```
import threading

# Return CurrentThread
t = threading.currentThread().getName()
print(t) # MainThread

# Return the main thread object
t = threading.main_thread().getName()
print(t) # MainThread

#check MainThread == currentThread
if threading.currentThread() == threading.main_thread():
    print("Main Thread Executed") # Main Thread Executed
else:
    print("Some other thread")

MainThread
MainThread
Main Thread Executed
```

```
import threading

t1 = threading.Thread()
print(t1.name) # Thread-1
t1.name = "New Thread 01"
print(t1.name)  # New Thread 01

t2 = threading.Thread()
print(t2.name) # Thread-2
t2.name = "New Thread 02"
print(t2.name)  # New Thread 02
```

```
import threading

t1 = threading.currentThread()
t2 = threading.main_thread()
t3 = None
if t1 == t2:
   print(True)
   print(t1.ident) # 17448
   print(t2.ident) # 17448
else:
   print(False)
```

```
from threading import Thread

# target the object
def displayNumbers():
   i = 0
   while(i<=10):
      print(i, end= " ")
      i=i+1 # or i+=1

t = Thread(target=displayNumbers) # target = object
t.start()
```

```
0 1 2 3 4 5 6 7 8 9 10
```

target is the callable object to be invoked by the run() method. Defaults to None, meaning nothing is called .

```
from threading import Thread

# Invoke run method
class MyThread(Thread):
    def run(self): # Overrides method in Thread
        i = 0
        while(i<=10):
            print(i, end = " ") # 0 1 2 3 4 5 6 7 8 9 10
            i+=1 # or i=i+1

t = MyThread()
t.start()

0 1 2 3 4 5 6 7 8 9 10
```

```python
from threading import Thread, currentThread

# Set Thread Name
def d1():
    print('Default Thread: ', currentThread().getName())
    currentThread().setName('New Thread')
    print('New Added Thread Name: ', currentThread().getName())

t1 = Thread(target=d1)
t1.start()

t2 = Thread(target=d1)
t2.start()
```

```
Default Thread:  Thread-1
Default Thread:  Thread-2
New Added Thread Name:  New Added Thread Name: New Thread
New Thread

Default Thread:  Thread-1
New Added Thread Name:  Default Thread: New Thread
Thread-2
New Added Thread Name:  New Thread

Default Thread:  Thread-1
New Added Thread Name:  New Thread
Default Thread:  Thread-2
New Added Thread Name:  New Thread
```

```
from threading import Thread

# Creating a Thread by Extending a Thread Class
class MyThread(Thread):

   def __init__(self):
      Thread.__init__(self)

   def run(self): # Overrides method in Thread
      print(self.getName()) # Thread-1

t = MyThread()
t.start()
```

```
from threading import Thread

# A class that extends the Thread class
class MyThread(Thread):
   def __init__(self, eid, ename):
      # Call the Thread class's init function
      Thread.__init__(self)
      self.eid = eid
      self.ename = ename

   def run(self): # Overrides method in Thread
      print(self.eid, self.ename)

t1 = MyThread(101, "NameOne")
t2 = MyThread(102, "NameTwo")
t1.start()
t2.start()

101 NameOne
102 NameTwo
```

```
import threading
from threading import Thread
import time

# Comment join and try the example
# Creating a Thread by Extending a Thread Class
# join() allows one thread to wait until another thread completes its execution
class MyThread(Thread):

    def run(self): # Overrides method in Thread
        for i in range(5):
            time.sleep(1)
            print(i, "Child Thread")

t1 = MyThread()
t1.start()
# t1.join()
th = threading.current_thread().getName()
print('Current Thread: ', th)  # Current Thread:  MainThread

t2 = MyThread()
t2.start()
# t2.join()
th = threading.current_thread().getName()
print('Current Thread: ', th)  # Current Thread:  MainThread
```

```
Current Thread:  MainThread
Current Thread:  MainThread
0 Child Thread
0 Child Thread
11  Child ThreadChild Thread

22  Child ThreadChild Thread

33 Child Thread Child Thread

4 4Child Thread
Child Thread
```

```
import threading
from threading import Thread
import time

# Comment join and try the example
# Creating a Thread by Extending a Thread Class
# join() allows one thread to wait until another thread completes its execution
class MyThread(Thread):
    def run(self): # Overrides method in Thread
        for i in range(5):
            time.sleep(1)
            print(i, "Child Thread")

t1 = MyThread()
t1.start()
t1.join()
th = threading.current_thread().getName()
print('Current Thread: ', th)  # Current Thread:  MainThread

t2 = MyThread()
t2.start()
t2.join()
th = threading.current_thread().getName()
print('Current Thread: ', th)  # Current Thread:  MainThread
```

```
0 Child Thread
1 Child Thread
2 Child Thread
3 Child Thread
4 Child Thread
Current Thread:  MainThread
0 Child Thread
1 Child Thread
2 Child Thread
3 Child Thread
4 Child Thread
Current Thread:  MainThread
```

```
from threading import Thread, currentThread

# Without extending thread class
class MyThread:

    def d1(self):
        for i in range(5):
            print(i, ": Thread : ", currentThread().getName())

t = MyThread()
t1 = Thread(target=t.d1)
t1.start()
t1.join()

t2 = Thread(target=t.d1)
t2.start()
t2.join()

0 : Thread :  Thread-1
1 : Thread :  Thread-1
2 : Thread :  Thread-1
3 : Thread :  Thread-1
4 : Thread :  Thread-1
0 : Thread :  Thread-2
1 : Thread :  Thread-2
2 : Thread :  Thread-2
3 : Thread :  Thread-2
4 : Thread :  Thread-2
```

```
from threading import Thread, currentThread

# Naming a Thread in Python
def d1():
   print('Default Thread: ', currentThread().getName())

   # object.setName(<name>) : To set our own name
   currentThread().setName('Thread d1')

   # object.getName() – Returns Name of Thread
   print('New Added Thread Name: ', currentThread().getName()) # Thread d1

t1 = Thread(target=d1)
t1.start()

t2 = Thread(target=d1)
t2.start()
```

```
Default Thread:  Thread-1
New Added Thread Name:  Default Thread:  Thread-2
New Added Thread Name:  Thread d1
Thread d1
```

```
Default Thread:  Thread-1
New Added Thread Name:  Thread d1
Default Thread:  Thread-2
New Added Thread Name:  Thread d1
```

```
import time
from threading import *

l = [10, 20, 30, 40, 50]
s = {100, 200, 300, 400, 500}
t = (60,70,80,90,100)

def d1(l):
    for i in l:
        time.sleep(1)
        print("List Elements:", i)

def d2(s):
    for i in s:
        time.sleep(1)
        print("Set Elements :", i)

def d3(t):
    for i in t:
        time.sleep(1)
        print("Tuple Elements :", i)

t1 = Thread(target=d1, args=(l,))
t1.start()
t1.join()

t2 = Thread(target=d2, args=(s,))
t2.start()
t2.join()

t3 = Thread(target=d3, args=(t,))
t3.start()
t3.join()
```

```
List Elements: 10
List Elements: 20
List Elements: 30
List Elements: 40
List Elements: 50
Set Elements : 400
Set Elements : 100
Set Elements : 500
Set Elements : 200
Set Elements : 300
Tuple Elements : 60
Tuple Elements : 70
Tuple Elements : 80
Tuple Elements : 90
Tuple Elements : 100
```