

MUNI
SCI

Užitečné balíčky

Kryštof Mrózek (445429@mail.muni.cz)

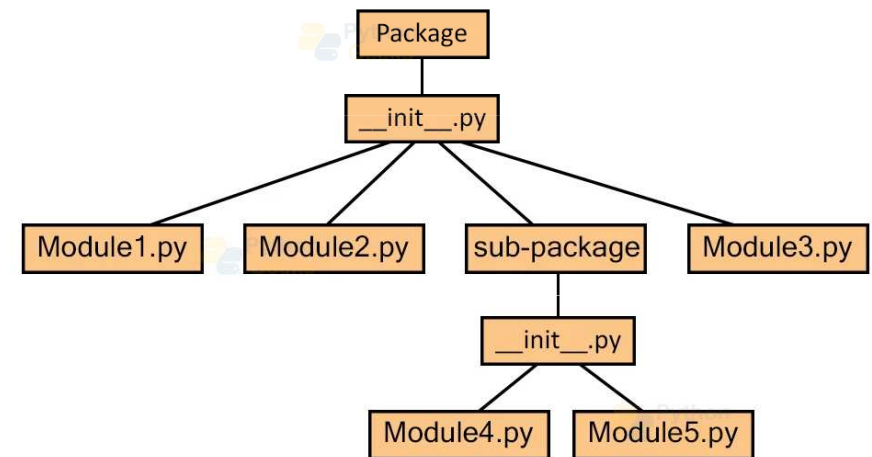
Kristína Tomanková (kristinatomankova@mail.muni.cz)

Radoslav Brunovský (rbrunovsky@mail.muni.cz)

Python balíček

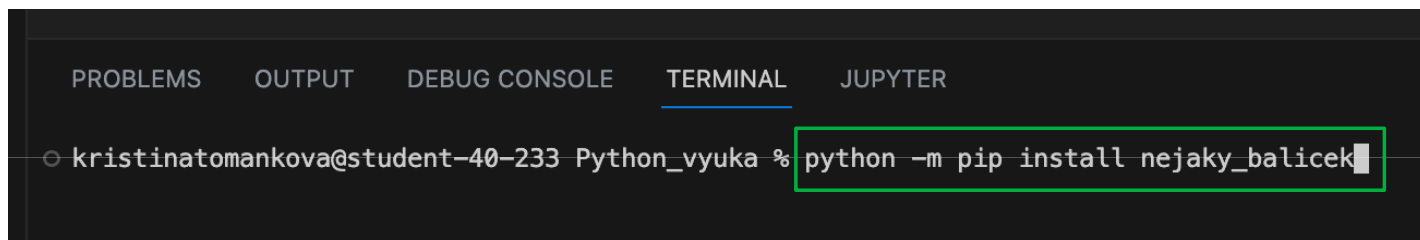
- Python balíček je štruktúra, ktorá organizuje / zoskupuje súbory s python kódom (moduly) do jednej zložky.
- Balíček si môžeme predstaviť ako zložku, ktorá obsahuje skripty + súbor “__init__.py” = tento súbor definuje zložku ako python balíček.
- Balíčky obsahujú mnoho užitočných funkcií, ktoré už boli naprogramované.

Structure of Packages



Inštalovanie balíčka

- Aby sme mohli využívať balíčky, tak je potrebné ich najprv nainštalovať.
- Inštalujeme cez terminál (vstavaný alebo vo VSCode):
 1. Vo VSCode otvorte terminal (buď ho už máte otvorený alebo View > Terminal alebo Ctrl/Cmd + J).
 2. Do terminálu píšeme štruktúru:
 - Windows: `pip install <balicek>`
 - MacOS: `python -m pip install <balicek>`



