the best programming language for beginners

Easiest to learn

Widely used in scientific, technical & academic field, i.e. Artificial Intelligence

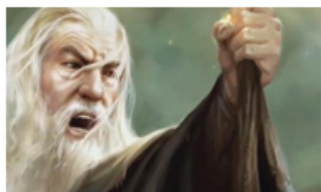
You can build website using Django, a popular Python web framework



Java

Gandalf

DIFFICULTY



Wants peace & works with everyone (portable)

Very popular on all platforms, OS, and devices due to its portability

One of the most in demand & highest paying programming languages

Slogan: write once, work everywhere



C

One Ring

DIFFICULTY



The power of C is known to them all

Everyone wants to get its Power

Lingua franca of programming language

One of the oldest and most widely used language in the world

Popular language for system and hardware programming

A subset of C++ except the little details



C++

Saruman

DIFFICULTY



Everyone thinks that he is the good guy

But once you get to know him, you will realize he wants the power, not good deeds

Complex version of C with a lot more features

Widely used for developing games, industrial and performance-critical applications

Learning C++ is like learning how to manufacture, assemble, and drive a car

Recommended only if you have a mentor to guide you



JavaScript

Hobbit

DIFFICULTY



Frequently underestimated (powerful)

Well-known for the slow, gentle life of the Shire (web browsers)

"Java and Javascript are similar like Car and Carpet are similar" - Greg Hewgill

Most popular clients-side web scripting language

A must learn for front-end web developer (HTML and CSS as well)

One of the hottest programming language now, due to its increasing popularity as server-side language (node.js)

Proč Python?

-
- Přehledná syntaxe
- Velká oblíbenost
 - Dokumentace, Stack Overflow
 - Množství tutoriálů a kurzů
 - Velké množství balíčků, všestranné nasazení
- jazyk
 - struktury
 -
- Využití ve vědě
 - Bioinformatika: Biopython, scikit-bio, gemmi, ...
 - Matematika: scipy, matplotlib, numpy, numba, bokeh, ...
 - Strojové učení: scikit-learn, TensorFlow, ...
 - Zpracování obrazu: scikit-image, ...

Python is the
easier language
to learn.
No brackets,
no main.



