Threading Experiment Notes

.py

import threading

class Hello(threading.Thread):

#def \_\_init\_\_(self):

# threading.Thread.\_\_init\_\_()

def run(self):

for n in range(0, 100):

print('Hello')

class World(threading.Thread):

#def \_\_init\_\_(self):

# threading.Thread.\_\_init\_\_()

def run(self):

for n in range(0, 100):

print('World')

t1 = Hello()

t2 = World()

t1.start()

t2.start()

print('Main thread exiting')

output:

Hello

Hello

Hello

World

World

World

World

World

World

World

Hello

World

Hello

Hello

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

World

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

Hello

World

Main thread exiting

Hello

Hello

Hello

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

World

Hello

Hello

World

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

World

Hello

Hello

World

Hello

Hello

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

World

Hello

World

Hello

World

Hello

World

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

Hello

World

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

Hello

World

World

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

World

Hello

World

Hello

World

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

Hello

World

World

World

World

World

World

World

# Experiment 2:

For this experiement I wanted to see if I could have:

1. A thread that would receive input from a queue originating outside of its function scope, perform operations on it, and store the results in a list also originating outside of its function scope
2. A thread that at the same time that the first thread is working, prints the list that thread 1 stores its results in

If the threads worked concurrently, then the console should print out a list multiple times, each time, the list grows in size

The experiment was a massive success!

.py:

import threading

import queue

class HelloWorker(threading.Thread):

""" This thread converts a queue of 'Hello' s into a list of 'World' s """

def \_\_init\_\_(self, hello\_queue, world\_list):

threading.Thread.\_\_init\_\_(self)

# Since Queues are mutable, this thread can access the original queue object

self.hello\_queue = hello\_queue

self.world\_list = world\_list

def run(self):

while self.hello\_queue.empty() == False:

self.hello\_queue.get()

print('HelloWorker is working')

self.world\_list.append('World')

class WorldWorker(threading.Thread):

""" operates on world\_list while the hello\_worker is feeding information into it. prints the list """

def \_\_init\_\_(self, world\_list):

threading.Thread.\_\_init\_\_(self)

self.world\_list = world\_list

def run(self):

while t1.is\_alive(): # this may give an error. may not have access to t1. probs should pass in thread as param.

print(self.world\_list)

# intitialize hello queue

hq = queue.Queue()

for x in range(50):

hq.put('Hello')

# intialize world list

wl = []

t1 = HelloWorker(hq, wl)

t2 = WorldWorker(wl)

print('hq size before threads: 50')

print('wl before threads:', wl)

t1.start()

t2.start()

t1.join()

t2.join()

print('hq is empty?:', hq.empty())

print('wl after threads:', wl)

print('Main thread exiting')

output:

hq size before threads: 50

wl before threads: []

HelloWorker is working

HelloWorker is working

HelloWorker is working

['World', 'World']

HelloWorker is working

['World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

HelloWorker is working

['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

hq is empty?: True

wl after threads: ['World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World', 'World']

Main thread exiting

# Experiment 3

Can I make api calls simultaneously? Meaning when the api is processing a request from my code, can I make another request t that api before its done processing my first request?

Result: yes, I can

.py :

import api\_interface as api

import threading

import queue

class DownloadWorker(threading.Thread):

""" This thread sends and receives api requests.

This thread intentionally modifies queues and lists that originate from outside its class/method definitions."""

def \_\_init\_\_(self, date\_queue, parsed\_data):

threading.Thread.\_\_init\_\_(self)

# stores link to queues and lists. takes advantage of Python's call-by-object-reference.

self.dq = date\_queue

self.pd = parsed\_data

def run(self):

""" makes requests for data. stores data in parsed\_data list. """

# test having multiple api calls at the same time

while self.dq.empty() == False:

data = api.get\_openex\_historical\_rates(self.dq.get())

print(data)

# test simultaneous api calls

date\_queue = queue.Queue()

date\_queue.put('2018-06-04')

date\_queue.put('2018-06-05')

date\_queue.put('2018-06-06')

t1 = DownloadWorker(date\_queue, parsed\_data=None)

t2 = DownloadWorker(date\_queue, parsed\_data=None)

t3 = DownloadWorker(date\_queue, parsed\_data=None)

t1.start()

t2.start()

t3.start()

t1.join()

t2.join()

t3.join()

print('is date queue empty?', date\_queue.empty())

print('exiting main thread')

Output:

{'disclaimer': 'Usage subject to terms: https://openexchangerates.org/terms', 'license': 'https://openexchangerates.org/license', 'timestamp': 1528243199, 'base': 'USD', 'rates': {'CHF': 0.984322, 'EUR': 0.852775, 'GBP': 0.745875, 'JPY': 109.88758333, 'USD': 1}, 'date': '2018-06-05'}

{'disclaimer': 'Usage subject to terms: https://openexchangerates.org/terms', 'license': 'https://openexchangerates.org/license', 'timestamp': 1528156799, 'base': 'USD', 'rates': {'CHF': 0.988259, 'EUR': 0.854692, 'GBP': 0.750977, 'JPY': 109.97061538, 'USD': 1}, 'date': '2018-06-04'}

{'disclaimer': 'Usage subject to terms: https://openexchangerates.org/terms', 'license': 'https://openexchangerates.org/license', 'timestamp': 1528329599, 'base': 'USD', 'rates': {'CHF': 0.986103, 'EUR': 0.848793, 'GBP': 0.745045, 'JPY': 110.1366, 'USD': 1}, 'date': '2018-06-06'}

is date queue empty? True

exiting main thread