

(BEZBOLESTNÝ) ÚVOD DO PROGRAMOVÁNÍ V PYTHONU

Národní technická knihovna
Praha

Pavel Synek

3.11. 2021

LEKTOR

- Pavel Synek
 - ↩ 2011-2019 - EBSCO Information Services
 - ↩ 2020 - Keboola Czech
 - ↩ 2021 -> Národní technická knihovna
- Python background
 - ↩ Coursera - od roku 2015
 - ↩ online kurzy - Codecademy, DataQuest, DataCamp
 - ↩ SoloLearn, ProjectEuler.com, Kaggle.com, ...
 - ↩ DMOJ - online judge server



ÚČASTNÍCI WEBINÁŘE

- Kdo jste?
- Odkud přicházíte?
- Jaká je Vaše motivace a očekávání od tohoto webináře?



OBSAH ŠKOLENÍ

Dnešní program

1. Proč Python
2. Softwarové vybavení na práci s Pythonem
 - Instalace
 - Specifika práce s jednotlivými nástroji
3. Lehký úvod
4. Skriptování v praxi
 - Minimum teorie a maximum psaní vlastních skriptů
5. Jak dál?
 - Vlastní projekty
 - Nástroje a platformy
 - DataCamp

PROČ PROGRAMOVAT?



- Většina lidí dnes umí pracovat s počítačem. Znalost MS Office je brána jako standard. Někdy se ale nespokojíme s tím, jaké možnosti nám hotové softwarové nástroje nabízejí...

PROČ PROGRAMOVAT?

- Hlavně klid...programování nemusí nutně znamenat doktorát v počítačové vědě

```
ValueError                                Traceback (most recent call last)
<ipython-input-3-fd9d5c5ef701> in <module>()
----> 1 b = tf.reverse(a, axis = 0)

/usr/local/lib/python2.7/dist-packages/tensorflow/python/ops/array_ops.py in reverse(tensor, axis, name)
2332
2333 def reverse(tensor, axis, name=None):
-> 2334     return gen_array_ops.reverse_v2(tensor, axis, name)
2335 reverse.__doc__ = gen_array_ops.reverse_v2.__doc__
2336

/usr/local/lib/python2.7/dist-packages/tensorflow/python/ops/gen_array_ops.py in reverse_v2(tensor, axis,
name)
2697     """
2698     result = _op_def_l1b.apply_op("ReverseV2", tensor=tensor, axis=axis,
-> 2699                                 name=name)
2700     return result
2701

/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/op_def_library.py in apply_op(self, op
_type_name, name, **keywords)
765     op = g.create_op(op_type_name, inputs, output_types, name=scope,
766                     input_types=input_types, attrs=attr_protos,
-> 767                     op_def=op_def)
768     if output_structure:
769         outputs = op.outputs

/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/ops.py in create_op(self, op_type, inp
uts, dtypes, input_types, name, attrs, op_def, compute_shapes, compute_device)
2506         original_op=self._default_original_op, op_def=op_def)
2507     if compute_shapes:
-> 2508         set_shapes_for_outputs(ret)
2509     self._add_op(ret)
2510     self._record_op_seen_by_control_dependencies(ret)

/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/ops.py in set_shapes_for_outputs(op)
1871     shape_func = _call_cpp_shape_fn_and_require_op
1872
-> 1873     shapes = shape_func(op)
1874     if shapes is None:
1875         raise RuntimeError(

/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/ops.py in call_with_requiring(op)
1821
1822 def call_with_requiring(op):
-> 1823     return call_cpp_shape_fn(op, require_shape_fn=True)
1824
1825     _call_cpp_shape_fn_and_require_op = call_with_requiring

/usr/local/lib/python2.7/dist-packages/tensorflow/python/framework/common_shapes.py in call_cpp_shape_fn(
op, input_tensors_needed, input_tensors_as_shapes_needed, debug_python_shape_fn, require_shape_fn)
608     res = _call_cpp_shape_fn_impl(op, input_tensors_needed,
609                                   input_tensors_as_shapes_needed,
-> 610                                   debug_python_shape_fn, require_shape_fn)
611     if not isinstance(res, dict):
612         # Handles the case where _call_cpp_shape_fn_impl calls unknown_shape(op).
```

PROČ PYTHON

1. Jednoduchý syntax
2. Flexibilita a modularita
3. Rozšířená a aktivní komunita uživatelů (mezi nimi i velká korporátní jména)
4. Široké využití v oblasti big data
5. Dynamický vývoj a velké množství knihoven (<https://pypi.org/>)

PROČ PYTHON



NAČ LZE PYTHON VYUŽÍT?

