import threading

t = threading.Thread(target=fn, args=[], kwargs={})

t.start()

t.join() # to wait until thread finished

import concurrent.futures  
from time import sleep, time  
  
  
def fn(seconds):  
 global id  
 id += 1  
 local\_id = id  
 print(f'Start sleep for {seconds} secs in fn{local\_id}')  
  
 sleep(seconds)  
 return f'End sleeping in fn{local\_id}'  
  
  
id = 0  
start\_time = time()  
with concurrent.futures.ThreadPoolExecutor() as executor:  
 secs = [5, 4, 3, 2, 1]  
 threads = [executor.submit(fn, sec) for sec in secs]  
 for thead in concurrent.futures.as\_completed(threads):  
 print(thead.result())  
  
print(f'Program finished in {time() - start\_time}s')

Start sleep for 5 secs in fn1

Start sleep for 4 secs in fn2

Start sleep for 3 secs in fn3

Start sleep for 2 secs in fn4

Start sleep for 1 secs in fn5

End sleeping in fn5

End sleeping in fn4

End sleeping in fn3

End sleeping in fn2

End sleeping in fn1

Program finished in 5.007075548171997s

with concurrent.futures.ThreadPoolExecutor() as executor:  
 secs = [5, 4, 3, 2, 1]  
 futures = executor.map(fn, secs)  
 for res in futures:  
 print(res)

Start sleep for 5 secs in fn1

Start sleep for 4 secs in fn2

Start sleep for 3 secs in fn3

Start sleep for 2 secs in fn4

Start sleep for 1 secs in fn5

End sleeping in fn1

End sleeping in fn2

End sleeping in fn3

End sleeping in fn4

End sleeping in fn5

Program finished in 5.00705623626709s

import multiprocessing  
from time import sleep, time  
  
  
def fn(seconds):  
 print(f'Start sleep for {seconds} secs')  
  
 sleep(seconds)  
 print('End sleeping')  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 start\_time = time()  
 secs = [5, 4, 3, 2, 1]  
 processes = [multiprocessing.Process(target=fn, args=(sec,)) for sec in secs]  
 for process in processes:  
 process.start()  
 for process in processes:  
 process.join()  
  
 print(f'Program finished in {time() - start\_time}s')

Start sleep for 4 secs

Start sleep for 5 secs

Start sleep for 2 secs

Start sleep for 3 secs

Start sleep for 1 secs

End sleeping

End sleeping

End sleeping

End sleeping

End sleeping

Program finished in 5.539706230163574s

with concurrent.futures.ProcessPoolExecutor() as executor:  
 secs = [5, 4, 3, 2, 1]  
  
 processes = [executor.submit(fn, sec) for sec in secs]  
 for process in processes:  
 print(process.result())

Start sleep for 5 secs

Start sleep for 4 secs

Start sleep for 3 secs

Start sleep for 2 secs

Start sleep for 1 secs

End sleeping 5s

End sleeping 4s

End sleeping 3s

End sleeping 2s

End sleeping 1s

Program finished in 5.947681665420532s

with concurrent.futures.ProcessPoolExecutor() as executor:  
 secs = [5, 4, 3, 2, 1]  
  
 res\_iter = executor.map(fn, secs)  
 for res in res\_iter:  
 print(res)

Start sleep for 5 secs

Start sleep for 4 secs

Start sleep for 3 secs

Start sleep for 2 secs

Start sleep for 1 secs

End sleeping 5s

End sleeping 4s

End sleeping 3s

End sleeping 2s

End sleeping 1s

Program finished in 5.993886947631836s