**THREADS - 1**

**SINGLE THREAD**

As of now whatever we have used in the program are called Single Thread. The Python virtual machine behind, will use a thread called main() Thread to execute the code which we was written.

If we are not creating any threads in our application, then it’s called Single Threaded Applications.

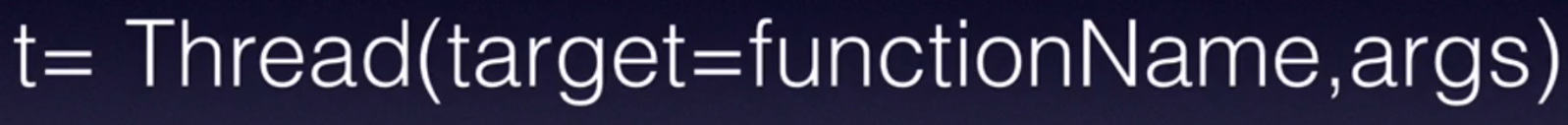
**MULTIPLE THREAD**

To make the best use of the underlying processor and to improve the performance of our application, we can create multiple threads, that can execute in parallel.

This will enhance the best use of the processor and will be very fast.

Three ways to create multiple threads in Python:

1. **Use a function:** Will create a normal function and will create an object of type thread. To that object we will pass the target which is the function, we have created, the name of the function. The second argument is arguments to the function.



Once we done, we will invoke

A white text on a blue background

Description automatically generated

The PVM will create a new thread that will run in parallel with the main() thread.

As a part of the thread, the code within this function will be executed and arguments are passed to that function.

1. **Extend the thread class:** This is the second way. This thread class again is from Python threading module, which we will import the module and extend the Thread class.

A close up of a sign

Description automatically generated

The next step is to override the run() method.

A close up of a letter

Description automatically generated

Inside the run() method, we will define all the functionality you want inside your thread.

The we will create an instance of the class (myThread) and invoke the start.

A white text on a blue background

Description automatically generated

This will spawn off a new thread and it will execute the run method internally.

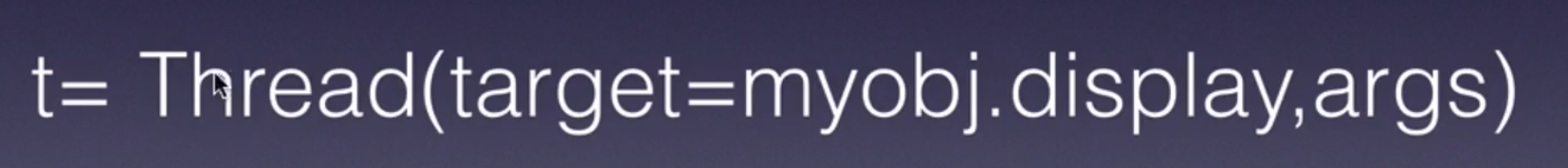
1. **With our extending the thread class: This** approach is hybrid approach, where we will create a class : A white letter on a black background

   Description automatically generated

We will add any function, what we want within the class, but this class does not extend the Thread Class A computer screen with a cursor and a word

Description automatically generated

Instead, we will create an instance of the thread and pass the object.function name and arguments.



**MAIN THREAD ACCESSED USING CODE**

**Code**:

import threading

# getName( is the older version, which was deprciated. hence can use name.

tName = threading.current\_thread().getName()

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

**Output**:

C:\Users\kamal\OneDrive\Desktop\Python\Python Examples\s\_threadEgs\mainThread.py:4: DeprecationWarning: getName() is deprecated, get the name attribute instead

tName = threading.current\_thread().getName()

Current Thread which is running : MainThread

**Code**:

import threading

# getName( is the older version, which was deprciated. hence can use name.

#tName = threading.current\_thread().getName()

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

print()

if(threading.current\_thread() == threading.main\_thread()):

print(*"In the main thread !!!"*)

else:

print(*"Other threads !!!"*)

**Output**:

Current Thread which is running : MainThread

In the main thread !!!

**MULTIPLE THREAD – USING FUNCTIONS**

**Code:**

from threading import Thread

import threading

def **displayNumber**():

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i)

i+=1

thread1 = Thread(target=displayNumber)

#CHECK FOR THREADS

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

# START THE THREAD

thread1.start()

**Output**:

Current Thread which is running : MainThread

Current Thread which is running : Thread-1 (displayNumber)

0

1

2

3

4

5

**MULTIPLE THREAD – SUBCLASSING THE SUPERCLASS THREAD**

**Code:**

from threading import Thread

import threading

class **MyThread**(Thread):

#OVERRIDING THE METHOD run() WHICH IS PRESENT ON THREAD CLASS

def **run**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i)

i+=1

# CREATING AN INSTANCE OF THE MyThread()

th = MyThread()

