INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Sapiranga

Linguagem de Programação III

Aula 12

Arquivos e Path de Arquivos

Prof. Guilherme Xavier

Todas as funções para gerenciar arquivos e pastas em Python estão no módulo **os**.

```
# Diretorio atual getcwd()
>>import os
>>retorno= os.getcwd()
>>print("meu diretorio atual é: ", retorno)
```

#Pasta local e Alterar Pastas

```
import os
# Diretorio atual
print("meu diretorio atual é:", os.getcwd())

# Alterar diretorios
os.chdir('c:\\windows\\system32')
os.getcwd()
print("alterei para:", os.getcwd())
```

#Criar Patas /Diretórios

 # Criar pastas
os.makedirs(**'C:\\Users\\Guilherme Xavier\\Documentos\\ifsul\\python\\Aula11'**)

```
import os
## Verificar o Tamanho do arquivo

print(os.path.getsize('C:\\Users\\Guilherme Xavier\\Documentos\\ifsul\\python\\Python.docx'))
```

```
# tamanho total de todos os arquivos da pasta
diretorio='C:\\Users\\Guilherme Xavier\\Documentos\\ifsul\\python'
totalSize=0;
for filename in os.listdir(diretorio):
    totalSize=totalSize + os.path.getsize(os.path.join(diretorio,filename))

print (totalSize)
```

```
# Conteudo das pastas
retorno=os.listdir('C:\\Users\\Guilherme Xavier\\Documentos\\ifsul\\python')
print(retorno)

#Verificando se o path (caminho) existe
retorno=os.path.exists('c:\\windows')
print("A pasta c:\windows Existe? ", retorno)
```

Organizando Arquivos

Módulo shutil

O módulo shutil (shell utilities, ou utilitários de shell) contém funções que permitem copiar, mover, renomear e apagar arquivos em seus programas Python.

Para usar as funções de shutil, inicialmente você deverá usar

```
import shutil
```

Copiando arquivos e pastas

O módulo shutil disponibiliza funções para copiar arquivos bem como pastas inteiras.
Chamar shutil.copy(origem, destino) copiará o arquivo no path origem para a pasta no path destino.
(Tanto origem quanto destino são strings.)

Se destino for o nome de um arquivo, ele será usado como o novo nome do arquivo copiado.
Essa função retorna uma string com o path do arquivo copiado.

Digite o seguinte no shell interativo para ver como shutil.copy() funciona:

```
import os , shutil
os.chdir("c:\\")
shutil.copy("C:\windows\system32\alttab.dll",
'C:\\Users\\aluno\\Documentos\\ifsul\\python')
```

#Copiando pastas

shutil.copytree('**C:\\pasta Original**', '**C:\\pasta_bkp**')

Movendo e renomeando arquivos e pastas

Chamar shutil.move(origem, destino)
Moverá o arquivo ou a pasta no path origem
para o path destino e retornará uma string com o path absoluto da nova localidade.
Se destino apontar para uma pasta, o arquivo origem será movido para destino e preservará seu nome de arquivo atual.
Por exemplo, digite o seguinte no shell interativo

```
>>> import shutil
>>> shutil.move('C:\\bacon.txt', 'C:\\eggs')
'C:\\eggs\\bacon.txt'
```

Supondo que uma pasta chamada eggs já exista no diretório C:\, essa chamada a shutil.move() diz para "mover C:\bacon.txt para a pasta C:\eggs".
Se já houver um arquivo bacon.txt em C:\eggs, ele será sobrescrito.
Como é fácil sobrescrever arquivos acidentalmente dessa maneira, você deve tomar alguns cuidados ao usar move().

O path destino também pode especificar um nome de arquivo.
No exemplo a seguir, o arquivo origem será movido e renomeado.

```
>>> shutil.move('C:\\bacon.txt', 'C:\\eggs\\new_bacon.txt')
'C:\\eggs\\new_bacon.tx
```

Apagando arquivos e pastas permanentemente

Podemos apagar um único arquivo ou uma única pasta vazia com funções do módulo os, ao passo que, para apagar uma pasta e todo o seu conteúdo, devemos usar o módulo shutil.

- Chamar os.unlink(path) apagará o arquivo em path.
- Chamar os.rmdir(path) apagará a pasta em path. Essa pasta deve estar vazia, ou seja, não deve conter nenhum arquivo ou pasta.
- Chamar shutil.rmtree(path) removerá a pasta em path; além disso, todos os arquivos e as pastas nele contidos também serão apagados.

Tome cuidado ao usar essas funções em seus programas! Geralmente, é uma boa ideia executar seu programa inicialmente com essas chamadas comentadas, adicionando chamadas a print() para mostrar os arquivos que seriam apagados.

Eis um programa Python que deveria apagar arquivos com extensão .txt, porém com um erro de digitação (destacado em negrito) que o faz apagar arquivos .rxt:

```
import os
for filename in os.listdir():
        if filename.endswith('.rxt'):
                os.unlink(filename)
```

Se houvesse algum arquivo importante terminado com .rxt, eles teriam sido apagados acidentalmente de forma permanente.

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SUL-RIO-GRANDENSE
Campus Sapiranga

1-Exercicios

Crie as pastas
AulaPython
- Apostilas
  - Python
  - TKInter
- Programas
  - Modelos
- Arquivos
  - Documentos

2-Exercicios

2.1 Copie os arquivos PDF da aula 19/04 para pasta AulaPython\Arquivos\Documentos

2.2 Copie os arquivos PDF da aula 05/04 para pasta AulaPython\Apostilas\Python

2.3 Copie os arquivos PDF da aula de hoje para pasta AulaPython\Apostilas\Python

3. Exercícios

3.1 Informe seu diretório

3.2 Alterar diretório AulaPython\Programas\Apostilas\Python

4. Exercicios

4.1 Liste os arquivos na pasta AulaPython\Programas\Apostilas\Python

42. Verifique o *tamanho total de todos os arquivos da pasta*

5. Exercícios

5.1 Copie os arquivos da pasta AulaPython\Programas\Apostilas\Python Para uma pasta BKP

52. Mostre o conteúdo dessa nova pasta