- Strojové učení
- Datová analýza
- Vývoj webu
- Automatizace procesů
- Testování softwaru a vývoj propotypů
- ...



PROČ PYTHON

- Kdo s ním pracuje....

- ↩ NASA
- ↩ Google
- ↩ Facebook
- ↩ IBM
- ↩ Netflix
- ↩ Amazon
- ↩ Walt Disney
- ↩



PROČ PYTHON

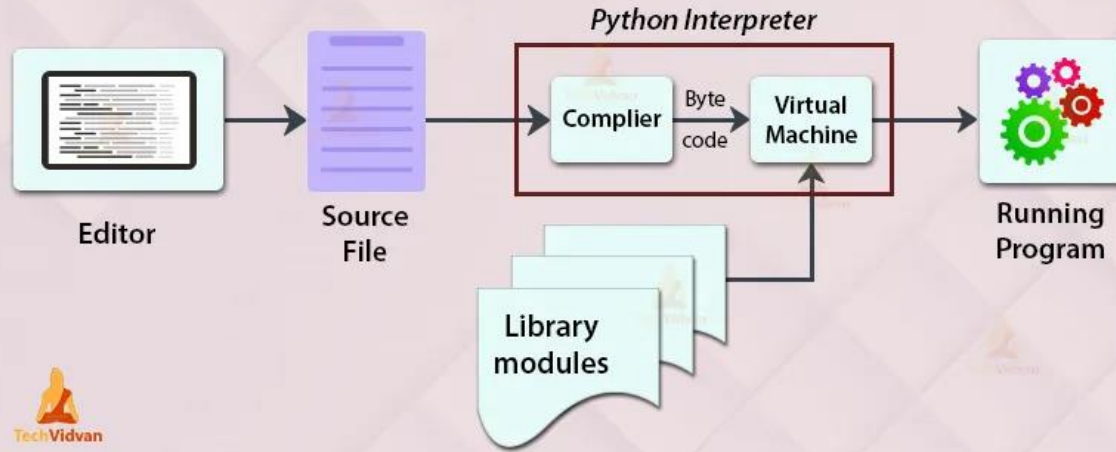
- V pořadí jazyků vyučovaných na základní škole se více dětí učí Python než francouzštinu...

....ŽE NEZNÁTE PYTHON?

- vývoj započal v 80. letech z nudy
- první vydání v r. 1991
- nyní verze Python 3.10
- Filozofie Pythonu - https://cs.wikipedia.org/wiki/Filozofie_Pythonu



How Python Interpreter Works?



JAK PYTHON FUNGUJE

INSTALACE POTŘEBNÉHO (VHODNÉHO) SOFTWAREVÉHO VYBAVENÍ

Instalace Python do počítače

- <https://www.python.org/>

Instalace Jupyter Notebook

- <https://jupyter.org/>

JetBrains - kompletní vývojářské prostředí

- <https://www.jetbrains.com/>

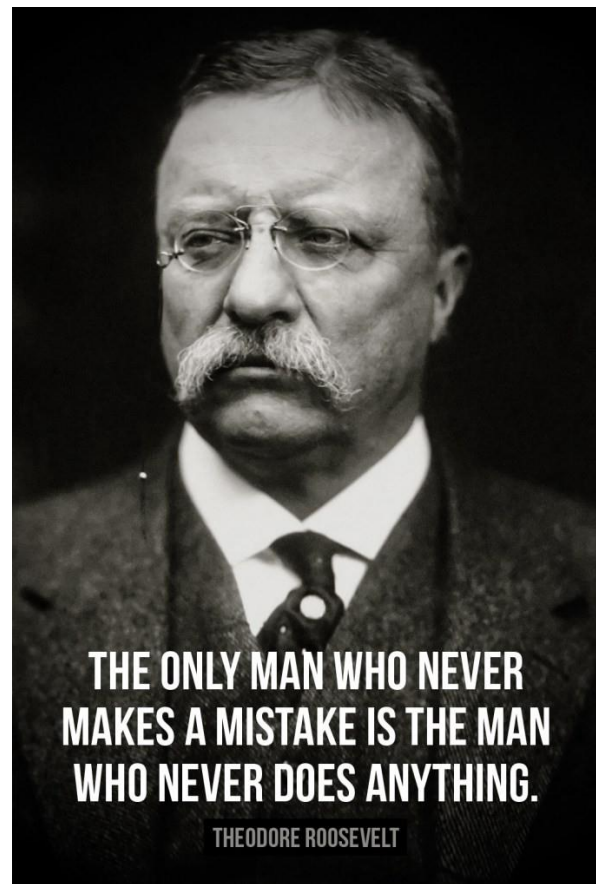
DataCamp - workspace

- <https://www.datacamp.com/>

CO SE DNES V PRAXI NAUČÍM?

