



Certified Tech Developer

The Ultimate Degree

Objetivos

Nos exercícios a seguir, vamos executar um script em Python para exemplificar seu conteúdo.

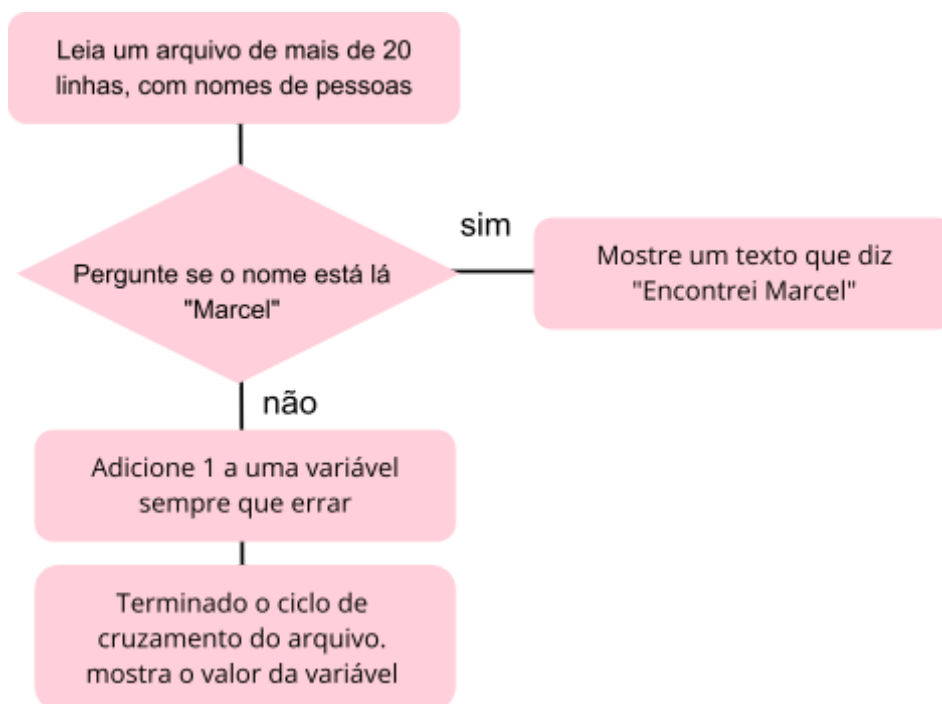
O que iremos fazer?

Um script simples de backup de arquivos.

Instruções

Exercício 1

Individualmente, faça um script no Python que deve fazer o seguinte:



Os passos que devemos seguir para preparar nosso exercício são:

- 1) Primeiro crie um arquivo, podemos usar o Vscode, Bloco de Notas ou Notepad++ e inserir uma lista de 20 nomes. Finalmente, salvamos o arquivo com o nome 'lista_nomes.txt'.
- 2) Agora vamos criar o **scriptnomes.py** (Podendo usar o Vscode ou Notepad++)
Nota: a extensão padrão para os scripts em Python é .py
- 3) Agora vamos colocar o seguinte conteúdo:

```
import fileinput
a = 0
for nome in fileinput.input("C:\\backupteste\\lista_nomes.txt"):
    if nome.strip() == "Marcel":
        print ("Encontrei o Marcel")
    else:
        a += 1

print ("Encontrei outros ", a, "nomes")
```

Salve e feche o arquivo e execute o script:

```
python3 scriptnomes.py
```

Exercício 2

Backup simples de arquivos

Na pasta do seu usuário crie duas subpastas **teste** e **backup**

Para fazer isso podemos usar o Powershell, cmd, bash etc. **Entrar na sua pasta de usuário** e digitar os comandos:

```
mkdir teste
mkdir backup
```

Dentro da pasta **teste** crie 3 arquivos .txt

```
touch teste1.txt teste2.txt teste3.txt
```

1. Vamos criar um arquivo no vscode ou notepad++ com o nome **scriptbackup.py**

O arquivo deverá ficar da seguinte forma:

(lembrando que o "Emp1" deve ser alterado para o seu nome de usuário do windows):

```
from datetime import datetime
from pathlib import Path
import zipfile

OBJECT_TO_BACKUP = 'C:\\Users\\Emp1\\Documents\\backup\\teste\\'
BACKUP_DIRECTORY = 'C:\\Users\\Emp1\\Documents\\backup\\backup\\'
MAX_BACKUP_AMOUNT = 5

object_to_backup_path = Path(OBJECT_TO_BACKUP)
backup_directory_path = Path(BACKUP_DIRECTORY)
assert object_to_backup_path.exists() # Validate the object we are
about to backup exists before we continue

# Validate the backup directory exists and create if required
backup_directory_path.mkdir(parents=True, exist_ok=True)

# Get the amount of past backup zips in the backup directory already
existing_backups = [
    x for x in backup_directory_path.iterdir()
        if x.is_file() and x.suffix == '.zip' and
        x.name.startswith('backup-')
]

# Enforce max backups and delete oldest if there will be too many after
the new backup
oldest_to_newest_backup_by_name = list(sorted(existing_backups,
key=lambda f: f.name))
```

```
while len(oldest_to_newest_backup_by_name) >= MAX_BACKUP_AMOUNT: # >=
because we will have another soon
    backup_to_delete = oldest_to_newest_backup_by_name.pop(0)
    backup_to_delete.unlink()

# Create zip file (for both file and folder options)
backup_file_name =
f'backup-{datetime.now().strftime("%Y%m%d%H%M%S")}.zip'
zip_file = zipfile.ZipFile(str(backup_directory_path +
backup_file_name), mode='w')
if object_to_backup_path.is_file():
    # If the object to write is a file, write the file
    zip_file.write(
        object_to_backup_path.absolute(),
        arcname=object_to_backup_path.name,
        compress_type=zipfile.ZIP_DEFLATED
    )
elif object_to_backup_path.is_dir():
    # If the object to write is a directory, write all the files
    for file in object_to_backup_path.glob('**/*'):
        if file.is_file():
            zip_file.write(
                file.absolute(),
                arcname=str(file.relative_to(object_to_backup_path)),
                compress_type=zipfile.ZIP_DEFLATED
            )
# Close the created zip file
zip_file.close()
```

Salve e feche o arquivo.

Agora execute o seguinte comando no Powershell ou cmd dentro da sua pasta de usuário:

```
python scriptbackup.py
```