

NOFY080 poznámky k přednášce

Emil Varga

18. září 2025

Obsah

1	Nastavení prostředí	2
1.1	Spouštění kódu v Pythonu	2
1.2	Krátká poznámka k textovým editorům	3
1.2.1	Jupyter Notebooky	4

Python Jazykové Intermezzy

Code Snippets

1 Nastavení prostředí

Běžné programování v Pythonu silně spoléhá na externí balíčky, které mohou mít různé verze a závislosti. Aby se předešlo konfliktům mezi různými projekty, doporučuje se pro každý projekt používat virtuální prostředí (*virtual environment*, alebo v skratce *venv*). V tomto kurzu budeme takové virtuální prostředí používat, abychom zajistili konzistentní vývojové prostředí pro všechny. Virtuální prostředí jsou doporučeným způsobem správy závislostí v projektech v Pythonu.

Pro správu těchto virtuálních prostředí budeme používat open-source nástroj *uv*, dostupný na <https://github.com/astral-sh/uv>. Pokud již máte nainstalovanou jakoukoli verzi Pythonu, můžete *uv* nainstalovat jednoduchým spuštěním

```
1 pip install uv
```

Pokud Python nainstalovaný nemáte, můžete si jej stáhnout z <https://www.python.org/downloads/> a poté použít výše uvedený příkaz nebo použít samostatné instalátory na <https://github.com/astral-sh/uv>.

Pro otestování instalace spusťte v příkazovém řádku

```
1 uv --version
```

a mělo by se objevit něco podobného jako *uv 0.5.30*.

Nyní vytvoříme projekt s názvem *NOFY080_2025*, který bude obsahovat všechny soubory související s tímto kurzem, spuštěním

```
1 uv init NOFY080_2025
```

kteřý vytvoří nový adresář se stejným názvem, s jednoduchým souborem "Hello World" v Pythonu, několika dalšími soubory, které *uv* používá ke sledování závislostí vašeho projektu, a nastaví v něm git repozitář (v tomto kurzu nemusíte používat git, ale zájemci se mohou naučit základy v dodatku ??).

Nyní se přesuňte do adresáře projektu

```
1 cd NOFY080_2025
```

a přidejte balíčky, které budeme v průběhu tohoto kurzu potřebovat

```
1 uv add numpy scipy matplotlib
```

což vytvoří virtuální prostředí a nainstaluje zadané balíčky (v adresáři projektu by se měl objevit adresář *.venv*).

Pokud byste chtěli vytvořit virtuální prostředí bez instalace jakýchkoli balíčků, mohli byste použít příkaz

```
1 uv venv <name of the virtual environment>
```

Pro aktivaci virtuálního prostředí spusťte následující příkaz v Linuxu nebo Mac OS

```
1 source .venv/bin/activate
```

a ve Windows

```
1 .venv\Scripts\activate
```

Nyní, když spustíte *python*, spustí se verze z virtuálního prostředí, nikoli vaše systémová verze. Pro deaktivaci virtuálního prostředí jednoduše spusťte *deactivate*.

1.1 Spouštění kódu v Pythonu

Python je interpretovaný jazyk, který není třeba před spuštěním kompilovat. Kód v Pythonu ukládáme do textových souborů s příponou *.py* nebo do takzvaných Jupyter notebooků (přípona souboru *.ipynb*).

Pro otestování naší instalace vytvořte v našem projektovém adresáři nový soubor Pythonu (textový soubor s příponou *.py*) *test.py* s následujícím obsahem

```

1 import matplotlib.pyplot as plt
2
3 plt.plot([1, 2, 3], [4, 5, 6], '-o')
4 plt.show()

```

Pro jeho spuštění můžete buď manuálně aktivovat virtuální prostředí a spustit jej pomocí `python test.py` nebo použít `uv run test.py`. Měl by se objevit jednoduchý graf. Ve zbytku těchto skriptů, kdykoli vyvoláme příkaz `python` z příkazového řádku, myslíme tím ten ve virtuálním prostředí.

Pro spuštění kódu v Pythonu přímo bez uložení do souboru otevřete vhodný terminál a zadejte `python` a objeví se výzva (*prompt*) jako

```

1 >>>

```

Toto je Read-Evaluate-Print-Loop (REPL); jakýkoli zadaný kód bude spuštěn a výsledek vytištěn na obrazovku. Pro ukončení zadejte `quit()`. Pro spuštění souboru `myfile.py` se nejprve přesuňte do jeho adresáře a poté jej jednoduše spusťte pomocí `python myfile.py` (všimněte si, že pokud máte aktivované virtuální prostředí, skutečný skript Pythonu se nemusí nacházet ve vašem projektovém adresáři).