You get errors
for writing an
extra space



Přehledná syntaxe

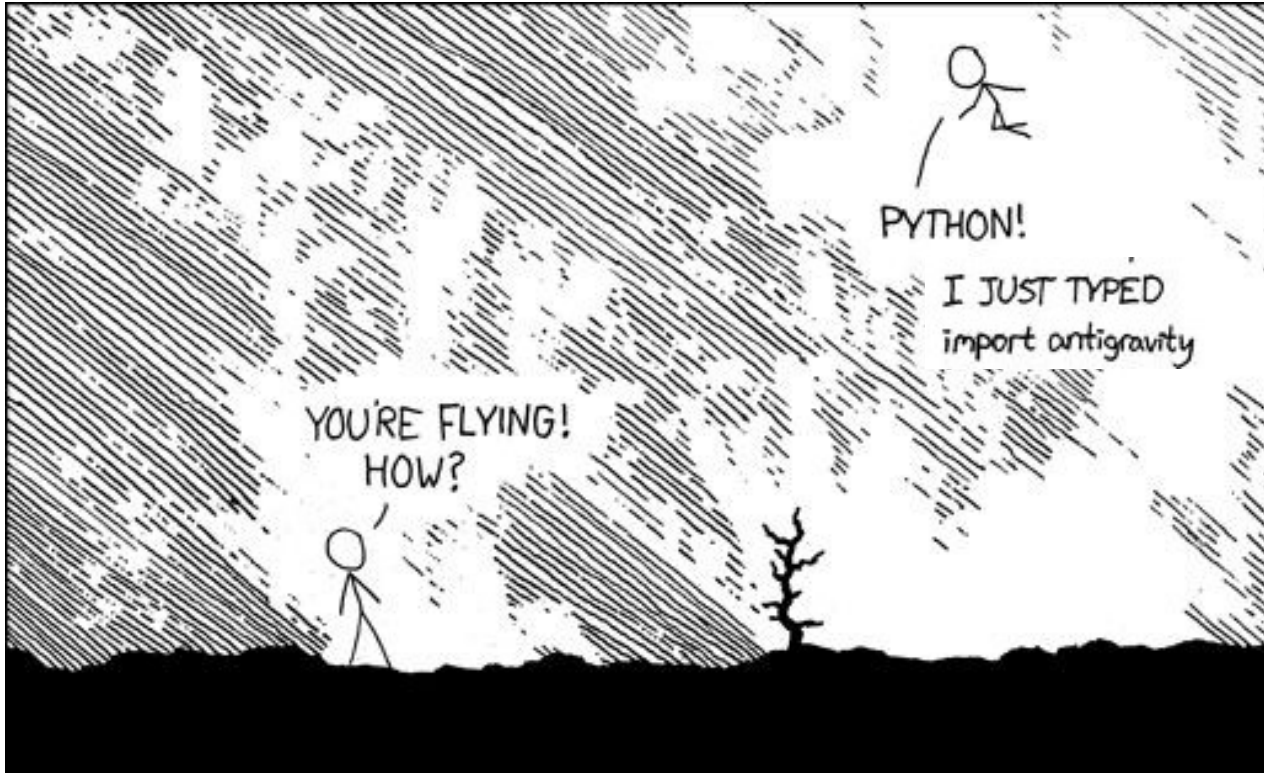
Funkce pro seřazení čísel:

```
def insert_sort(numbers):  
    '''Sort numbers from the smallest to the greatest.'''  
    for i in range(1, len(numbers)):  
        key = numbers[i]  
        # Move elements of numbers[0..i-1], that are greater than key,  
        # to one position ahead of their current position  
        j = i-1  
        while j >= 0 and key < numbers[j]:  
            numbers[j+1] = numbers[j]  
            j -= 1  
        numbers[j+1] = key
```

Poznámka: tuto funkci by bylo zbytečné psát, protože to už někdo udělal za nás:

```
numbers.sort()
```

Síla importu



Oblíbenost

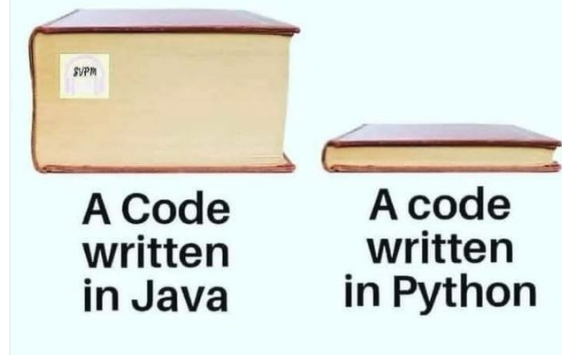
[PYPL index](#) – 1. místo

[TIOBE index](#) – 1. místo



Filozofie – jednoduchost a konvence

The Pythonic way:



There should be one – and preferably only one – obvious way to do it.

```
i = 0
while i < len(my_list):
    do_something(my_list[i])
    i = i + 1
```

Bude fungovat

```
for i in range(len(my_list)):
    do_something(my_list[i])
```

Bude fungovat

```
for item in my_list:
    do_something(item)
```

Jednodušší a přehlednější způsob
(the Pythonic way)

Slabiny Pythonu

Rychlost

- Pomalejší než třeba C nebo Java :(
- Existují způsoby, jak napsat rychlý kód :)
- Pravděpodobně na tento problém nenarazíte

Spotřeba paměti

- Pravděpodobně na tento problém nenarazíte

Runtime errors

- Je to dynamicky typovaný jazyk, proto z kódu nelze poznat, jaké chyby se mohou vyskytnout při běhu programu (lze řešit např. pomocí *mypy*)