- absolutní základy
 - ↪ definice proměnných a práce s nimi
 - ↪ vlastní vstupy
 - ↪ práce s “metodami”
 - ↪ booleovské operátory
 - ↪ podmínky IF, ELIF, ELSE
 - ↪ Smyčky (LOOPS)
 - ↪ využití knihoven

CO ŘÍCI DO ZAČÁTKU?



CO ŘÍCI DO ZAČÁTKU?

- Najděte si v Pythonu vlastní “proč”...vlastní smysl
- to za Vás nikdo neudělá...

A TEĎ DO PRAXE....



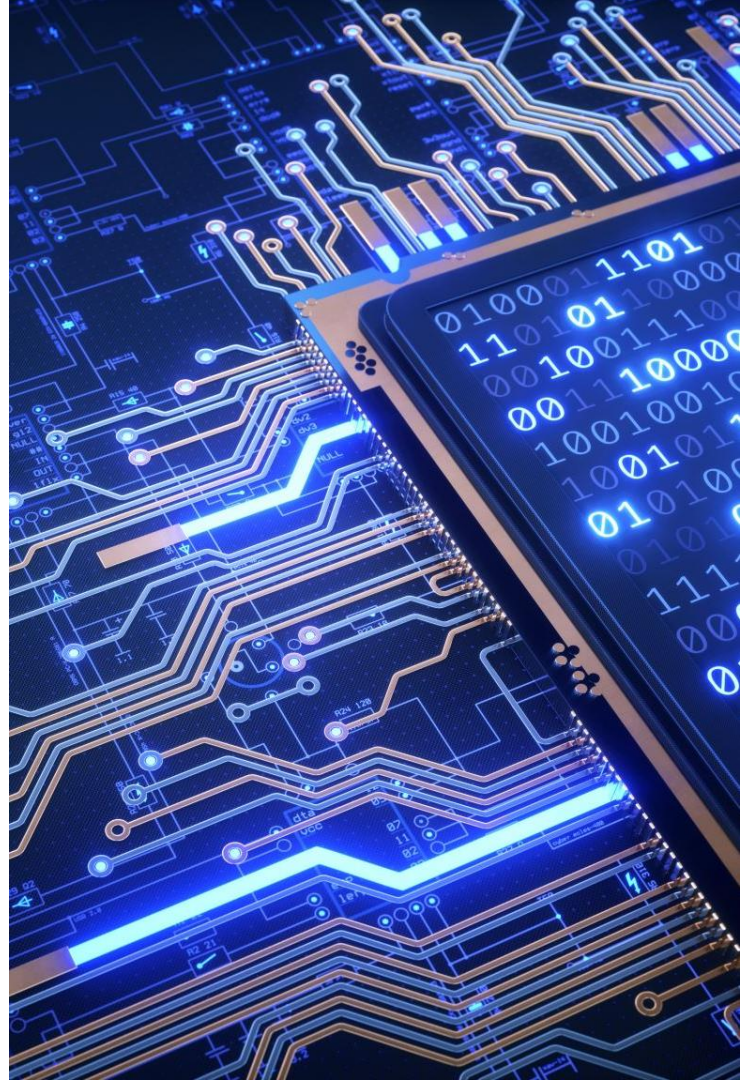
JDEME NA TO...

