[library set up]

Import threading

[create a thread object]

* Way 1 (create directly):

a = threading.Thread(target=<function>, [args=<argument>, name=<name>, daemon=None])

* Way 2 (create by define a class first):

class <class\_name>(threading.Thread):

def run(self):

“”” what the thread does “””

a = <class\_name>()

[identity]

* The variable of thread object calling by <thread\_object>.ident
* It is None before the thread start.
* The identity remain the same after the thread die,
* The identity can be reused by other thread, if one thread die, and the other thread start sequentialy. (it is possible that two threads have same ID but it is impossible for two alive thread share the same ID)

[daemon thread]

Deamon thread is the thread kind that when there is no other non-daemon thread alive, the program will exist. (daemon thread can be terminated when there is no other non-daemon thread) (daemon thread is meaningless without non-daemon thread)

* Main thread is non-daemon thread
* Can be set during the thread object construction

(the default value of daemon=None, which means it inherit the daemon flag from the thread that call the constructor)

[useful function]

threading.active\_count() >> return the number of active thread

threading.enumerate() >> return the thread list

threading.current\_thread() >> return the current thread

<thread\_object>.is\_alive() >> return true if it not finish, false if it finish

[reference]

<https://www.youtube.com/playlist?list=PL1H1sBF1VAKVMONJWJkmUh6_p8g4F2oy1>