Méně vhodný pro mobilní aplikace a databáze

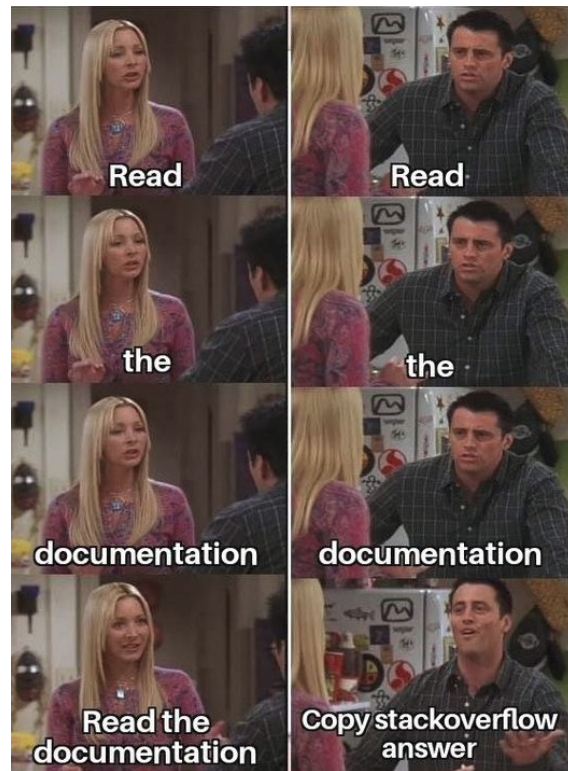
Slowest things on earth:



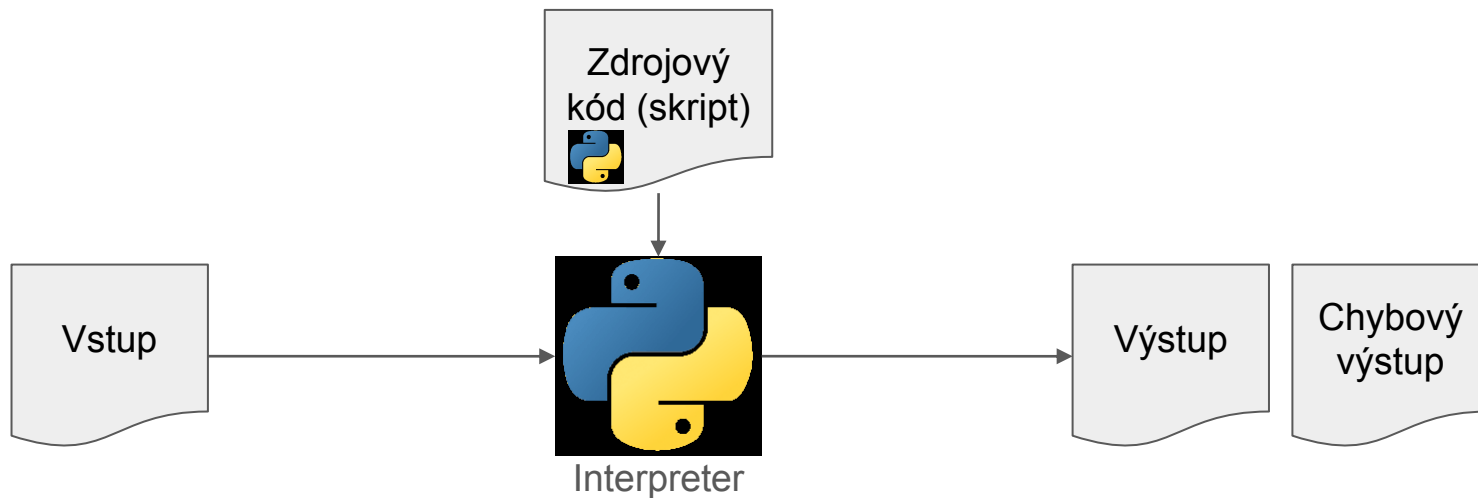
Co je vlastně Python?