Kdo to myslí vážně:

- Instalace Python do počítače
↳ <https://www.python.org/>
- Instalace Jupyter Notebook
↳ <https://jupyter.org/>
- JetBrains - kompletní vývojářské prostředí
↳ <https://www.jetbrains.com/>

Python “nezávazně” (bez instalace):

- DataCamp - workspace
↳ <https://www.datacamp.com/>
- Google Drive - Jupyter Notebook
- Jupyter Notebook v prohlížeči



ZÁKLADNÍ MATEMATICKÉ OPERACE

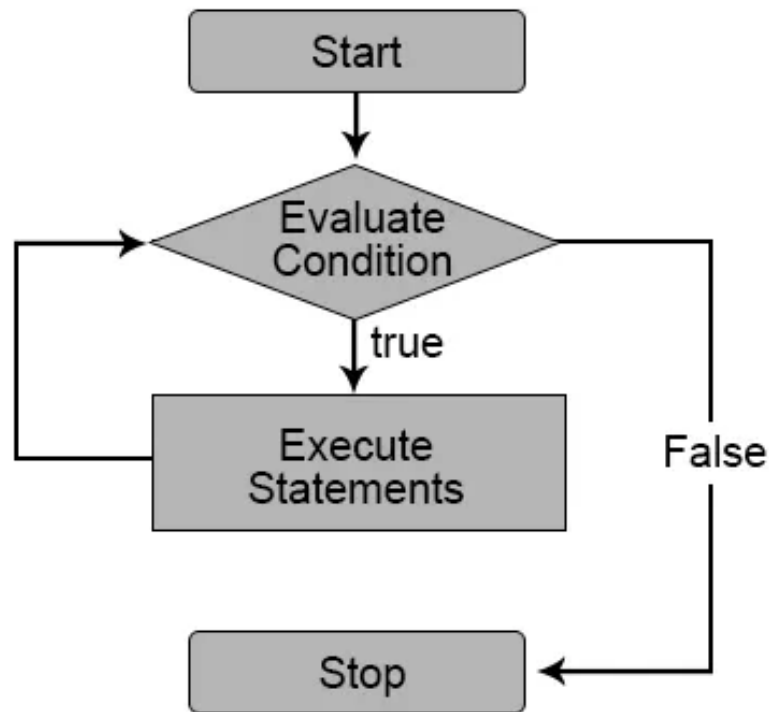
Operátor	Operace	Příklad	Výsledek
**	mocnina	2 ** 3	8
%	zbytek	22 % 8	6
//	podíl (na celá čísla)	22 // 8	2
/	podíl	22 / 8	2.75
*	součin	3 * 5	15
-	odečet	5 - 2	3
+	součet	2 + 2	4

ZÁKLADNÍ DATOVÉ TYPY

Označení v kódu	Název - AJ	Název - ČJ	Příklad
str	string	text	"hello world"
float	float number	číslo s desetinnou čárkou	3,14
int	integer	celé číslo	2
list	list	seznam	[2, 'heslo', 3,12]
dict	dictionary	slovník	{1: 'telefon'}
tuple	tuple	N-tice	(32, 11)
+	součet	2 + 2	4

SMYČKY

- WHILE / FOR
- Smyčky umožňují vytvářet celé procesní workflow a rozhodovací stromy pro provádění akcí a příkazů za předem daných podmínek.



ZÁKLADNÍ UŽITEČNÉ FUNKCE

<https://pyvec.github.io/cheatsheets/basic-functions/basic-functions-cs.pdf>

ZÁKLADNÍ ALGORITMY

0

Sudá a lichá

Zadání:

Navrhněte, jak rozpoznat, zda se jedná o sudé nebo liché číslo.




```
vek = int(input('Kolik ti je let? '))
if vek >= 150:
    print('A ze kterépak jsi planety?')
elif vek >= 18:
    # Tahle větev se např. pro "200" už neprovede.
    print('Můžeme nabídnout: víno, cider, nebo vodku.')
elif vek >= 1:
    print('Můžeme nabídnout: mléko, čaj, nebo vodu')
elif vek >= 0:
    print('Sunar už bohužel došel.')
else:
    # Nenastala ani nedna ze situací výše – muselo to být záporné
    print('Pro návštěvy z budoucnosti bohužel nemáme nic v nabídce.')
```

PRÁCE S PODMÍNKAMI (IF, ELIF, ELSE)

Pokud je splněna podmínka A, pak se spustí akce 1. Je-li splněna akce B, pak se spustí akce 2,....

```
print('Odpovídej "ano" nebo "ne".')
stastna_retezec = input('Jsi šťastná? ')
if stastna_retezec == 'ano' or stastna_retezec == 'Ano':
    stastna = True
elif stastna_retezec == 'ne' or stastna_retezec == 'Ne':
    stastna = False
else:
    print('Nerozumím!')

bohata_retezec = input('Jsi bohatá? ')
if bohata_retezec == 'ano' or bohata_retezec == 'Ano':
    bohata = True
elif bohata_retezec == 'ne' or bohata_retezec == 'Ne':
    bohata = False
else:
    print('Nerozumím!')

if bohata and stastna:
    # Je bohatá a zároveň šťastná, ta se má.
    print('Gratuluji!')
elif bohata:
    # Je bohatá, ale není „bohatá a zároveň šťastná“,
    # takže musí být jen bohatá.
    print('Zkus se víc usmívat.')
elif stastna:
    # Tady musí být jen šťastná.
    print('Zkus míň utrácet.')
else:
    # A tady víme, že není ani šťastná, ani bohatá.
    print('To je mi líto.')
```

