



Gerenciando os ambientes virtuais com Pyenv

≡ Módulo	Ciclo 03: Gerenciando ambientes virtuais com pyenv
# Aula	15
<input checked="" type="checkbox"/> Ready	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Finished	<input type="checkbox"/>
<input checked="" type="checkbox"/> Validated	<input checked="" type="checkbox"/>

Objetivo da Aula:

- ☐ Executar as principais tarefas de gerenciamento de ambientes virtuais com o Pyenv
- ☐ Resumo
- ☐ Próxima aula

Conteúdo:

▼ 1. Contexto do problema

No primeiro ciclo, entendemos o que acontece se não usarmos ambientes virtuais. No segundo, entendemos a importância de usar o CLI.

Na aula anterior do ciclo atual, aprendemos como instalar o Pyenv no nosso computador.

Na presente aula, iremos aprender as realizar as principais tarefas relacionadas ao gerenciamento de ambientes virtuais

com o Pyenv.

▼ 2. Executar as principais tarefas de gerenciamento de ambientes virtuais com o Pyenv

- Listar os comandos pyenv e uma explicação resumida deles

```
pyenv help
```

- Listar as versões do python disponiveis. A com asterisco é a versão ativa no momento

```
pyenv versions
```

- Listar todas as versões do python disponíveis para serem instaladas

```
pyenv install --list
```

- Instalar uma versão específica do python. Ex.: instalar a última versão do python **3.9**

```
pyenv install 3.9
```

- Desinstalar uma versão específica do python. Ex.: desinstalar a versão **3.8** do python

```
pyenv uninstall 3.8
```

- Listar todos os ambientes virtuais criados

```
pyenv virtualenvs
```

- Criar um novo ambiente virtual com o nome de `new_env` usando a versão `3.9` do python

```
pyenv virtualenv 3.9 new_env
```

- Ativar o ambiente virtual com o nome de `new_env`

```
pyenv activate new_env
```

- Desativar o ambiente virtual que está ativo

```
pyenv deactivate
```

- Configurar para que sempre que você entrar na pasta `new_env` o ambiente virtual `new_env` seja ativado. Esse ambiente local terá maior prioridade que o ambiente global.

```
# entrar na pasta new_env
cd new_env

# listar o nome de todos os ambientes virtuais criados
pyenv virtualenvs

# listar todos os arquivos na pasta
ls -la

# configurar que o ambiente padrão da pasta new_env é o new_env
pyenv local new_env

# listar todos os arquivos e perceber o arquivo .python-version
ls -la

# checar o conteúdo do arquivo .python-version -> nome do ambiente pyenv
cat .python-version
```

- Desfazer a configuração do pyenv local, ou seja, não ativar automaticamente o ambiente `new_env` quando entrar na pasta `new_env`

```
# entrar na pasta new_env onde foi feito o pyenv local
cd new_env
```

```
# listar todos os arquivos e perceber o arquivo .python-version
ls -la

# desativar o pyenv local
pyenv local --unset

# listar todos os arquivos e perceber que o arquivo .python-version não existe mais
ls -la
```

- Configurar a versão global padrão do python que será usado caso nenhuma versão específica seja ativada. Essa versão global terá menor prioridade que a versão local e a versão shell.

```
# listar o nome de todos os ambientes virtuais criados
pyenv versions

# definir o python padrão como o 3.9
pyenv global 3.9

# listar o nome de todos os ambientes virtuais criados
pyenv versions

# mostrar a versão global
pyenv global
```

- Configurar a versão shell padrão do python que será usado caso nenhuma versão específica seja ativada para a sessão do terminal aberta. Essa versão shell terá maior prioridade que a versão global e versão local. Caso abra uma nova sessão do terminal, o `pyenv shell` não terá “efeito”

```
# listar o nome de todos os ambientes virtuais criados
# e checar a versão padrão
pyenv versions

# abrir o python shell e verificar a versão do python
python
# sair do python shell
exit()

# definir o python 3.10.6 como o padrão do shell
pyenv shell 3.10.6

# abrir o python shell e verificar a versão do python
python
```

```
# sair do python shell
exit()
```

- Deletar o ambiente virtual `new_env`

```
# listar o nome de todos os ambientes virtuais criados
pyenv versions

# deletar o ambiente new_env
pyenv virtualenv-delete new_env

# listar o nome de todos os ambientes virtuais criados
pyenv versions
```

▼ 3. Resumo

- O Pyenv é uma biblioteca bem simples e acessível de usar para gerenciamento de ambientes virtuais.
- Existem muitos comandos bem como detalhes técnicos por trás do Pyenv; todavia, para o dia a dia de data science, precisaremos de poucos comandos que são intuitivos de usar com um pouco de prática.
- A documentação do Pyenv é muito completa sendo recomendado ler caso queira se aprofundar mais.

- **REFERENCIAS:**

- <https://github.com/pyenv/pyenv/blob/master/COMMANDS.md>
- <https://realpython.com/intro-to-pyenv/>
- **Vídeos CDS:**
 - **Ambientes Virtuais em Python - O QUE SÃO E COMO FUNCIONAM!**
 - <https://www.youtube.com/watch?v=nDC5pUyPqjA>

▼ 4. Próxima aula

Entendendo como o Pyenv funciona por trás dos comandos