Потік – це найменша одиниця виконання з незалежним набором інструкцій. Він є частиною процесу і працює в таких же виконуваних ресурсах програми, як пам'ять. Потік має початкову точку, послідовність виконання та результат. Він має вказівник інструкцій, який зберігає поточний стан потоку і контролює, що буде виконано в наступним. Здатність процесу виконувати кілька потоків паралельно називається багатопоточність.

Оте, багатопоточність в Python реалізується за допомогою двох модулів: thread і threading.

Використовуючи модуль thread для створення потоку нам потрібно прописати наступний метод:

thread.start\_new\_thread (function, args [, kwargs]). Цей метод запускає новий потік і повертає його ідентифікатор. Він виклкає функцію, зазначену в якості параметра «function» з переданим списком аргументів. Тут args - це кортеж аргументів; використавши порожній кортеж, ми викличемо метод без будь-яких аргументів.

Модуль threading об'єднує всі методи модуля thread і надає кілька додаткових методів.

* threading.activeCount (): знаходить загальне число активних об'єктів потоку.
* threading.currentThread (): його можна використовувати для визначення кількості об'єктів потоку в елементі управління потоком зухвалої сторони.
* threading.enumerate (): надає повний список об'єктів потоку, які в даний момент активні.

Лябмда-вирази - це особливий синтаксис в Python, необхідний для створення анонімних функцій.

Наприклад, функція, яка повертає свій аргумент, виражається стандартним визначенням функції Python з використанням ключового слова def наступним чином:

>>> def identity (x):

... return x

identity () приймає аргумент x і повертає його при виклику.

Якщо ми використаємо лямбда-конструкцією, отримаємо:

>>> lambda x: x

Застосуємо вказану вище функцію до аргументу, записавши функцію і її аргумент в круглі дужки:

>>> (lambda x: x + 1) (2) (в дужках аргумент двійка)

3 (відповідь)

Оскільки лямбда-функція є виразом, вона може мати ім’я:

>>> add\_one = lambda x: x + 1

>>> add\_one (2)

3