ZÁKLADNÍ ALGORITMY

1 (DMOJ “dmopc15c7p2”)

Počítání slov

- “Uved'te počet slov náhodně zadaného textu.”

ZÁKLADNÍ ALGORITMY

2 (DMOJ “dmopc14c5p1”)

Objem kuželu

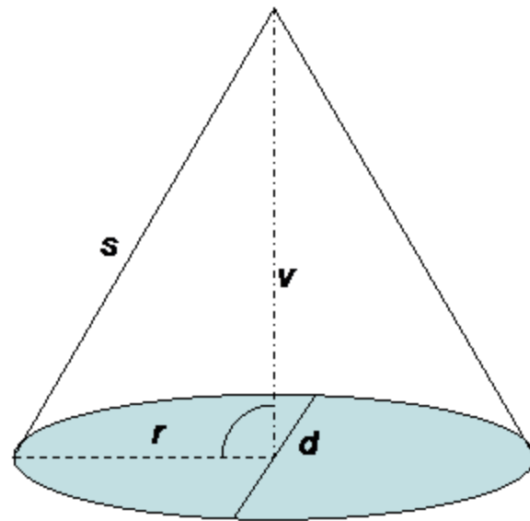
- “Vypočítejte objem rotačního kuželu.”

Vstupy

- poloměr a výška

Očekávané výstupy

- výsledek vzorce: $(\pi r^2 h)/3$



ZÁKLADNÍ ALGORITMY

3a teď to zkuste sami

Objem kuželu

- “Vypočítejte plochu a objem kvádru.”

Vstupy

- šířka, výška, délka

Očekávané výstupy

- Odpověď: “Objem kvádru o šířce ##, výšce ##

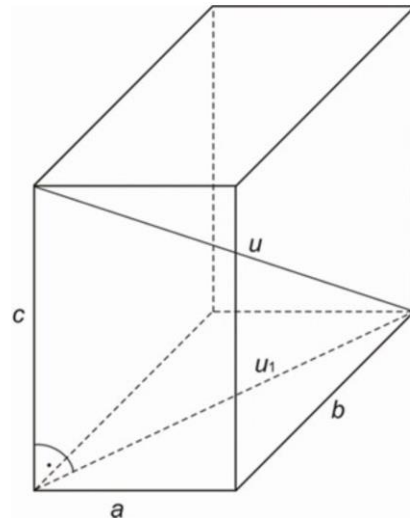
Kvádr

Objem

$$V = a \cdot b \cdot c$$

Povrch

$$S = 2(ab + ac + bc)$$



ZÁKLADNÍ ALGORITMY

4 (DMOJ “ccc06j1”)

Počet kalorií podle menu

- “Vypočítejte, kolik kalorií přijme strážník na základě výběru v menu”

Vstupy

- 4 položky jednotlivých chodů v menu
- Výběr vždy 1,2,3 nebo 4

Očekávané výstupy

- Odpověď: “Váš dnešní oběd bude obsahovat celkem ### kalorií.”

ZÁKLADNÍ ALGORITMY

Here are the three burger choices:

- 1 – Cheeseburger (461 Calories)
- 2 – Fish Burger (431 Calories)
- 3 – Veggie Burger (420 Calories)
- 4 – no burger

Here are the three drink choices:

- 1 – Soft Drink (130 Calories)
- 2 – Orange Juice (160 Calories)
- 3 – Milk (118 Calories)
- 4 – no drink

Here are the three side order choices:

- 1 – Fries (100 Calories)
- 2 – Baked Potato (57 Calories)
- 3 – Chef Salad (70 Calories)
- 4 – no side order

Here are the three dessert choices:

- 1 – Apple Pie (167 Calories)
- 2 – Sundae (266 Calories)
- 3 – Fruit Cup (75 Calories)
- 4 – no dessert

“AND NOW
FOR
SOMETHING
COMPLETELY
DIFFERENT.”