Cvičení 1. Vytvořte textový soubor `hello.py`, který obsahuje jeden řádek

```

1 print("Hello world!")
2

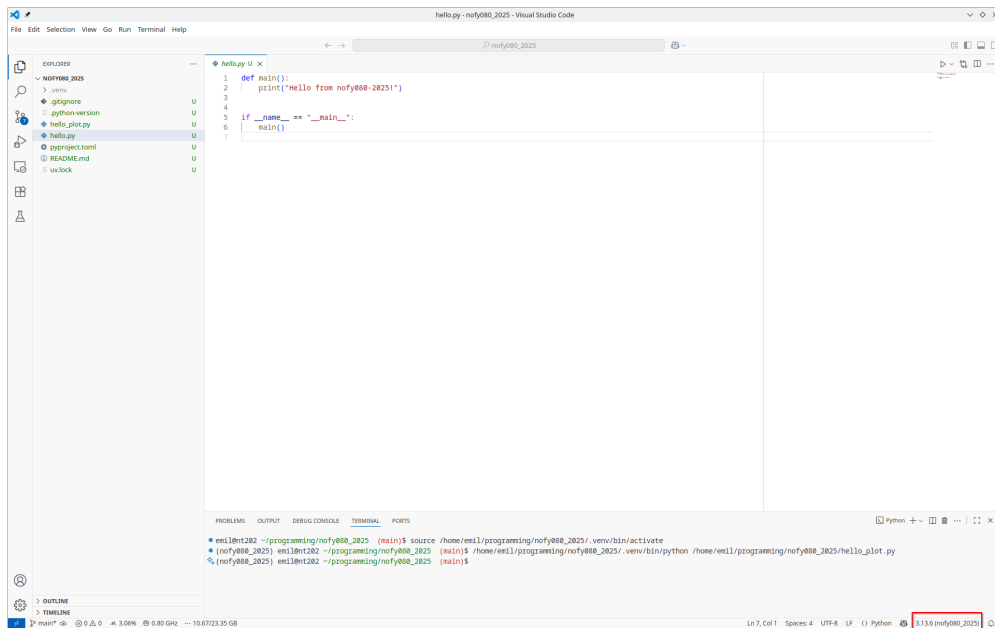
```

a poté tento soubor spusťte.

Někdy nechceme, aby se interpreter Pythonu ukončil ihned po dokončení běhu našeho programu (např. chceme zkontrolovat proměnné vytvořené během běhu programu); to lze provést pomocí `python -i file.py`, kde `-i` znamená *interaktivní*. Alternativně lze uživatelsky přívětivější (barevně odlišená syntaxe, automatické doplňování atd.) verzi interaktivního Python REPL vyvolat pomocí `ipython` (nebo `ipython3`, v závislosti na vaší instalaci), který lze také použít k interaktivnímu spouštění souborů pomocí `ipython3 -i myfile.py`.

1.2 Krátká poznámka k textovým editorům

K psaní kódu v Pythonu lze použít jakýkoli textový editor, včetně výchozího Poznámkového bloku ve Windows. Pro vaše duševní zdraví je však přínosné používat alespoň něco se zvýrazňováním syntaxe, jako je Notepad++, nebo editor s více funkcemi, jako je VS Code nebo Spyder, kde můžete soubor spustit bez přepínání do terminálu. V tomto kurzu doporučuji používat VS Code s nainstalovaným rozšířením pro Python, protože usnadňuje práci s virtuálními prostředími. Otevření adresáře ve VS Code automaticky detekuje virtuální prostředí a použije ho při spuštění souboru. Název aktuálně používaného virtuálního prostředí je uveden v pravém dolním rohu (zvýrazněno červeně):



Práce s prostředími ve Spyderu je poněkud složitější; viz průvodce zde.

1.2.1 Jupyter Notebooky

Jupyter notebooky (spouštěné pomocí `jupyter-notebook` v příkazovém řádku) poskytují snadno použitelné interaktivní prostředí s rozhraním běžícím ve webovém prohlížeči. Jsou dobré pro zkoušení věcí¹. Jupyter notebooky také dobře fungují s virtuálními prostředími a VS Code. Přejděte do adresáře svého projektu a spusťte

```
1 uv pip install jupyter ipyml
```

Balíček `ipyml` umožňuje použití interaktivních grafů Matplotlib v Jupyter noteboocích. Dále si také nainstalujte rozšíření `jupyter` ve VS Code. Nyní jednoduše vytvořte soubor s příponou `.ipynb` a můžete jej používat ve VS Code stejně jako v prohlížeči. Při prvním spuštění budete požádáni o výběr virtuálního prostředí, ve kterém by měl notebook běžet; to, které jsme vytvořili, by mělo být jednou z možností, pokud je notebook uložen v projektovém adresáři (stejný adresář jako adresář `.venv`).

¹V tomto kurzu budeme (většinou) používat skripty kvůli snazšímu rozdělení do modulů a menšímu počtu problémů s paralelním programováním, se kterými se setkáme později.

