Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Às vezes temos a necessidade de armazenar informações intermediárias no disco para uma tarefa específica.

Para isso podemos criar arquivos e/ou pastas temporários.

Nesta aula veremos como trabalhar com arquivos e pastas temporárias no ***Python***.

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Para trabalhar com arquivos e diretórios temporários, vamos utilizar a biblioteca padrão denominada ***tempfile***.

Esta biblioteca possui um método chamado ***TemporaryFile()*** que retorna um objeto similar a um arquivo para armazenarmos informações.

Criamos o objeto, o ***Python*** cria o arquivo em disco, trabalhamos com ele e, após fecharmos este objeto, o arquivo é destruído automaticamente.

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Veja um exemplo e o resultado de sua execução.

```
import tempfile

# Criando um arquivo temporário
arq = tempfile.TemporaryFile()

# Imprimindo o objeto tempfile criado
print("Arquivo criado:", arq)

# Imprimindo o local e nome do arquivo temporário
print("Nome do arquivo:", arq.name)

# Fechando o arquivo
arq.close()
```

```
Arquivo criado: <tempfile._TemporaryFileWrapper object at 0x000001DC9A8671C0>
Nome do arquivo: C:\Users\Evaldo\AppData\Local\Temp\tmpeuuiq51s
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Agora vamos adicionar conteúdo ao arquivo. Mas o conteúdo a ser adicionado tem que ser em ***byte***. Antes do próximo exemplo, vale a pena explicar como gravar uma String em bytes usando o prefixo **b** antes da string.

Faça este exemplo e veja o resultado.

```
a = "Python"
b = b"Python"
print("O conteúdo de 'a' é:", a)
print("O conteúdo de 'b' é:", b)
print("O tipo de 'a' é:", type(a))
print("O tipo de 'b' é:", type(b))
```

```
O conteúdo de 'a' é: Python
O conteúdo de 'b' é: b'Python'
O tipo de 'a' é: <class 'str'>
O tipo de 'b' é: <class 'bytes'>
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Vamos agora adicionar o texto “Curso Python para Todos” ao arquivo criado. Ler e imprimir seu conteúdo.

```
import tempfile

# Criando um arquivo temporário
arq = tempfile.TemporaryFile()

# Gravando informação no arquivo
arq.write(b'Curso Python para Todos’)

# Indo para o início do arquivo
arq.seek(0)

# Imprimir o conteúdo do arquivo
print(arq.read())

# Fechar o arquivo
arq.close()
```

Veja o programa e o resultado da execução.

```
b'Curso Python para Todos'
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Agora vamos usar **with** para trabalhar com um arquivo temporário. Usando **with**, o arquivo é fechado automaticamente e consequentemente, excluído.

```
import tempfile

with tempfile.TemporaryFile() as arq:
    arq.write(b'Curso Python para Todos')
    arq.seek(0)
    print(arq.read())
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Caso precise identificar o arquivo temporário em seu sistema, pode adicionar um prefixo e/ou sufixo no nome do arquivo através dos parâmetros prefix e suffix.

```
import tempfile

with tempfile.TemporaryFile(prefix="Python_", suffix="_ParaTodos") as arq:
    print("Nome do arquivo:", arq.name)
```

Veja a execução:

```
Nome do arquivo: C:\Users\Evaldo\AppData\Local\Temp\Python_dfbjaf7u_ParaTodos
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Agora vamos escrever várias linhas no arquivo usando ***writelines***. O método ***writelines*** recebe uma lista contendo as strings. Para dar um "enter", colocamos o "\n" dentro da string, no final da frase.

O modo ***w*** informa que vamos escrever no arquivo (já é padrão) e o ***t*** informa que será um arquivo de texto simples (plain text).

```
import tempfile

with tempfile.TemporaryFile(mode='w+t') as arq:
    arq.writelines(['Curso Python para Todos\n',
                    'Trabalhando com arquivos e diretórios temporários\n'])
    arq.seek(0)
    print(arq.read())
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Agora vamos alterar o caminho do diretório temporário onde os arquivos são criados.

```
import tempfile

print("Diretório temporário corrente:", tempfile.gettempdir())
# Vamos alterar o diretório temporário para C:\temp
# Para que seja reconhecida a contra-barra devemos colocar duas "\\"
tempfile.tempdir = "C:\\temp"
print("Diretório temporário após alteração:", tempfile.gettempdir())

# Veja agora onde nosso arquivo temporário será criado
with tempfile.TemporaryFile() as arq:
    print("Nome do arquivo:", arq.name)
```

Resultado:

```
Diretório temporário corrente: C:\Users\Evaldo\AppData\Local\Temp
Diretório temporário após alteração: C:\temp
Nome do arquivo: C:\temp\tmp8h9ssn7g
```

Arquivos e diretórios

Trabalhando com arquivos e diretórios temporários

Para criar um diretório/pasta podemos usar o método ***TemporaryDirectory***.

```
import tempfile

with tempfile.TemporaryDirectory() as tmpdirname:
    print('Diretório temporário:', tmpdirname)

# O diretório e seu conteúdo foi deletado
```

Veja o nome do diretório criado, vá ao caminho exibido no print e veja que o diretório foi apagado após ser criado.

FIM