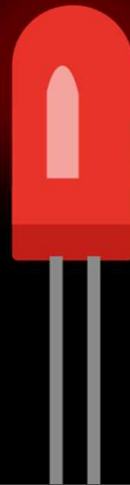


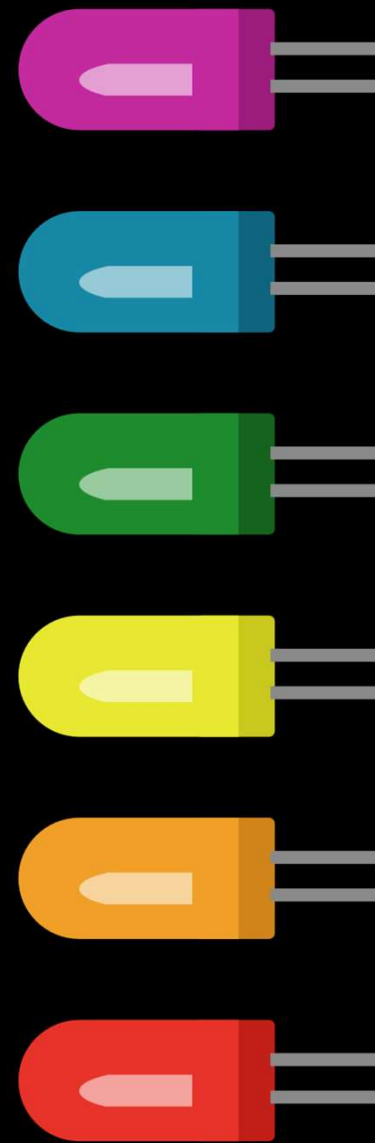
LED There Be Light!

Andres D. Flores | 2026



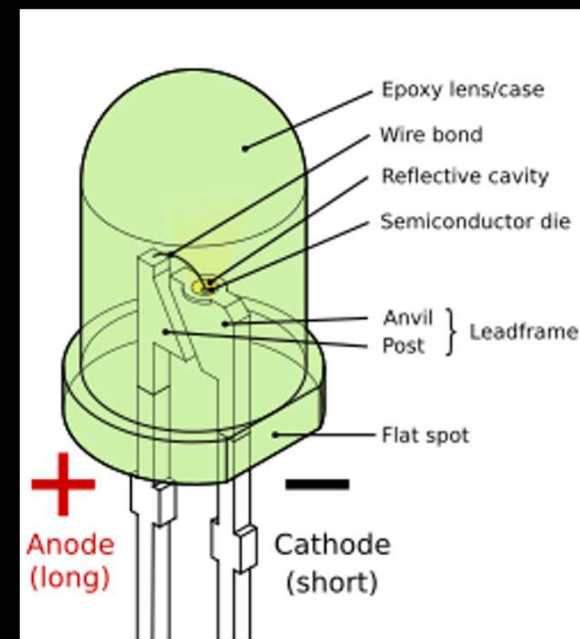
What is an LED?

LED There Be Light!



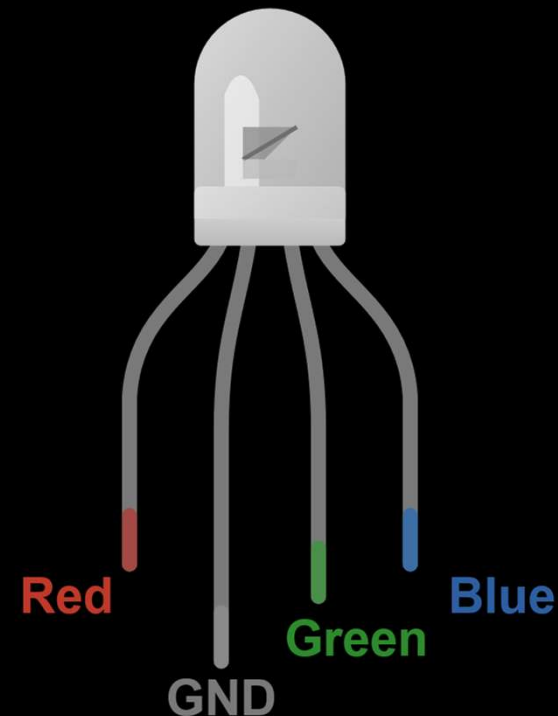
LED: Light Emitting Diode

- Diode: Two-terminal electronic component that acts as a one-way valve for electrical current
 - Current flows in through anode (long leg), out through cathode (short leg) to illuminate
- LED: Emits light when an electric current passes through it in the forward direction
 - Transducer of electricity to light
 - Transducers convert energy from one form to another
- Advantages:
 - Low-voltage, lower power consumption
 - Reduced waste heat
 - Longer lifetime



