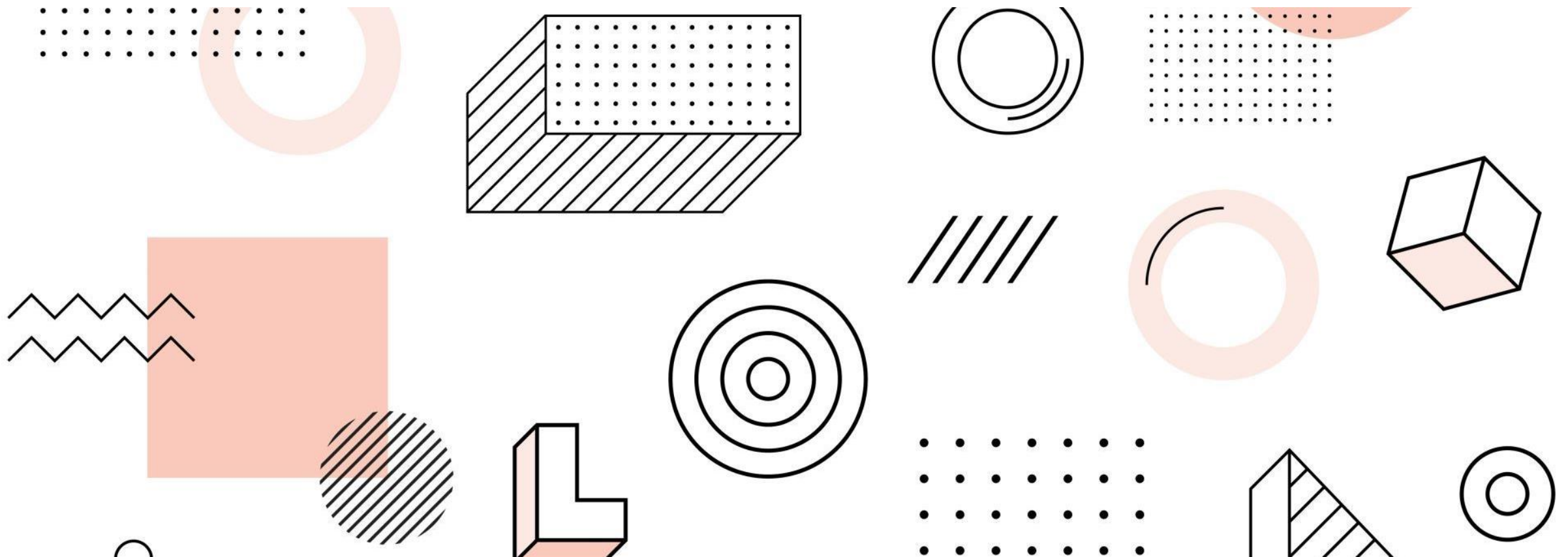


# PYTHON PRO INŽENÝRSKÉ VÝPOČTY

Týden 10. Programovací kultura. Git



# Tak začneme!

Protože už umíme psát delší kód v Pythonu, je na čase promluvit o programovacím stylu a dokumentaci:

- Rozebereme **PEP 8** - stylovou příručku, kterou se řídí většina projektů;
- Podíváme se na doporučené použití komentářů a dokumentačních řetězců (**docstrings**);
- Naučíme se spravovat verze Pythonu a nainstalovaných balíčků ve virtuálních prostředích pomocí **venv**, **pip** nebo **conda**;
- Naučíme se taky spravovat vývojové verze kódu pomocí **Git**.

## PEP 8 – HLAVNÍ BODY

- Používejte odsazení 4 mezery a žádné tabulátory.
- Řádky obalujte tak, aby nepřesáhly 79 znaků.
- Pro oddělení funkcí a tříd a větších bloků kódu uvnitř funkcí používejte prázdné řádky.
- Pokud je to možné, umístěte komentáře na samostatný řádek.
- Používejte docstrings.
- Používejte mezery kolem operátorů a za čárkami, ale ne přímo uvnitř konstrukcí se závorkami.
- Třídy a funkce pojmenovávejte důsledně; konvencí je používat pro třídy velká písmena (UpperCamelCase) a pro funkce a metody malá písmena (Lowercase\_with\_underscores).
- Používejte jenom kódování jazyka Python, UTF-8 nebo ASCII.
- Nepoužívejte znaky jiné než ASCII v identifikátorech.