And now
for something
completely different...



“AND NOW FOR SOMETHING COMPLETELY DIFFERENT.”

5

“Guess My Word”

- Vytvořte hru pro jednoho hráče, který hádá náhodně vygenerované slovo/jméno. Viz <https://hryanjones.com/guess-my-word/>

Vstupy

- Seznam jmen (korpus) s náhodně vygenerovaným slovem
- Uživatel hádá slovo a dostává zpětnou vazbu v řazení ve slovníku dle abecedy (“Vámi zadané slovo je před/za slovem, které hledáte.”)

Očekávané výstupy

- Uživatel dostává zpětnou vazbu, kolik pokusů potřeboval na uhádnutí slova.
- Uživatel dostane zpětnou vazbu ve chvíli, kdy uhodl “tajenku”.

DALŠÍ ZDROJE

České výukové stránky =

- <https://nauce.python.cz/>

Platformy pro interaktivní výuku programování:

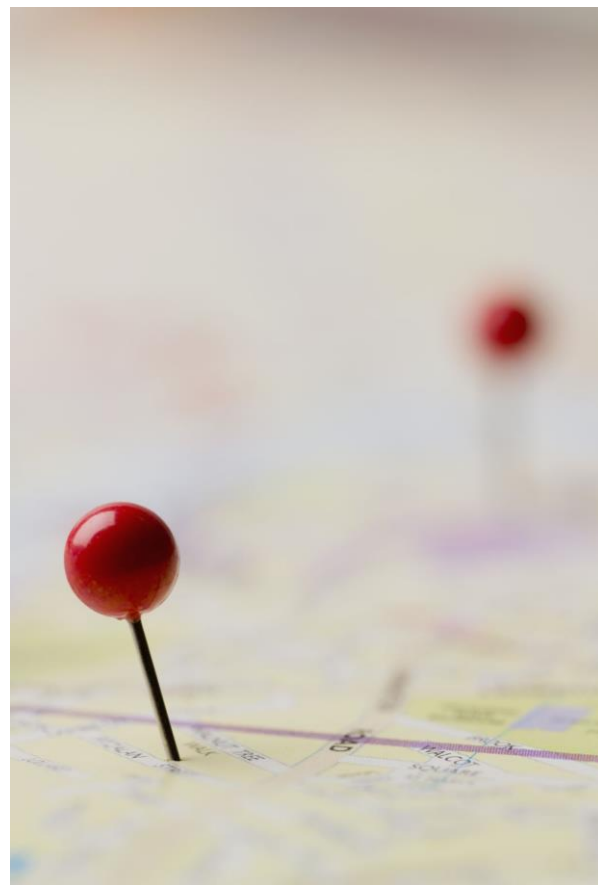
- <https://www.datacamp.com/>
- <https://www.codecademy.com/>
- <https://www.dataquest.com/>

Matematické “hlavolamy”:

- <https://projecteuler.net/>

Úvod do světa strojového učení:

- <https://www.kaggle.com/>



DALŠÍ ZDROJE — KAM PRO RADU?

Online tutoriály a reference pro programátory:

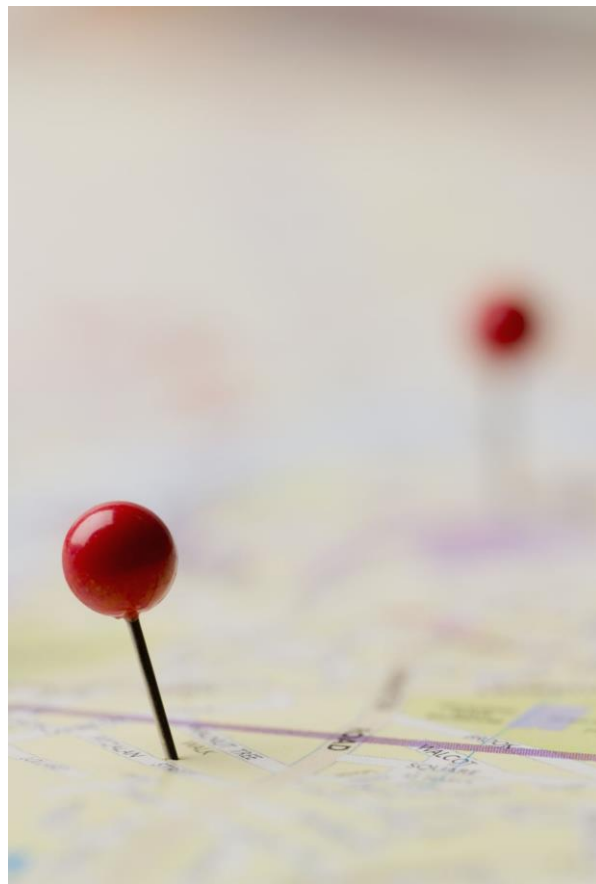
- <https://www.w3schools.com>

Sociální síť/Diskusní fórum (pro hledání konkrétních příkladů):

