

Python - Diretórios

Todos os arquivos estão contidos em vários diretórios e o Python também não tem problemas em lidar com eles. O módulo **os** possui vários métodos que ajudam a criar, remover e alterar diretórios.

O método mkdir()

Você pode usar o método mkdir() do módulo os para criar diretórios no diretório atual. Você precisa fornecer um argumento para este método, que contém o nome do diretório a ser criado.

Sintaxe

```
os.mkdir("newdir")
```

Exemplo

A seguir está um exemplo para criar um diretório de teste no diretório atual -

```
#!/usr/bin/python3
import os

# Create a directory "test"
os.mkdir("test")
```

O método chdir()

Você pode usar o método chdir() para alterar o diretório atual. O método chdir() recebe um argumento, que é o nome do diretório que você deseja tornar o diretório atual.

Sintaxe

```
os.chdir("newdir")
```

Exemplo

A seguir está um exemplo para entrar no diretório "/home/newdir" -



```
import os

# Changing a directory to "/home/newdir"
os.chdir("/home/newdir")
```

O método getcwd()

O método getcwd() exibe o diretório de trabalho atual.

Sintaxe

```
os.getcwd()
```

Exemplo

A seguir está um exemplo para fornecer o diretório atual -

```
#!/usr/bin/python3
import os

# This would give location of the current directory
os.getcwd()
```

O método rmdir()

O método rmdir() exclui o diretório, que é passado como argumento no método.

Antes de remover um diretório, todo o seu conteúdo deve ser removido.

Sintaxe

```
os.rmdir('dirname')
```

Exemplo

A seguir está um exemplo para remover o diretório "/tmp/test". É necessário fornecer o nome totalmente qualificado do diretório, caso contrário, ele procuraria esse diretório no diretório atual.

```
#!/usr/bin/python3
import os
```



```
# This would remove "/tmp/test" directory.  
os.rmdir( "/tmp/test" )
```