# DOKUŘETĚZCE

Dokumentační řetězec (**docstring**) je řetězec, který se vyskytuje jako první příkaz v definici modulu, funkce, třídy nebo metody. Takový řetězec doc se stává speciálním atributem `__doc__` daného objektu.

```
def add(x, y):  
    """Add both arguments and returns their sum."""  
    return x + y  
  
print(add.__doc__)
```

## VIRTUÁLNÍ PROSTŘEDÍ

Aplikace Pythonu často používají balíčky a moduly, které nejsou součástí standardní knihovny. Někdy budou aplikace potřebovat konkrétní verzi knihovny. To znamená, že nemusí být možné, aby jedna instalace Pythonu splňovala požadavky každé aplikace. Pokud aplikace A potřebuje verzi 1.0 určitého modulu, ale aplikace B potřebuje verzi 2.0, pak jsou požadavky v konfliktu a instalace verze 1.0 nebo 2.0 způsobí, že jedna aplikace nebude moci být spuštěna.

Řešením tohoto problému je vytvoření **virtuálního prostředí**, samostatného stromu adresářů, který obsahuje instalaci Pythonu pro určitou verzi Pythonu a řadu dalších balíčků. Různé aplikace pak mohou používat různá virtuální prostředí.

## VENV A PIP

Pomocí balíků **venv** a **pip** můžete vytvářet nová virtuální prostředí a instalovat/upgradovat/odinstalovat balíky.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Me>cd C:\Users\Me\my\env\folder
C:\Users\Me\my\env\folder>python -m venv myenv
C:\Users\Me\my\env\folder>myenv\Scripts\activate
(myenv) C:\Users\Me\my\env\folder>python -m pip
install nova
Collecting nova
  Downloading nova-27.1.0-py3-none-any.whl (5.8
  MB)
  -----
  5.8/5.8 MB 5.8 MB/s eta 0:00:00
(myenv) C:\Users\Me\my\env\folder>deactivate
```

## VENV A PIP

Pomocí balíků **venv** a **pip** můžete vytvářet nová virtuální prostředí a instalovat/upgradovat/odinstalovat balíky.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Me>cd C:\Users\Me\my\env\folder
C:\Users\Me\my\env\folder>python -m venv myenv
C:\Users\Me\my\env\folder>myenv\Scripts\activate
(myenv) C:\Users\Me\my\env\folder>python -m pip
install nova
Collecting nova
  Downloading nova-27.1.0-py3-none-any.whl (5.8
  MB)
  -----
  5.8/5.8 MB 5.8 MB/s eta 0:00:00
(myenv) C:\Users\Me\my\env\folder>deactivate
```

# CONDA

Pokud jste nainstalovali Python prostřednictvím systému pro správu balíčků **Conda**, můžete jeho funkce použít místo venv a pip.

```
C:\> Command Prompt
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Me>conda create --name myenv
C:\Users\Me>conda activate myenv
C:\Users\Me>conda install novas
(myenv) C:\Users\Me>conda install novas
Collecting nova
  Downloading nova-27.1.0-py3-none-any.whl (5.8 MB)
-----
5.8/5.8 MB 5.8 MB/s eta 0:00:00
(myenv) C:\Users\Me\my\env\folder>conda deactivate
```



## GIT

Git je distribuovaný systém pro správu verzí, který umožňuje sledovat změny v souborech a spolupracovat s ostatními na projektech.

Zvažte použití systému Git v kombinaci se službou cloudových úložišť, jako je GitHub nebo GitLab, pro plynulou spolupráci a zálohování kódu.