- <https://stackoverflow.com>

Motivace a obecnější problémy

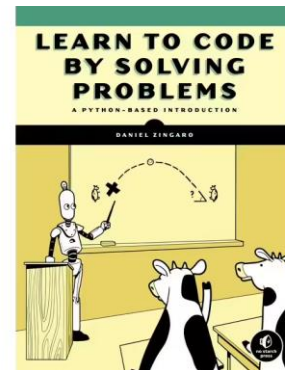
- <https://www.reddit.com>



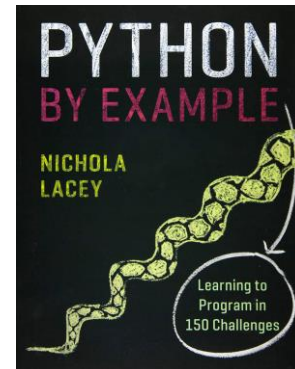
REFERENCE A POUŽITÁ LITERATURA

ZINGARO, Daniel. *Learn to code by solving problems: a Python programming primer*. San Francisco: No Starch Press, [2021]. ISBN 978-1-7185-0132-4.

PILGRIM, Mark. *Ponořme se do Python(u) 3 = Dive into Python 3*. Praha: Cz.Nic, c2010. CZ.NIC. ISBN 978-80-904248-2-1.



LACEY, Nichola. *Python by example: learning to program in 150 challenges*. Cambridge, United Kingdom: Cambridge University Press, 2019. ISBN 978-1-108-71683-3.



SAMOSTUDIUM

DataCamp

Komerční eLearningová platforma pro interaktivní výuku především v oblasti datové analytiky a strojového učení

Podporované technologie:

 python™

 R

 SQL

 Power BI

ORACLE®


 X

 Scala

 tableau

 SPREADSHEETS

 spark

 Shell

 git

DATA CAMP

- Kdo z DataCampu může těžit nejvíce:
 - ↩ Studenti a vědečtí pracovníci, kteří potřebují zkoumat data
 - ↩ Lidé, kteří hledají cestu pro rekvalifikaci a vstup do nového odvětví pracovního trhu
 - ↩ Nadšenci do dat
 - ↩ Zájemci o programovací jazyky Python, R či SQL



Line plot

DATA CAMP

With matplotlib, you can create a bunch of different plots in Python. The most basic plot is the line plot. A general recipe is given here.

```
import matplotlib.pyplot as plt
plt.plot(x,y)
plt.show()
```

Výhody pro uživatele:



Obsah pro všechna témata a úroveň znalostí v oblasti data science



Lze využívat bez nutnosti instalace dalšího software



Praktická cvičení na reálných případových studiích



Intuitivní rozhraní a volitelnost látky, kterou potřebujete probrat.

This course touches on a lot of concepts you may have forgotten, so if you ever need a quick refresher, download the [Python for data science Cheat Sheet](#) and keep it handy!

- `print()` the last item from both the `year` and the `pop` list to see what the predicted population for the year 2100 is. Use two `print()` functions.



script.py

```
1 # Print the last item from year and pop
2 print(year[-1])
3 print(pop[-1])
4
5
6 # Import matplotlib.pyplot as plt
7 import matplotlib.pyplot as plt
8
9 # Make a Line plot: year on the x-axis, pop on the y-axis
10 plt.plot(year, pop)
11
12 # Display the plot with plt.show()
13 plt.show()
```



Run C

IPython Shell

Slides

In [1]:

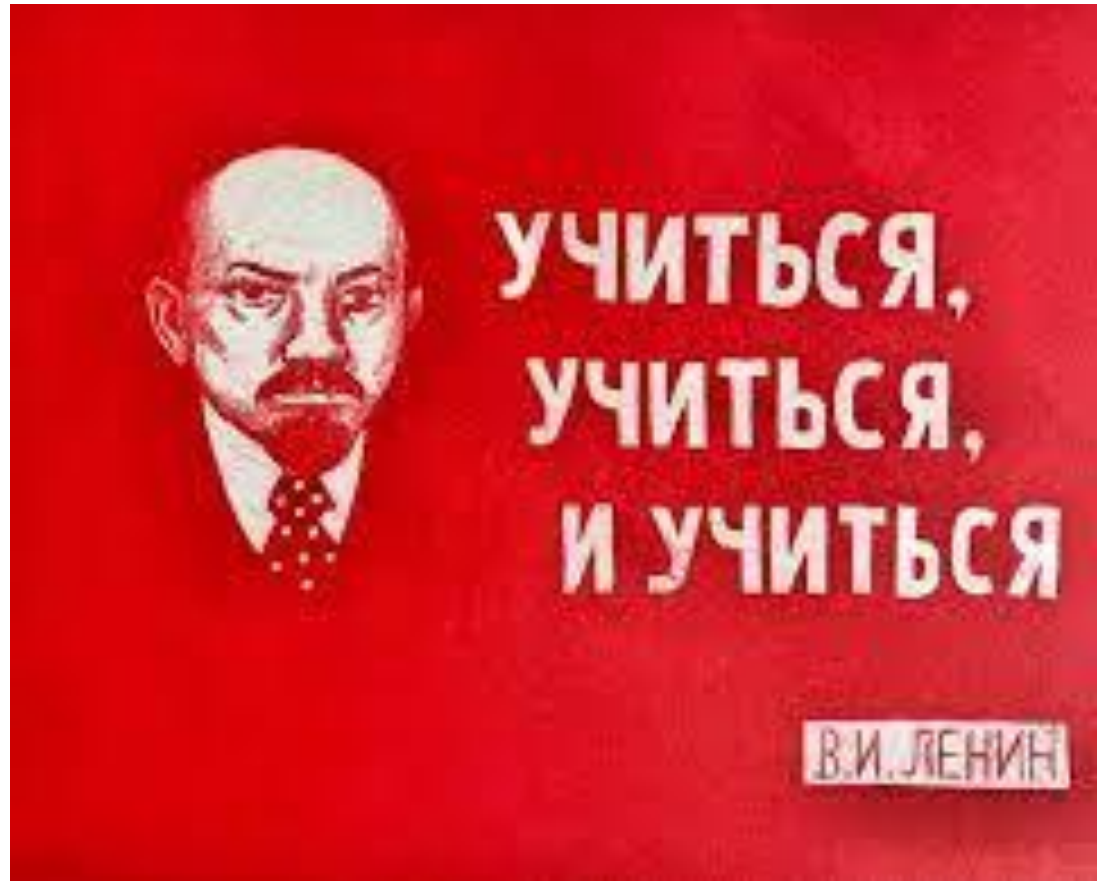


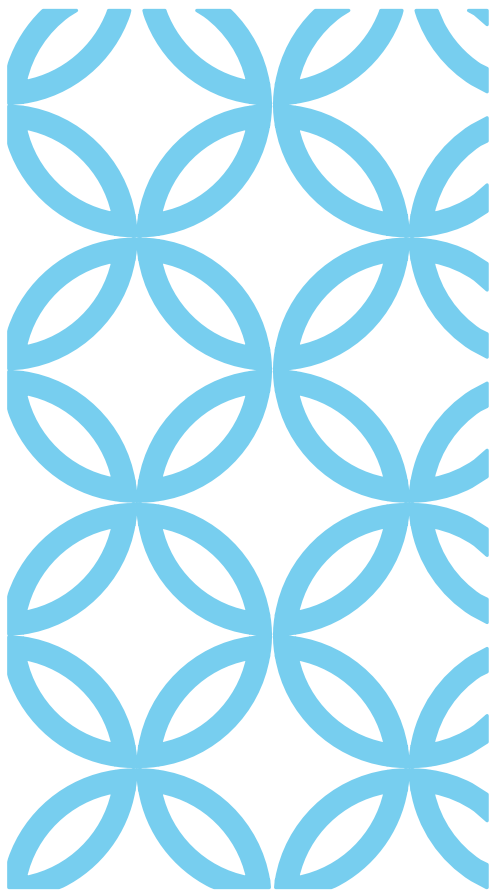
A CO DÁL?

Nebojte se opisovat!

A CO DÁL?

Učit se...





Pavel.Synek@techlib.cz
<https://www.linkedin.com/in/psynek/>

DĚKUJI ZA POZORNOST!