

Python - Threads Daemon

Às vezes é necessário executar uma tarefa em segundo plano. Um tipo especial de thread é usado para tarefas em segundo plano, chamado **thread daemon**. Em outras palavras, os threads daemon executam tarefas em segundo plano.

Pode-se notar que os threads daemon executam tarefas não críticas que, embora possam ser úteis para o aplicativo, não o prejudicam se falharem ou forem canceladas no meio da operação.

Além disso, um thread daemon não terá controle sobre quando será encerrado. O programa será encerrado assim que todos os threads não-daemon terminarem, mesmo se ainda houver threads daemon em execução naquele momento.

Esta é uma grande diferença entre threads daemon e threads não-daemon. O processo será encerrado se apenas threads daemon estiverem em execução, mas não poderá sair se pelo menos um thread não-daemon estiver em execução.

Demônio	Não-daemon
Um processo será encerrado se apenas threads daemon estiverem em execução (ou se nenhum thread estiver em execução).	Um processo não será encerrado se pelo menos um thread não-daemon estiver em execução.

Criando um Thread Daemon

Para criar um thread daemon, você precisa definir a propriedade **do daemon** como True.

```
t1=threading.Thread(daemon=True)
```

Se um objeto thread for criado sem nenhum parâmetro, sua propriedade daemon também poderá ser definida como True, antes de chamar o método start().

```
t1=threading.Thread()  
t1.daemon=True
```

Exemplo

Dê uma olhada no exemplo a seguir -

```
from time import sleep  
from threading import current_thread
```



```

from threading import Thread

# function to be executed in a new thread
def run():

    # get the current thread
    thread = current_thread()
    # is it a daemon thread?
    print(f'Daemon thread: {thread.daemon}')

# create a new thread
thread = Thread(target=run, daemon=True)

# start the new thread
thread.start()

# block for a 0.5 sec for daemon thread to run
sleep(0.5)

```

Ele produzirá a seguinte **saída** -

```

Daemon thread: True

```

Threads daemon podem executar tarefas de execução que suportam threads não-daemon no programa. Por exemplo -

- Crie um arquivo que armazene informações de log em segundo plano.
- Execute web scraping em segundo plano.
- Salve os dados automaticamente em um banco de dados em segundo plano.

Exemplo

Se um thread em execução estiver configurado para ser daemon, um RuntimeError será gerado. Dê uma olhada no exemplo a seguir -

```

from time import sleep
from threading import current_thread
from threading import Thread

# function to be executed in a new thread
def run():
    # get the current thread
    thread = current_thread()

```



```
# is it a daemon thread?
print(f'Daemon thread: {thread.daemon}')
thread.daemon = True

# create a new thread
thread = Thread(target=run)

# start the new thread
thread.start()

# block for a 0.5 sec for daemon thread to run
sleep(0.5)
```

Ele produzirá a seguinte **saída** -

Exception in thread Thread-1 (run):

Traceback (most recent call last):

....

....

thread.daemon = True

^^^^^^^^^^^^^^

File "C:\Python311\Lib\threading.py", line 1219, in daemon

raise RuntimeError("cannot set daemon status of active thread")

RuntimeError: cannot set daemon status of active thread