Python no terminal

Alguns IDEs (Integrated Development Environment)

- Kate (leve e multiplataforma)
- PyCharm (Community Edition é free e cheio de recursos)
- Visual Studio Code (multiplataforma, git embutido)
- Ou ambientes mais sofisticados e pesados se estiver trabalhando num projeto com múltiplos scripts e pacotes (Ex: PyDev for Eclipse)

Aqui iremos usar vim, gvim ou gedit (ou qualquer outro editor de textos)

Dicas pra quem quiser usar o editor vim ou gvim para editar códigos Python

- Pacote vim-X11 (use o instalador de pacotes da sua distribuição)
- Edite o arquivo chamado .vimrc do seu homedir e adicione as seguintes linhas no final
- Se você for o administrador (root), pode fazer a configuração para todos os usuários editando o arquivo /etc/vimrc

```
syntax on
filetype indent plugin on
set modeline
:set tabstop=8 expandtab shiftwidth=4 softtabstop=4
```

Que versão do python?

- `python --version`
- `python3 --version`
- Comando: `which python` (retorna o caminho do interpretador python que está sendo usado por default quando vc digita python no terminal)
- `which python3`
- Podemos mudar a variável de ambiente `PATH` para encontrar primeiro um ou outro interpretador python.

Como rodar um script python no terminal

- crie um diretório para colocar seu(s) script(s)
  - mkdir aula5
- cd aula5
- Usando o editor de texto de sua preferência crie um script chamado programa1.py
- A primeira linha do script deve ter o “shebang” com o caminho do interpretador python.
- #!/usr/bin/env python3

Dê permissão de execução e rode seu programa

- No diretório do programa1.py digite:
  - chmod +x programa1.py
- Agora rode com: ./programa1.py
- Alternativamente: python3 programa1.py
- Ou se seu interpretador python padrão já for da versão 3, simplesmente: python programa1.py

Como saber quantos parâmetros foram passados?

- len(sys.argv)
- Boa prática: testar se seu programa recebeu o número de argumentos correto

Virtual environment

- É útil colocar seu programa junto com todos os módulos que ele usa num ambiente virtual
- No ambiente virtual é possível instalar pacotes que só serão visíveis dentro do ambiente virtual, sem influenciar o resto do sistema
- Você pode ter quantos ambientes virtuais quiser com cada um instalado em um diretório diferente.

Como criar um ambiente virtual?

- Entre no diretório onde você quer criar o ambiente virtual
- virtualenv -p python3 NOME_DO_AMBIENTE_VIRTUAL

- Ex: virtualenv -p python3 venv
- A linha acima irá criar um subdiretório venv contendo o interpretador python, a biblioteca padrão e tudo mais pro python funcionar

Ativando o ambiente virtual

source venv/bin/activate

pip install numpy

pip install matplotlib

...

python meu_programa.py (ou ./programa.py)

Teste do numpy

- #!/usr/bin/env python3
- import numpy as np
- X = np.array([[1,2],[3,4]])
- Y = np.array([[2,3],[5,8]])
- e = np.dot(X,Y)
- print(e)

Desativando o ambiente virtual

- Comando: deactivate
- Você pode criar vários ambientes virtuais, um para cada programa, e instalar os módulos que cada um usa com o pip (instalador de pacotes python)
- Python Package Index https://pypi.org/
- Trabalhando desta forma, você poderá usar o python numa máquina compartilhada sem mexer nas bibliotecas do sistema, evitar conflitos de versões de pacotes etc.