**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:

| python ou 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 |
| --- |