

The thread module

(Optional). This module provides a low-level interface for threading. It's only available if your interpreter is built with thread support. New code should use the higher-level interface in the [threading](#) module instead.

Example: Using the thread module

```
# File: thread-example-1.py

import thread
import time, random

def worker():
    for i in range(50):
        # pretend we're doing something that takes 10-100 ms
        time.sleep(random.randint(10, 100) / 1000.0)
        print thread.get_ident(), "-- task", i, "finished"

#
# try it out!


for i in range(2):
    thread.start_new_thread(worker, ())

time.sleep(1)

print "goodbye!"

311 -- task 0 finished
265 -- task 0 finished
265 -- task 1 finished
311 -- task 1 finished
...
265 -- task 17 finished
311 -- task 13 finished
265 -- task 18 finished
goodbye!
```

Note that when the main program exits, all threads are killed. The [threading](#) module doesn't have that problem.

 rendered by a [django](#) application. hosted by [webfaction](#).