Sleeping and Terminating Threads

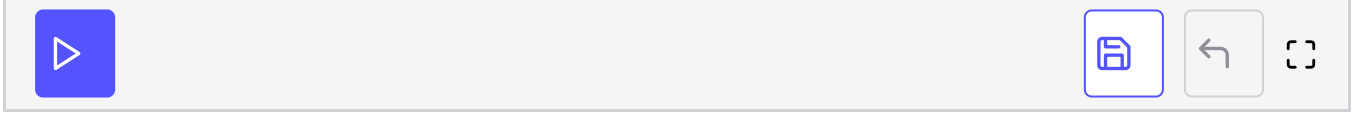
This lesson talks about putting threads to sleep and terminating them.

Sleeping Threads

It's trivial to make a thread sleep and give up its time slice on the CPU using the `sleep()` method, which accepts a number as an argument. The argument can also be fractional and specifies the number of seconds to sleep.

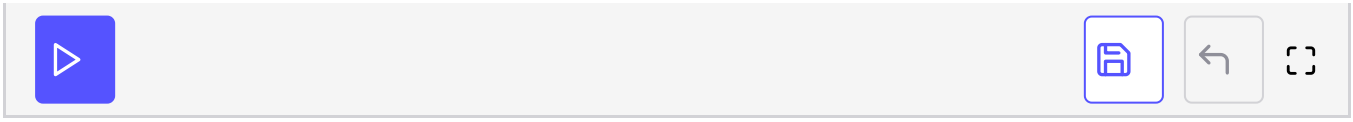
Consider the example below. The spawned child thread sleeps for 5 seconds. The main thread remains blocked until the child thread exits.

```
require 'date'

Thread.current.name = "mainThread"
start = DateTime.now.to_time
# spawn a child thread
thread = Thread.new do
  # sleep for 5.5 seconds
  sleep(5)
end

# wait for the thread to finish
thread.join()

time_difference_in_sec = (DateTime.now.to_time.to_i - start.to_time.to_i).abs
puts("Program takes to run #{time_difference_in_sec}")
```



In the previous example, the main thread explicitly waits for the child thread to complete using `join()`. Run the code widget below and see what happens when we don't wait for a spawned thread to complete.

```
require 'date'

Thread.current.name = "mainThread"
start = DateTime.now.to_time
# Spawn a child thread
thread = Thread.new do
  # sleep for 1 day
  sleep(60*60*24)

  # Never gets printed
  puts("Child thread exiting")
end

puts("Main thread exiting")
```

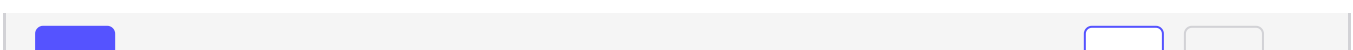


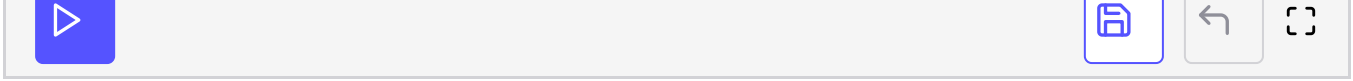
The main thread and the application exit without waiting for the child thread to complete. Contrast this to Python and some other technologies, where if a spawned thread isn't marked as background/daemon thread, the entire program hangs until the background/daemon threads complete execution.

Terminating Threads

We can use the `kill()` method to terminate a thread.

```
require 'date'

Thread.current.name = "mainThread"
start = DateTime.now.to_time
# Spawn a child thread
thread = Thread.new do
  # sleep for 1 day
  sleep(60*60*24)

  # Never gets printed
  puts("Child thread exiting")
end

sleep(1)
Thread.kill(thread)

puts("Main thread exiting while child thread status #{thread.status}")
```





On **line#15**, we invoke the kill command to terminate the rogue child thread. An alternative to killing the thread is to invoke the `exit()` method on the child thread object.

```
require 'date'

Thread.current.name = "mainThread"
start = DateTime.now.to_time
# Spawn a child thread
thread = Thread.new do
  # sleep for 1 day
  sleep(60*60*24)

  # Never gets printed
  puts("Child thread exiting")
end

sleep(1)
thread.exit()
sleep(1)

puts("Main thread exiting while child thread status #{thread.status}")
```