

## What is a Loop in Python?

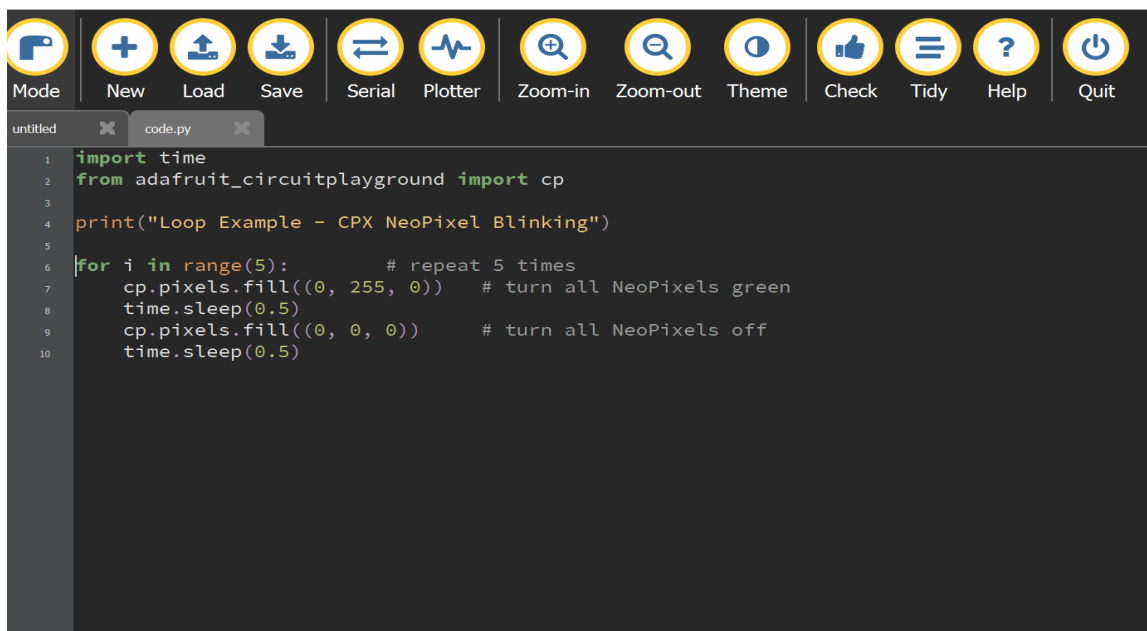
### Simple meaning:

A **loop** is used when you want to **repeat** something **again and again** automatically — instead of writing the same line many times.

### How it works in Python:

- Python checks the loop condition.
- If it's true, it runs the code inside the loop.
- Then it goes back to the beginning and repeats the steps.
- When the condition becomes false (or items are finished), the loop stops.

### Example Using CPX NeoPixel Lights

A screenshot of a code editor interface. The top toolbar contains icons for Mode, New, Load, Save, Serial, Plotter, Zoom-in, Zoom-out, Theme, Check, Tidy, Help, and Quit. Below the toolbar, there are two tabs: 'untitled' and 'code.py'. The 'code.py' tab is active, showing a Python script. The script is as follows:

```
1 import time
2 from adafruit_circuitplayground import cp
3
4 print("Loop Example - CPX NeoPixel Blinking")
5
6 for i in range(5):          # repeat 5 times
7     cp.pixels.fill((0, 255, 0)) # turn all NeoPixels green
8     time.sleep(0.5)
9     cp.pixels.fill((0, 0, 0))   # turn all NeoPixels off
10    time.sleep(0.5)
```

## Step-by-Step Explanation

### Step 1:

```
for i in range(5):
```

This means “repeat the inside lines **5 times**.”

The variable `i` changes each time (0, 1, 2, 3, 4).

### Step 2:

```
cp.pixels.fill((0, 255, 0))
```

Lights all 10 CPX NeoPixels in **green** (R, G, B = 0, 255, 0).

**Step 3:**

```
time.sleep(0.5)
```

Wait for half a second — the light stays ON briefly.

**Step 4:**

```
cp.pixels.fill((0, 0, 0))
```

Turns all NeoPixels **off**.

**Step 5:**

```
time.sleep(0.5)
```

Waits again before the next loop cycle starts.

**Result**

The lights **blink ON and OFF in green color five times** using a simple `for` loop.