Python je

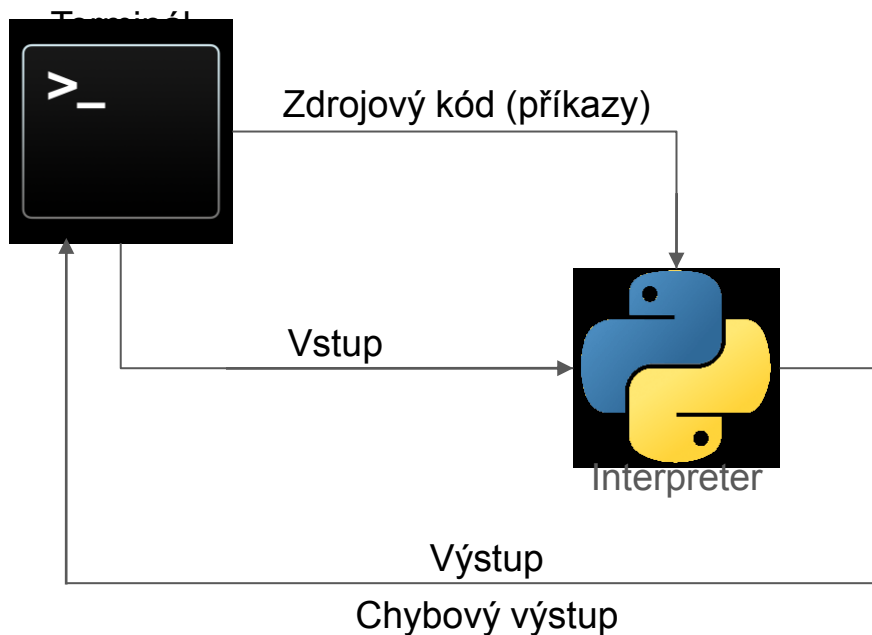
- Programovací jazyk
 - Jazyk, který umožňuje zápis zdrojového kódu programu
 - Jeho syntax a sémantika je popsána [dokumentací](#)
 - Konkrétně verze *Python 3.12*
- Interpreter
 - Program, který umožňuje spouštění (interpretaci) zdrojového kódu napsaného v jazyce Python
 - Konkrétně *CPython 3.12* (existují i další interpretery, např. *PyPy*, *Jython*)



Jak funguje interpreter?




Interpreter v interaktivním módu



Rozdíly od neinteraktivního módu:

- Zdrojový kód se zadává z terminálu
- Pro zadání příkazů se vypisuje prompt (>>>)
- Výsledky příkazů se vypisují na výstup

Co je IDE?

- IDE = Integrated Development Environment = vývojové prostředí
- “Chytrý editor”
- Ulehčuje nám práci při programování
- My budeme používat **Visual Studio Code** 
 - Zadarmo
 - Rychlé
 - Není vázané na konkrétní jazyk
 - Lze do něj přidávat funkcionalitu pomocí extensions (využijeme extension **Python**)
- Jiná IDE: *PyCharm, Spyder...*

insert_sort.py - 2019 - Visual Studio Code

File Edit Selection View Go Debug Terminal Help

EXPLORER

OPEN EDITORS

- insert_sort.py

2019

- .ipynb_checkpoints
- Standa_videa_2018
- ~lock.plan_semestr...
- 1920px-Python_logo...
- 159193_2_text.step
- 159193_3_video.step
- Ada.jpg
- insert_sort.py**
- kisspng-programmin...
- NCBR podzim 2018.x...
- plan_semestra_v2.0....
- portion-Charles-Bab...
- python_logo.png
- python_logo2.png
- s1.step
- s2.step
- s3.step
- s4.step
- sci-lg-eng-rgb.png
- sci-la-text-cze-rab.png