#CHECK FOR THREADS

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

# INSTEAD OF CALLING THE run(), WE HAVE TO INVOKE THE START METHOD, WHICH IS AVAILABLE IN THREAD CLASS

# INTERNALLY IT WILL SPAWN OFF A NEW THREAD AND IVOKE THE RUN METHOD THAT WE HAVE OVERRIDDEN.

th.start()

**Output:**

Current Thread which is running : MainThread

Current Thread which is running : Thread-1

0

1

2

3

4

5

**MULTIPLE THREAD – CLASS WHICH DOESNOT INHERIT THE THREAD CLASS**

**Code:**

from threading import Thread

import threading

class **MyThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i)

i+=1

# CREATE AN OBJECT OF CLASS MyThread

thObj = MyThread()

# CREATE A THREAD OBJECT

th = Thread(target=thObj.displayNumber)

#CHECK FOR THREADS

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

#START THE TREAD

th.start()

**Output:**

Current Thread which is running : MainThread

Current Thread which is running : Thread-1 (displayNumber)

0

1

2

3

4

5

**MULTI THREADING**

**Code:**

from threading import Thread

import threading

class **FirstThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i)

i+=1

class **SecondThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i\*2)

i+=1

class **ThirdThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

while(i<=5):

print(i-2)

i+=1

# CREATE AN OBJECT OF CLASS MyThread

thObj1 = FirstThread()

# CREATE A THREAD OBJECT

th1 = Thread(target=thObj1.displayNumber)

#START THE TREAD

th1.start()

# CREATE AN OBJECT OF CLASS MyThread

thObj2 = SecondThread()

# CREATE A THREAD OBJECT

th2 = Thread(target=thObj2.displayNumber)

#START THE TREAD

th2.start()

# CREATE AN OBJECT OF CLASS MyThread

thObj3 = ThirdThread()

# CREATE A THREAD OBJECT

th3 = Thread(target=thObj3.displayNumber)

#START THE TREAD

th3.start()

**Output:**

Current Thread which is running : Thread-1 (displayNumber)

0

1

2

3

4

5

Current Thread which is running : Thread-2 (displayNumber)

0

2

4

6

8

Current Thread which is running : 10

Thread-3 (displayNumber)

-2

-1

0

1

2

3

**MULTI THREADING WITH SLEEP**

**Code:**

from threading import Thread

from time import sleep

import threading

import datetime

class **FirstThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

dt = datetime.datetime.today()

print(*"FirstThread processing before Sleep : "*, dt)

print()

sleep(15)

dt = datetime.datetime.today()

print(*"FirstThread processing after Sleep : "*, dt)

print()

while(i<=5):

print(i)

i+=1

class **SecondThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

dt = datetime.datetime.today()

print(*"SecondThread processing before Sleep : "*, dt)

print()

sleep(15)

dt = datetime.datetime.today()

print(*"SecondThread processing after Sleep : "*, dt)

print()

while(i<=5):

print(i\*2)

i+=1

class **ThirdThread**:

def **displayNumber**(*self*):

i=0

if(threading.current\_thread() == threading.main\_thread()):

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

else:

tName1 = threading.current\_thread().name

print(*"Current Thread which is running : "*,tName1)

dt = datetime.datetime.today()

print(*"ThirdThread processing before Sleep : "*, dt)

print()

sleep(15)

dt = datetime.datetime.today()

print(*"ThirdThread processing after Sleep : "*, dt)

print()

while(i<=5):

print(i-2)

i+=1

# CREATE AN OBJECT OF CLASS MyThread

thObj1 = FirstThread()

# CREATE A THREAD OBJECT

th1 = Thread(target=thObj1.displayNumber)

#START THE TREAD

th1.start()

# CREATE AN OBJECT OF CLASS MyThread

thObj2 = SecondThread()

# CREATE A THREAD OBJECT

th2 = Thread(target=thObj2.displayNumber)

#START THE TREAD

th2.start()

# CREATE AN OBJECT OF CLASS MyThread

thObj3 = ThirdThread()

# CREATE A THREAD OBJECT

th3 = Thread(target=thObj3.displayNumber)

#START THE TREAD

th3.start()

**Output:**

Current Thread which is running : Thread-1 (displayNumber)

FirstThread processing before Sleep : Current Thread which is running : Thread-2 (displayNumber)

SecondThread processing before Sleep : 2024-04-01 16:18:14.952051

Current Thread which is running : 2024-04-01 16:18:14.952051

Thread-3 (displayNumber)

ThirdThread processing before Sleep : 2024-04-01 16:18:14.953051

SecondThread processing after Sleep : 2024-04-01 16:18:29.953401

0

2FirstThread processing after Sleep : ThirdThread processing after Sleep :

4

2024-04-01 16:18:29.953401

0

1

2

3

4 6

8

2024-04-01 16:18:29.953401

-2

105

-1

0

1

2

3