In Python, the threading.Thread class has several important **methods** that are used to control the execution and management of threads

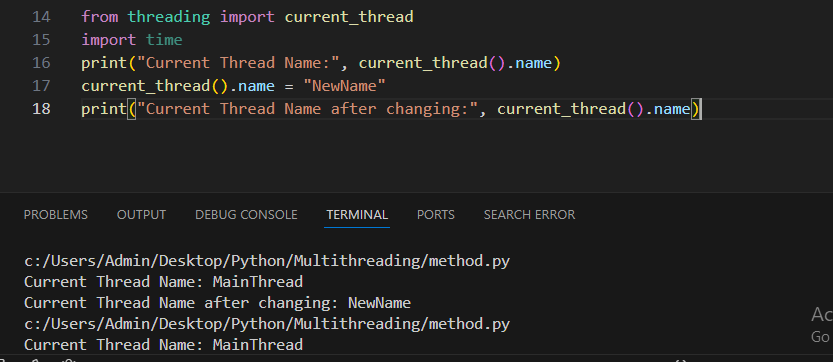
1. **setName(name) and getName():**

* **Definition**:
  + setName(name): Assigns a name to the thread. This is useful for identifying threads, especially in debugging or logging.
  + getName(): Returns the name of the thread.
* **Syntax**:

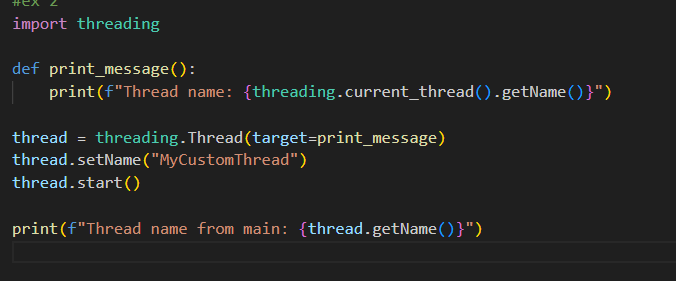
thread.setName(name)

thread.getName()

Example1:



Ex2:



Output:

Thread name: MyCustomThread

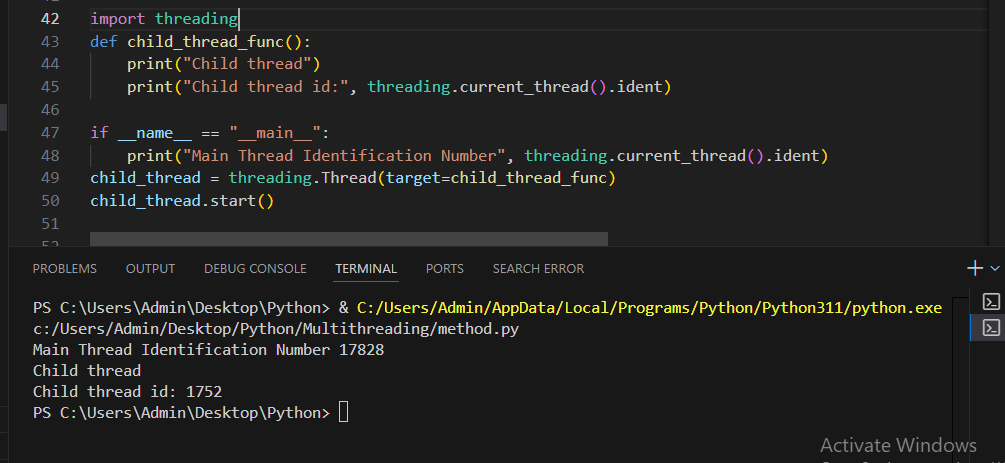
Thread name from main: MyCustomThread

2. Ident(Property):

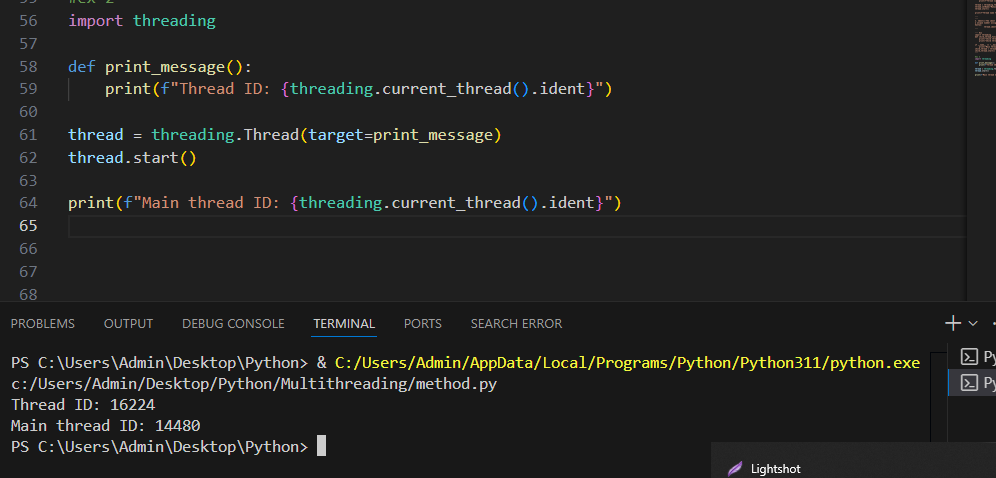
* **Definition**: The ident property returns the thread’s identifier, a unique number assigned by the operating system. This can be useful for tracking or debugging threads.
* **Syntax**:

thread.ident

Example1:



Example2:



3.Active-count():

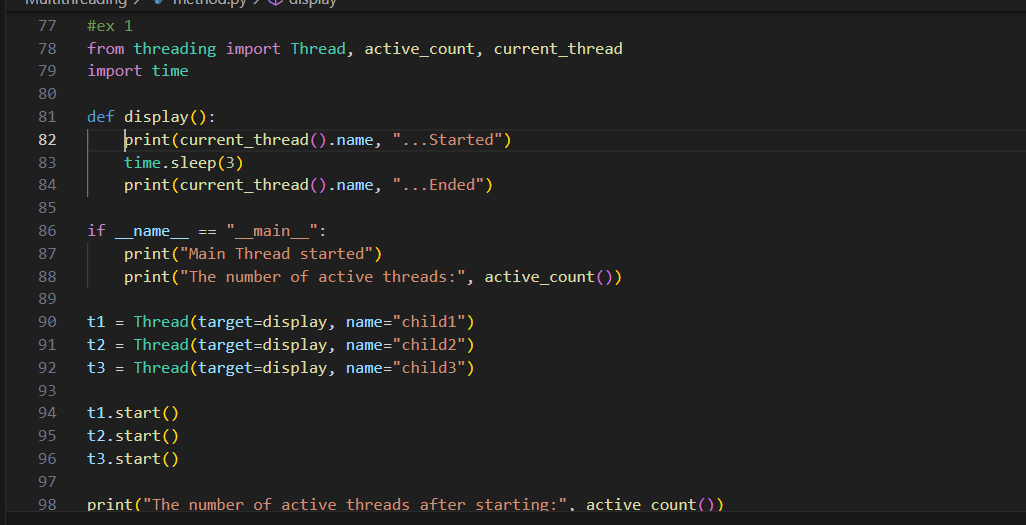
Description:

threading.active\_count() is a function that returns the number of **currently active threads** in the Python interpreter. This includes both the main thread and all currently running threads, except for terminated threads and threads that have not yet been started.

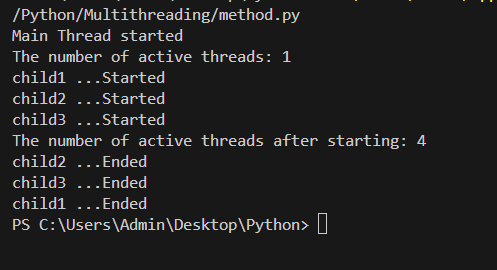
Syntax:

threading.active\_count()

Example1:



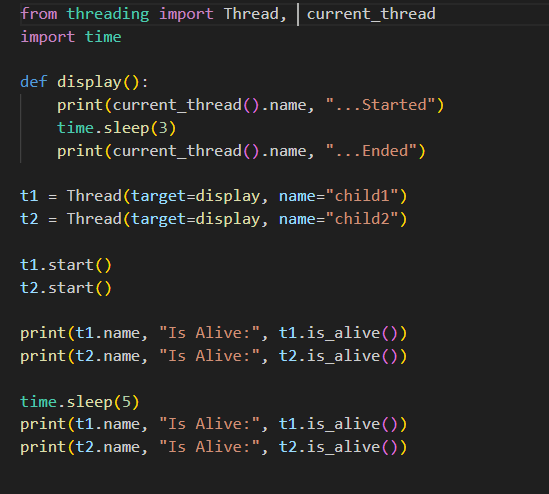
Output:



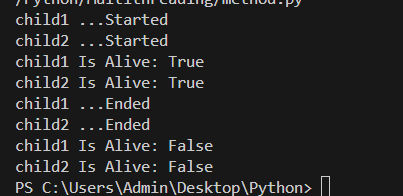
4. is\_alive():

* **Definition**: Returns True if the thread is still alive (running or ready to run), and False if it has finished its execution.
* **Syntax**:
* thread.is\_alive()

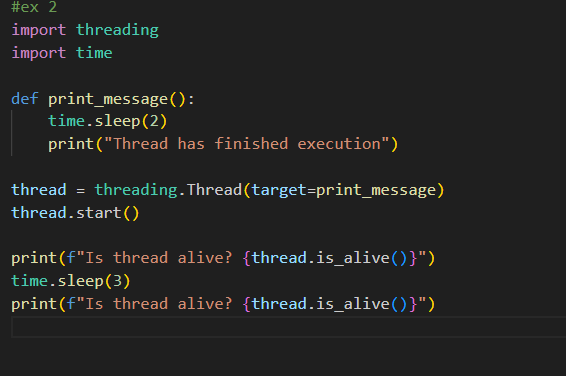
Example1:



Output:



Example 2:



Output:

