Introduction to Arrays using NeoPixels (CPX)

What is an Array?

- An **array** is a collection of similar data values stored together in one place.
- It helps store **multiple items under a single name** instead of using many variables.
- Arrays make it easy to manage and access data using an **index number**.

Array in Python

- In Python, arrays are represented using **lists**.
- Lists are created using **square brackets [ ]**.
- Each item can be accessed by its **index**, starting from `0`.

Example:

```
numbers = [10, 20, 30, 40, 50]
print(numbers[0])   # Output: 10
print(numbers[3])   # Output: 40
```

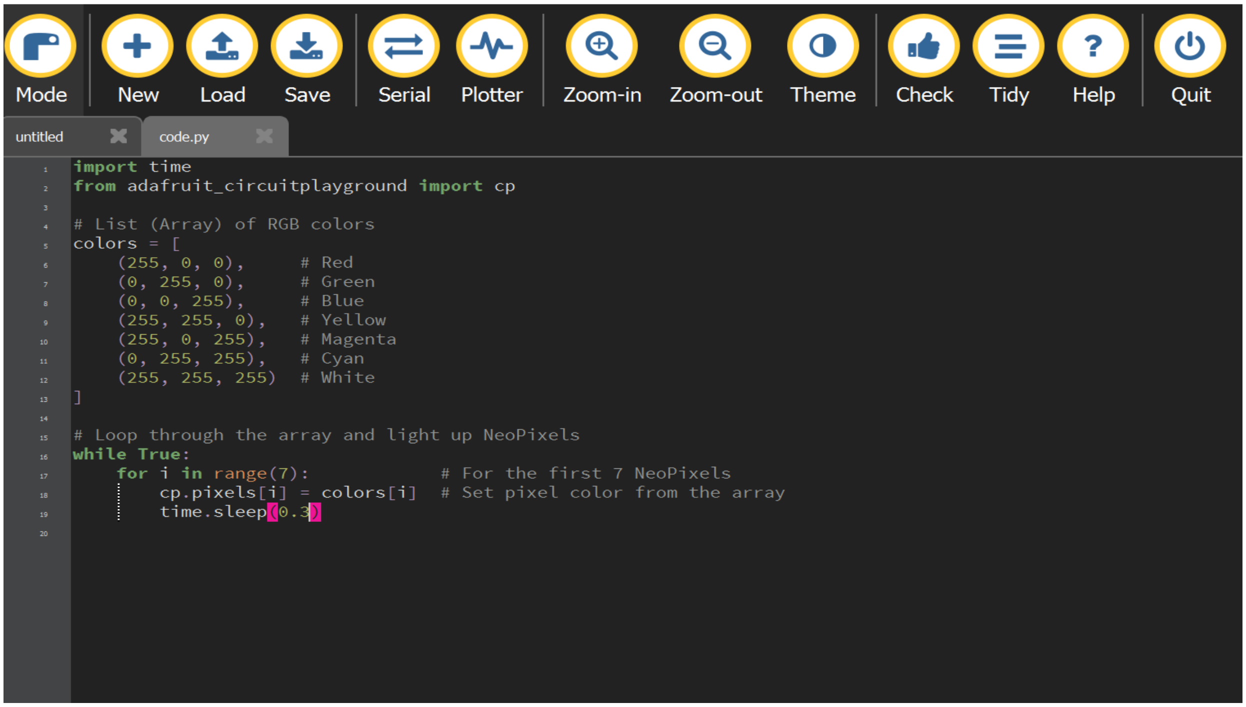
Why Arrays are Useful

- Easy to store and use multiple values.
- Reduces code length.
- Simple to update or loop through values.
- Perfect for controlling **multiple NeoPixels** on CPX.

Using Arrays with NeoPixels (CPX)

We can store color values in an array (list) and use them to light up NeoPixels.

Example Code:



```
import time
from adafruit_circuitplayground import cp

# List (Array) of RGB colors
colors = [
    (255, 0, 0),       # Red
    (0, 255, 0),       # Green
    (0, 0, 255),       # Blue
    (255, 255, 0),     # Yellow
    (255, 0, 255),     # Magenta
    (0, 255, 255),     # Cyan
    (255, 255, 255)    # White
]

# Loop through the array and light up NeoPixels
while True:
    for i in range(7):              # For the first 7 NeoPixels
        cp.pixels[i] = colors[i]    # Set pixel color from the array
        time.sleep(0.3)
```

Step-by-Step Explanation

1. Importing Required Modules

```
import time
from adafruit_circuitplayground import cp
```

- `time` — used for creating small delays between lighting each pixel.
- `cp` — gives access to the Circuit Playground Express board functions like LEDs (NeoPixels), sensors, etc.

2. Creating an Array of Colors

```
colors = [
    (255, 0, 0),      # Red
    (0, 255, 0),      # Green
    (0, 0, 255),      # Blue
    (255, 255, 0),    # Yellow
    (255, 0, 255),    # Magenta
    (0, 255, 255),    # Cyan
    (255, 255, 255)   # White
]
```

- Here, `colors` is a **list (array)** that stores **7 RGB color values**.
- Each color is represented as an **RGB tuple** — `(R, G, B)`.
    - Example: `(255, 0, 0)` means **Red** (full red light, no green, no blue).

- So, instead of writing 7 separate color variables, we use one array.

3. Starting an Infinite Loop

```
while True:
```

- This means the code will **run forever** — continuously lighting up pixels.
- Used so the colors keep showing in a loop.

4. Looping Through NeoPixels

```
for i in range(7):
```

- `range(7)` means `i` will go from 0 to 6 (7 numbers total).
- It controls the **first 7 NeoPixels** on the CPX board.

5. Setting Pixel Colors

```
cp.pixels[i] = colors[i]
```

- `cp.pixels[i]` selects the NeoPixel number **i**.
- `colors[i]` picks the matching color from the array.
- So:
  - Pixel 0 → Red
  - Pixel 1 → Green
  - Pixel 2 → Blue
  - Pixel 3 → Yellow
  - Pixel 4 → Magenta
  - Pixel 5 → Cyan
  - Pixel 6 → White

6. Adding Delay

```
time.sleep(0.3)
```

- Waits for **0.3 seconds** before lighting the next pixel.
- Helps us see each light turning on one by one slowly.

Output

- Each NeoPixel lights up **sequentially** in a new color.
- It looks like a rainbow effect — one pixel after another glowing beautifully.

- The loop continues forever until stopped manually.