insert_sort.py

```
1 def insert_sort(numbers):
2     '''Sort numbers from the smallest to the greatest.'''
3     for i in range(1, len(numbers)):
4         key = numbers[i]
5         # Move elements of numbers[0..i-1], that are greater than key,
6         # to one position ahead of their current position
7         j = i-1
8         while j >=0 and key < numbers[j]:
9             numbers[j+1] = numbers[j]
10            j -= 1
11            numbers[j+1] = key
12
13 numbers = [2, 5, 8, 0, 1]
14 numbers.sort()
15 print(numbers)
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

2: Python

```
adam ~/School/Praca/Python/2019 $ /usr/bin/python3
Python 3.6.5 | packaged by conda-forge | (default, Apr 6 2018, 13:39:56)
[GCC 4.8.2 20140120 (Red Hat 4.8.2-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> numbers = [2, 5, 8, 0, 1]
>>> numbers.sort()
>>> print(numbers)
[0, 1, 2, 5, 8]
>>>
```

Python 3.6.5 64-bit ('usr': virtualenv) 0 0

Ln 12, Col 1 Spaces: 8 UTF-8 LF Python

Instalace Pythonu a VSCode – Windows

Python

- Stáhněte Python 3.12 z <https://www.python.org/downloads/>
- Důležité: na začátku instalace zaškrtněte **Add Python 3.12 to PATH**
- V jednom z dalších kroků zaškrtněte **Disable path length limit**
- Kontrola: v nabídce Start najděte Python 3.12 – měl by se otevřít terminál

VSCode

- Stáhněte z <https://code.visualstudio.com/>
- Při instalaci také zaškrtněte **Add to PATH**
- Spusťte VSCode a na levé liště klikněte na ikonu Extensions
- Vyhledejte a nainstalujte extension *Python* od autora Microsoft
 - Automaticky se nainstalují i další extensions: *Pylance*, *Jupyter*, *Jupyter Keymap*
- Restartujte VSCode

Instalace Pythonu a VSCode – Linux, Mac OS X

Obdobná instalace

Ve většině distribucí Linuxu je už Python nainstalován, ale jmenuje se `python3`

- Starší verze, do 2019 (Ubuntu 19.10), starší Macy:
 - `python3` = Python 3 (`python` = Python 2 → nepoužívat)
- Novější verze, od 2020 (Ubuntu 20.04):
 - `python3` = Python 3 (po instalaci balíku `python-is-python3`: `python` = `python3` = Python 3)
- Doinstalujte si balík `python3-pip` (přes `sudo apt install python3-pip`)

Instalace VSCode může být složitější (<https://code.visualstudio.com/docs/setup/setup-overview>)

První kroky

- Vytvořte si složku, ve které si budete ukládat svoje programy během semestru
- Stáhněte si `cviko_01.zip` z interaktivní osnovy a rozbalte do své složky
- Ve VSCode na levé liště klikněte na ikonu Explorer a otevřete složku `cviko_01`
 - Otevření složky: zkratka `Ctrl+K Ctrl+O`
- Na horní liště klikněte Terminal > New Terminal, otevře se vám terminál s shellem (*bash* nebo *powershell*)
 - Terminál lze skrýt / znovu otevřít zkratkou `Ctrl+J`

Základní příkazy shellu

- `ls` – vypiš obsah aktuální složky (nebo ve Windows `dir`)
- `cd ..` / `cd homeworks` – jdi o složku výš / jdi do podsložky homeworks
- `pwd` – vypiš mi kde jsem
- `python` – spust' program blabla
- `man python` – vypiš nápovědu k příkazu/programu python (Linux, MacOS)
- `help python` – `– || –` (Windows)
- `exit` – ukonči shell a zavři terminál

Klávesy `↑↓` – návrat k předchozím příkazům

`Tab` – automatické doplnění příkazu / programu / názvu souboru / složky

Spuštění Pythonu v interaktivním módu

- Do terminálu ve VSCode napište `python`, stiskněte Enter
- Vypíše se verze Pythonu a objeví se prompt (`>>>`)
- Zkuste zadávat příkazy, např. `2+2` nebo `print('Hello')`
- K předchozím příkazům se umíte vrátit pomocí šipek `↑↓` na klávesnici
- Python ukončíte signálem end-of-file (Windows: `Ctrl+Z` a Enter; Linux, Mac: `Ctrl+D`), nebo zadáním příkazu `exit()`

Úkol

- Spustíte Python v interaktivním módu a vypočítejte $(123 + 4/5) * 6$