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER
○ kristinatomankova@student-40-233 Python_vyuka % python -m pip install nejaky_balicek
```

Pozn. Virtuálne prostredie

- Zodpovedný programátor by si mal v prípade pracovania v pythone vytvoriť tzv. virtuálne prostredie.
- Ide o izolované prostredie, ktoré obsahuje inštaláciu pythonu (v akejkol'vek verzii) a ďalšie balíčky.

Q: Prečo použiť virtuálne prostredie?

A: Kontrola závislostí (rôzne projekty môžu vyžadovať rôzne verzie pythonu), takmer bez rizika (ak si náhodou poškodím samotný python, tak nejde o jeho systémovú verziu), kompatibilita s ďalšími užívateľmi (všetci programátori pracujú v rovnakom prostredí s rovnakými balíčkami).

V prípade záujmu odporúčam využiť **pyenv**:

<https://github.com/pyenv/pyenv>

<https://github.com/pyenv/pyenv-virtualenv>

V našom predmete si ukážeme použitie len najzákladnejších balíčkov, ktoré môžeme nainštalovať do systémového pythonu bez rizika.

Základné balíčky

- Nainštalujte si nasledujúce balíčky:
 - numpy
 - scipy
 - pandas
 - matplotlib

Terminal:

Windows: `pip install <balicek>`

MacOS: `python -m pip install <balicek>`

Použitie balíčka

- Aby sme mohli využiť funkcie z nainštalovaného balíčka, musíme ho nahrat' (importovať) do nášho kódu.

- Nahratie celého balíčka

```
import numpy
```

- Nahratie celého balíčka so skratkou

```
import numpy as np
```

- Nahratie špecifickej funkcie z balíčka

```
from numpy import sinh
```

Použitie balíčka

- Funkcie následne voláme odkazujúc sa na balíček, v ktorom sa nachádzajú:

```
import numpy  
  
import numpy as np  
  
from numpy import sinh
```



```
numpy.sinh(x)  
  
np.sinh(x)  
  
