

Lead Mentoring - Aula 02 – Versionamento de Ambiente

Nome: Jonnas Christian

Mentor: Gustavo Soares

1. Diretório

Antes de iniciar o versionamento tem que estar no diretório onde você vai salvar o seu projeto. Pra isso se usa o comando “cd” no terminal do Linux.

```
jonnas@JonnasCSP:~$ cd Documentos/'Lead Mentoring'  
jonnas@JonnasCSP:~/Documentos/Lead Mentoring$
```

2. Pyenv

O Pyenv é uma ferramenta que te permite selecionar qual versão do Python está sendo utilizada.

Tendo instalado já no seu computador, é preciso ser digitado o comando “pyenv” no terminal. Uma janela como essa é esperada:

```
jonnas@JonnasCSP:~/Documentos/Lead Mentoring$ pyenv  
pyenv 2.2.0  
Usage: pyenv <command> [<args>]  
  
Some useful pyenv commands are:  
  activate      Activate virtual environment  
  commands      List all available pyenv commands  
  deactivate     Deactivate virtual environment  
  doctor         Verify pyenv installation and development tools to build pythons.  
  exec          Run an executable with the selected Python version  
  global         Set or show the global Python version(s)  
  help          Display help for a command  
  hooks         List hook scripts for a given pyenv command  
  init          Configure the shell environment for pyenv  
  install        Install a Python version using python-build  
  local         Set or show the local application-specific Python version(s)  
  prefix        Display prefix for a Python version  
  rehash        Rehash pyenv shims (run this after installing executables)  
  root          Display the root directory where versions and shims are kept  
  shell         Set or show the shell-specific Python version  
  shims         List existing pyenv shims  
  uninstall     Uninstall a specific Python version  
  --version     Display the version of pyenv  
  version       Show the current Python version(s) and its origin  
  version-file  Detect the file that sets the current pyenv version  
  version-name  Show the current Python version  
  version-origin Explain how the current Python version is set  
  versions     List all Python versions available to pyenv  
  virtualenv   Create a Python virtualenv using the pyenv-virtualenv plugin  
  virtualenv-delete Uninstall a specific Python virtualenv  
  virtualenv-init Configure the shell environment for pyenv-virtualenv  
  virtualenv-prefix Display real_prefix for a Python virtualenv version  
  virtualenvs  List all Python virtualenvs found in '$PYENV_ROOT/versions/*'.  
  whence      List all Python versions that contain the given executable  
  which       Display the full path to an executable  
  
See 'pyenv help <command>' for information on a specific command.  
For full documentation, see: https://github.com/pyenv/pyenv#readme
```

Dentre todas as funcionalidades do pyenv, você pode escolher entre instalar uma versão específica do Python ou já usar uma instalada no seu computador. Nesse caso, irei instalar uma versão em específico. Para isso, se utiliza da funcionalidade “install” para instalar e “-l” para listar todas as versões disponíveis para serem instaladas.

```
jonnas@JonnasCSP:~/Documentos/Lead Mentoring$ pyenv install -l
```

Ao escolher a versão que gostaria de utilizar no seu projeto, é hora de instalá-la. Desse modo, repita o processo anterior, posicionando a versão escolhida no lugar do “-l”:

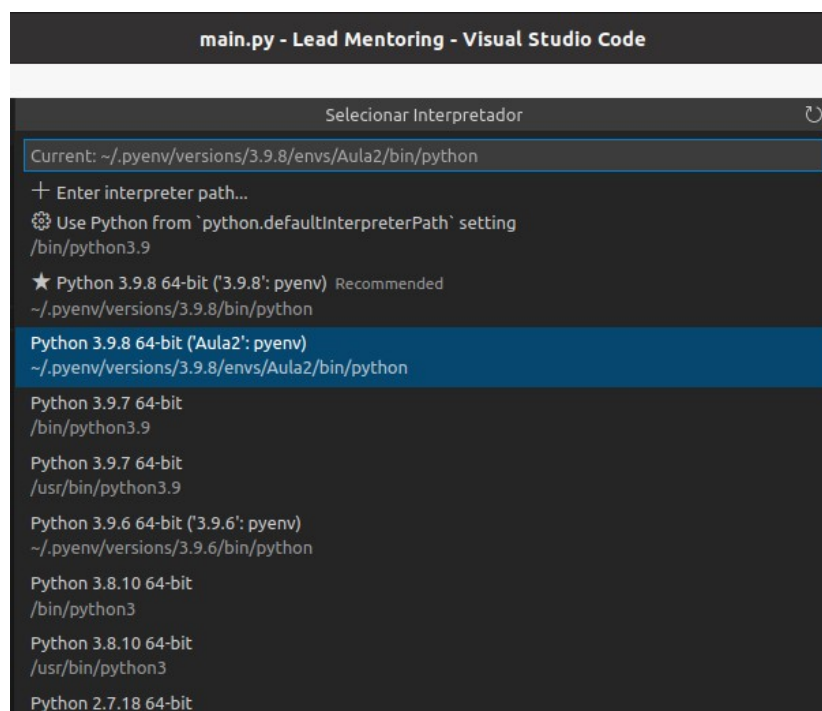
```
jonnas@JonnasCSP:~/Documentos/Lead Mentoring$ pyenv install 3.9.8
```

Pronto, agora é hora de criar um ambiente virtual e renomear com a funcionalidade “*virtualenv*”. Desta forma é preciso informar a versão que você vai querer para o projeto e o nome do ambiente.

```
jonnas@JonnasCSP:~/Documentos/Lead Mentoring$ pyenv virtualenv 3.9.8 Aula2
```

3. Ambiente Virtual

Sendo assim, agora no Visual Studio Code, vai conter o ambiente criado, é só selecionar.