 - A) 127.8
 - B) 738
 - C) 742.8
 - D) 9



Spuštění skriptu

- V nabídce Explorer ve VSCode vytvořte nový soubor `hello.py` a otevřete ho (pythonovské skripty mají příponu `.py`)
- Do souboru napište `print('Hello World!')` a uložte (* nebo • nahoře u názvu souboru znamená, že není uložen)
- Do terminálu zadejte `python hello.py` (shell umí dokončit název souboru stisknutím Tab)
- Spustí se Python a vypíše `Hello World!`
- Python se hned i ukončí, nevidíte `>>>`

- Trik: aktuální řádek nebo označenou část skriptu lze spustit přes `Shift+Enter`

Úkol

- Spust'te skript `ukazkovy_skript.py`. Co se vypíše?

A) 2 5 8 0 1

B) 0 1 2 5 8

C) 8 5 2 1 0

D) 0 2 5 8 4

Jupyter Notebook

- Nástroj, který umožňuje kombinovat text, zdrojový kód, rovnice, grafy ... v jednom dokumentu (notebooku, přípona `.ipynb`).
- Notebook se skládá z buněk dvou typů
 - *Markdown* – obsahují formátovaný text
 - *Code* – obsahují zdrojový kód
- Na pozadí běží interaktivně interpreter, který vykonává kód v buňkách (*kernel*)
- Výstup interpreteru se zobrazuje přímo v notebooku
- Buňky lze upravovat, přesouvat, mazat a opakovaně spouštět

Spuštění Pythonu přes Jupyter notebooks

- V nabídce Explorer ve VSCode otevřete soubor `ukazkovy_notebook.ipynb`, potvrďte všechny výzvy k instalaci doplňků a pod. – může to chvíli trvat
- Zkuste spouštět a upravovat buňky v notebooku
- Možné problémy:
 - Notebook je otevřený v Restricted módu
→ při otevření složky potvrdit “Yes, I trust the authors”
 - Není zvolený kernel (pokud máme nainstalovaných více verzí Pythonu)
→ vpravo nahoře zvolit kernel Python 3
 - Není nainstalovaný balík *ipykernel*
→ potvrdit výzvu k instalaci

Úkol

- Spustíte poslední buňku v notebooku `ukazkovy_notebook.ipynb`. Co vypíše?
 - A) Python Rules
 - B) Secret Message
 - C) Spanish Inquisition