sinh(x)
```

NumPy = NumericalPython (alias np)

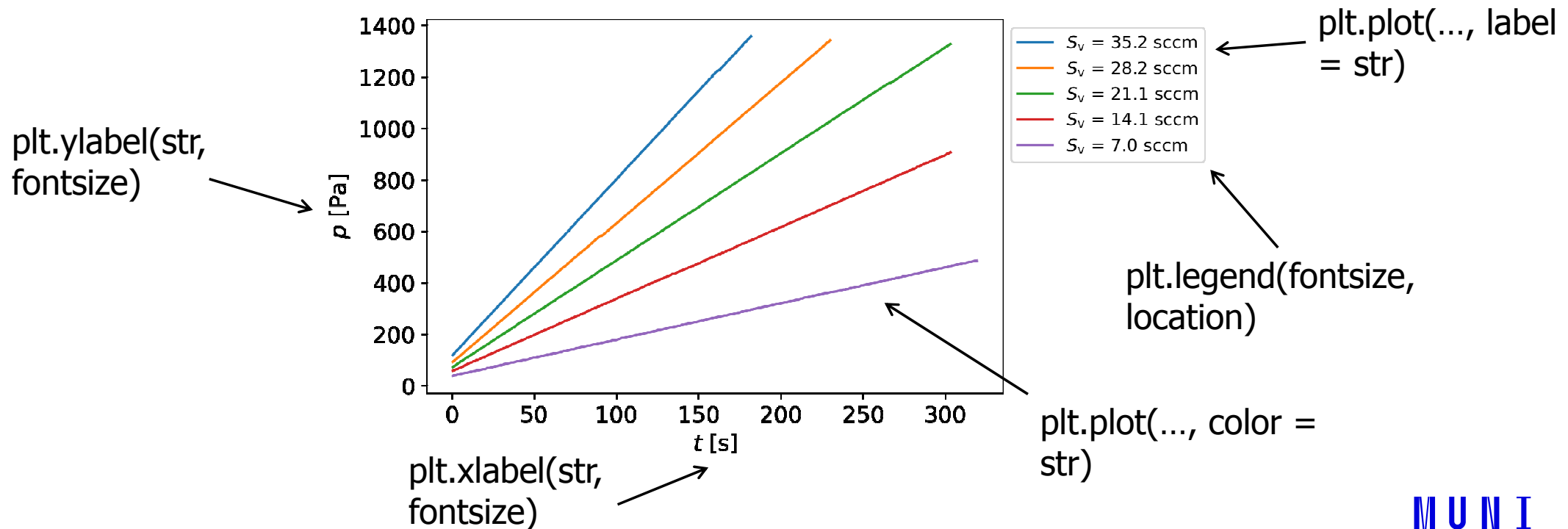
- Jeden z najužitočnejších balíčkov v pythone, používa sa na numerické výpočty.
- Umožňuje ukladanie dát do viacrozmerných polí, matíc a poskytuje množstvo komplexných matematických funkcií a operácií.
- Python list vs numpy pole:
 - Numpy pole sú homogénne (dáta jedného typu) – jednoduchšia manipulácia
 - Numpy pole sú lepšie optimalizované – zaberajú menej pamäte a operácie s nimi sú rádovo rýchlejšie
- Polia najčastejšie reprezentujú vektory (1D) alebo matice/tenzory (2+D)
- Najčastejšie využívané funkcie:
 - `np.array()`, `np.zeros()`, `np.ones()`, `np.empty()`, `np.arange()`, `np.linspace()`, `np.logspace()`, `np.random()`, `np.sum()`, `np.mean()`, `np.std()`, `np.dot()` / `@`, `np.min()`, `np.max()`, `np.sin()` ...
 - `dtype` – voliteľný parametr špecifikujúci typ dát (`int`, `float`...)

SciPy = ScientificPython

- Nadstavba numpy so zabudovanými funkciami pre dátovú analýzu
- Najčastejšie knižnice:
 - `scipy.constants` – obsahuje fyzikálne a matematické konštanty
 - `scipy.optimize` – obsahuje metódy riešenia rovníc (minimalizácia, korene)
 - `scipy.integrate` – obsahuje metódy riešenia integrálov a diferenciálnych rovníc
- Najčastejšie používané funkcie:
 - `scipy.optimize.curve_fit()` – fitovanie dát (ukážka v príklade)
 - `scipy.optimize.root()` – nájde koreň lineárnej rovnice
 - Napríklad rovnica $y = x + \sin(x)$: `scipy.optimize.root(y, odhad koreňa)`
 - `scipy.integrate.odeint()` – riešenie obyčajných diferenciálnych rovníc

Matplotlib.pyplot (alias plt)

- Balíček na vykresľovanie grafov (prípadne obrázkov) – `plt.plot(dataX, dataY)`



Pandas (alias pd)

- Balíček na načítanie a analýzu dát – vhodný pre spracovávanie vašich meraní
- Dve základné dátové štruktúry:
 - `pd.Series` = stĺpec
 - `pd.DataFrame` = tabuľka
- Načítanie dát: možnosť z `.csv`, `.txt`, `.xlsx`, JSON (formát totožný s python slovníkom), atď.
 - Rovnako môžeme dáta exportovať do niektorého z formátov
- Nutnosť hlavičky napr. dĺžka, šírka, výška
 - Prístup k dátam: `df = pd.read_csv("dimenzie.csv")`
 - Šírka: `df['šírka']` (podobne ako pri práci so slovníkmi)

Spoločné cvičenie

– Meranie tiažového zrýchlenia Zeme (FP1 - úloha 4):

- Predstavme si, že meriate periódu kmitov kyvadla pre rôzne dĺžky závesu L .

$$T = 2\pi\sqrt{L/g}$$

- Pomocou stopiek zmeriate vždy 10 periód pre dané L a toto meranie zopakujete 5x.
- Z týchto meraní spočítajte tiažové zrýchlenie Zeme.