Tendo tudo feito, para comprovar que tenho um ambiente virtual novo, vê-se que o Python não reconhece os módulos abaixo, justamente por não ter instalado.

```
2 import fastapi
3 import numpy as np
4 import pandas
5 import matplotlib
```

Instalando no terminal:

```
(Aula2) jonnas@JonnasCSP:~/Documentos/Lead Mentoring/02_Aula_Exercicio$ pip install fastapi
```

```
(Aula2) jonnas@JonnasCSP:~/Documentos/Lead Mentoring/02_Aula_Exercicio$ pip install numpy
```

```
(Aula2) jonnas@JonnasCSP:~/Documentos/Lead Mentoring/02_Aula_Exercicio$ pip install pandas
```

```
(Aula2) jonnas@JonnasCSP:~/Documentos/Lead Mentoring/02_Aula_Exercicio$ pip install matplotlib
```

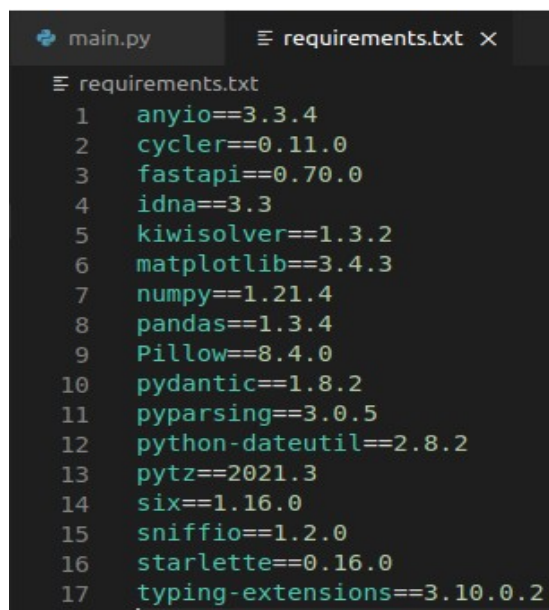
Após isso, com um “*pip freeze*” é possível visualizar todas as bibliotecas e dependências delas baixadas no ambiente virtual.

```
(Aula2) jonnas@JonnasCSP:~/Documentos/Lead Mentoring/02_Aula_Exercicio$ pip freeze
anyio==3.3.4
cyclr==0.11.0
fastapi==0.70.0
idna==3.3
kiwisolver==1.3.2
matplotlib==3.4.3
numpy==1.21.4
pandas==1.3.4
Pillow==8.4.0
pydantic==1.8.2
pyparsing==3.0.5
python-dateutil==2.8.2
pytz==2021.3
six==1.16.0
sniffio==1.2.0
starlette==0.16.0
typing-extensions==3.10.0.2
```

Com o módulo “os” do Python, podemos colocar a informação escrito em um arquivo de forma automatizada. Para isso:

```
56 # Criando o requirements automatizado com todas as bibliotecas do ambiente virtual
57 os.system('pip freeze > requirements.txt')
```

Portanto, isso tudo resulta em um *requirements* com as bibliotecas instaladas do ambiente virtual:

A screenshot of a code editor with two tabs: 'main.py' and 'requirements.txt'. The 'requirements.txt' tab is active, showing a list of 17 dependencies, each on a new line, numbered 1 through 17. The dependencies are: anyio==3.3.4, cyclr==0.11.0, fastapi==0.70.0, idna==3.3, kiwisolver==1.3.2, matplotlib==3.4.3, numpy==1.21.4, pandas==1.3.4, Pillow==8.4.0, pydantic==1.8.2, pyparsing==3.0.5, python-dateutil==2.8.2, pytz==2021.3, six==1.16.0, sniffio==1.2.0, starlette==0.16.0, and typing-extensions==3.10.0.2.

```
main.py requirements.txt x
requirements.txt
1 anyio==3.3.4
2 cyclr==0.11.0
3 fastapi==0.70.0
4 idna==3.3
5 kiwisolver==1.3.2
6 matplotlib==3.4.3
7 numpy==1.21.4
8 pandas==1.3.4
9 Pillow==8.4.0
10 pydantic==1.8.2
11 pyparsing==3.0.5
12 python-dateutil==2.8.2
13 pytz==2021.3
14 six==1.16.0
15 sniffio==1.2.0
16 starlette==0.16.0
17 typing-extensions==3.10.0.2
```