 - D) 9

Ukázkový domácí úkol

- Do své složky si stáhněte a rozbalte `cviko_01.zip`
- Ve VSCode otevřete `du_1.ipynb`
- Řešte zadaný úkol
- Řešení si můžete zkontrolovat spuštěním testovací buňky na konci notebooku, před tím ale nezapomeňte notebook uložit (nebo si nastavte `File > Auto Save`)
- Když vám projdou testy (vypíše se `Tests passed`), můžete odevzdat notebook do odevzdáárny v ISu
 - Vždy odevzdávejte soubor s původním názvem, jinak se vám nezapočtou body
- Body za DÚ budeme cca 1x za týden zapisovat do poznámkových bloků v ISu

Alternativní odevzdávání domácích úkolů

Kdo nechce používat Jupyter notebooks:

- Stejně zadání jako v `du_X.ipynb` naleznete v `du_X.pdf`
- Jednotlivé úkoly řešte v souborech `du_X_Y.py`
 - např. DÚ 2.5 řešte do souboru `du_2_5.py`
- Testy si spustíte z příkazového řádku:
`python testing.py du_X_Y.py`
- Odevzdejte jednotlivé soubory nebo je zabalte jako ZIP (ne RAR, TAR, 7Z...).

Domácí úkoly – časté problémy

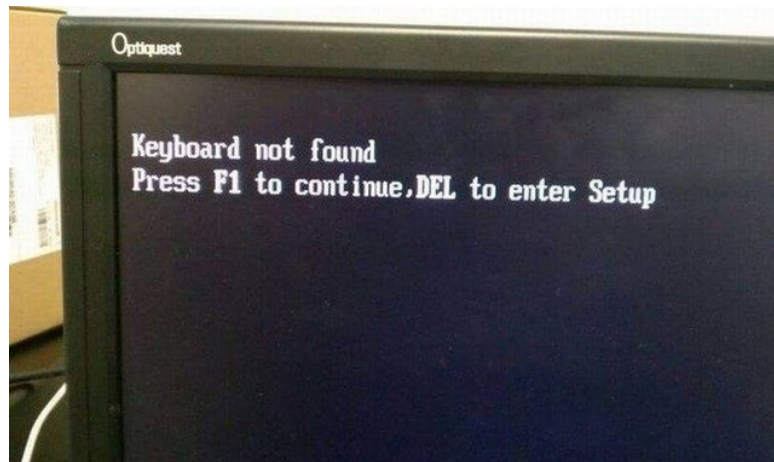
- Nepřejmenovávejte odevzdané soubory.
Nepřipisujte jméno, učo, ...
Opravné skripty vám pak nepřihadí body!
- Vždy otestujte před odevzdáním úkol!
Hello world! & hello world!
- Dodržujte deadline!



Užitečné triky

VSCoDe

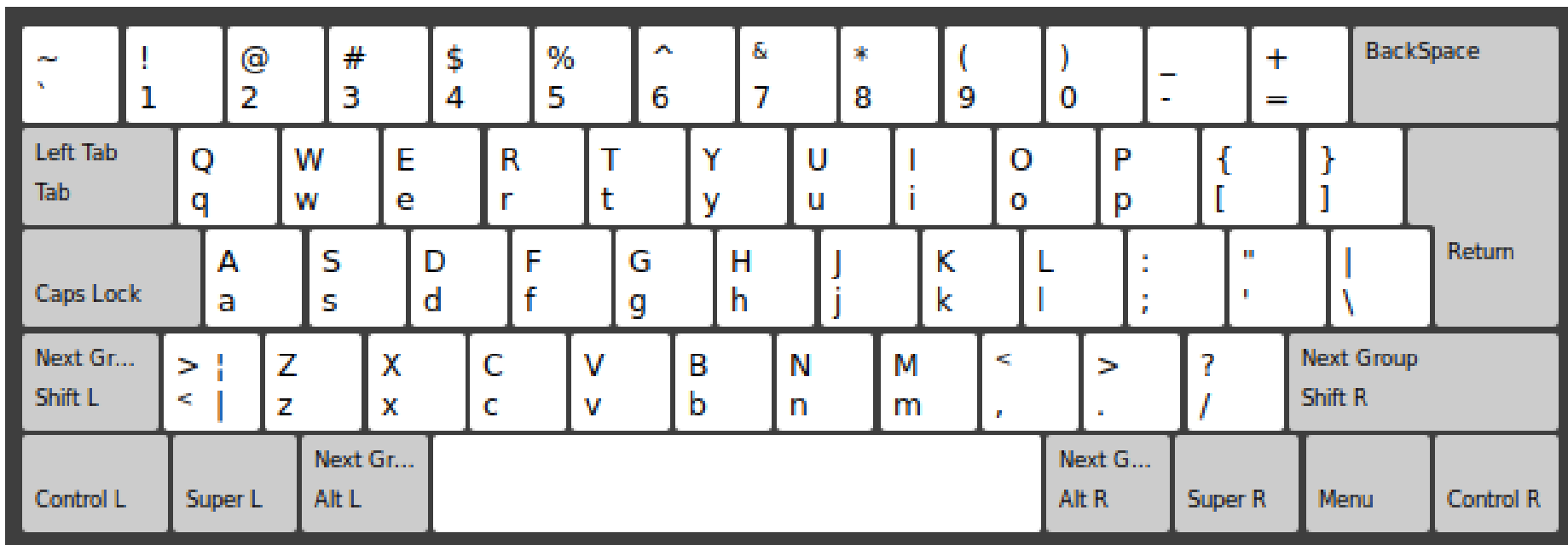
- Ctrl+J – zobrazí/skryje terminál
- Ctrl+/ – zakomentuje/odkomentuje řádek (text za # interpreter ignoruje)
- Ctrl+X – smaže řádek (Ctrl+C, Ctrl+V kopíruje aktuální řádek)
- Ctrl+K, Ctrl+O – otevření složky
- Ctrl+↑↓ – scrollování
- Alt+↑↓ – přesouvání aktuálního řádku
- 2x klik – označí celé slovo
- 3x klik – označí celý řádek



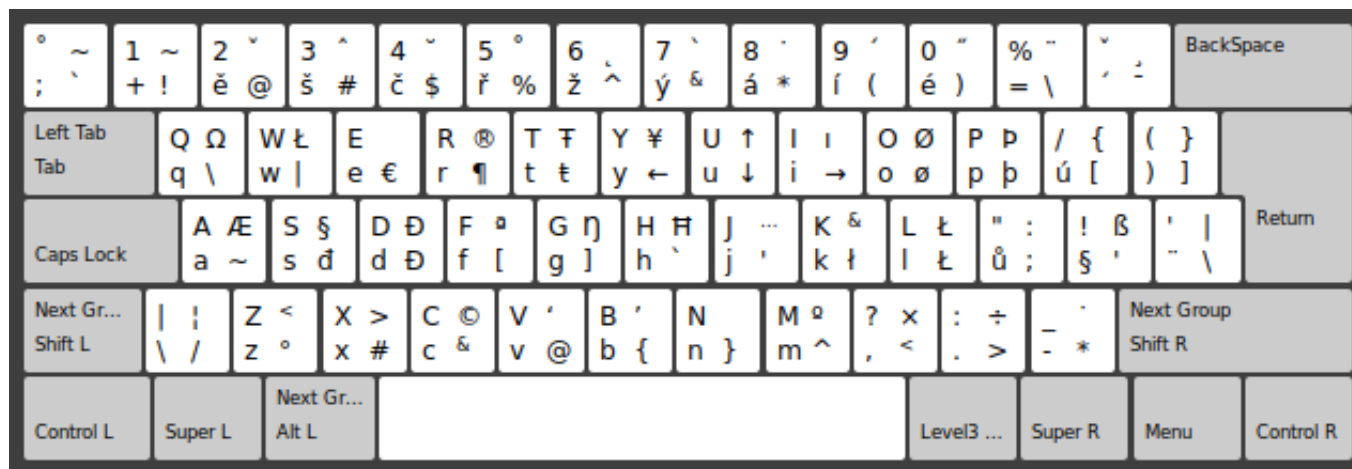
Doporučení: používejte anglickou klávesnici

Obsahuje všechny speciální znaky používané při programování

@ # % ^ & () [] { } < > + - = * / | \ : ; ' " ` _ ...



Česká



Slovenská

