

Python – O bloco try-finalmente

Você pode usar um bloco **finalmente** : junto com um bloco **try** :. O bloco **finalmente** : é um local para colocar qualquer código que deva ser executado, independentemente de o bloco try ter gerado uma exceção ou não.

A sintaxe da instrução **try-finalmente** é esta -

```
try:
    You do your operations here;
    .....
    Due to any exception, this may be skipped.
finally:
    This would always be executed.
    .....
```

Note - Você pode fornecer cláusula(s) exceto ou uma cláusula final, mas não ambas. Você não pode usar a cláusula else junto com uma cláusula final.

Exemplo

```
try:
    fh = open("testfile", "w")
    fh.write("This is my test file for exception handling!!")
finally:
    print ("Error: can't find file or read data")
    fh.close()
```

Se você não tiver permissão para abrir o arquivo no modo de escrita, a seguinte **saída** será produzida -

```
Error: can't find file or read data
```

O mesmo exemplo pode ser escrito de forma mais clara da seguinte forma -

```
try:
    fh = open("testfile", "w")
    try:
        fh.write("This is my test file for exception handling!!")
    finally:
        print ("Going to close the file")
```

```
fh.close()
except IOError:
    print ("Error: can\'t find file or read data")
```

Quando uma exceção é lançada no bloco try, a execução passa imediatamente para o bloco **final** . Depois que todas as instruções no bloco **final** são executadas, a exceção é levantada novamente e tratada nas instruções except se estiver presente na próxima camada superior da instrução try-except.

Exceção com argumentos

Uma exceção pode ter um argumento, que é um valor que fornece informações adicionais sobre o problema. O conteúdo do argumento varia por exceção. Você captura o argumento de uma exceção fornecendo uma variável na cláusula except da seguinte maneira -

```
try:
    You do your operations here
    .....
except ExceptionType as Argument:
    You can print value of Argument here...
```

Se você escrever o código para lidar com uma única exceção, poderá fazer com que uma variável siga o nome da exceção na instrução except. Se você estiver capturando várias exceções, poderá fazer com que uma variável siga a tupla da exceção.

Esta variável recebe o valor da exceção contendo principalmente a causa da exceção. A variável pode receber um único valor ou vários valores na forma de uma tupla. Essa tupla geralmente contém a string do erro, o número do erro e um local do erro.

Exemplo

A seguir está um exemplo de uma única exceção -

```
# Define a function here.
def temp_convert(var):
    try:
        return int(var)
    except ValueError as Argument:
        print("The argument does not contain numbers\n",Argument)
# Call above function here.
temp_convert("xyz")
```

Ele produzirá a seguinte **saída** -

The argument does not contain numbers
invalid literal for int() with base 10: 'xyz'