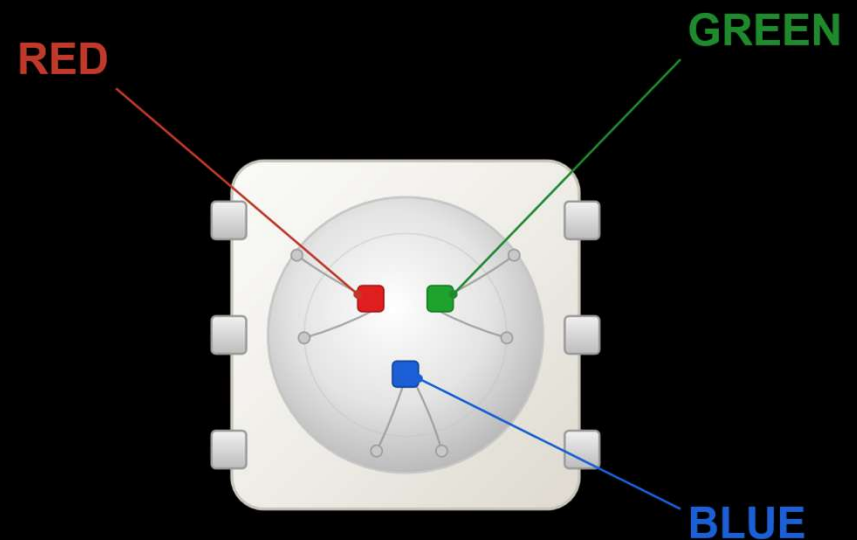
LED: Light Emitting Diode

- An **RGB** LED combines a **red**, **green**, and **blue** LEDs into a single package to produce over 16 million different color hues
- Each LED is individually defined 0-255, and the combination of the RGB definitions yields a desired color
 - Desired Color: **Magenta**
 - **Red**: 255
 - **Green**: 0
 - **Blue**: 255



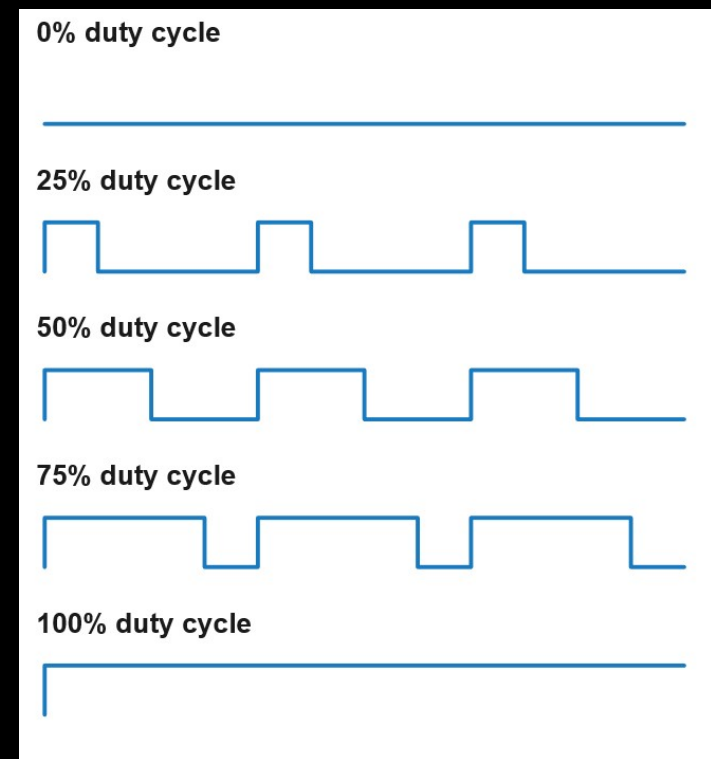
LED: Light Emitting Diode

- Each unit (pixel) has three LEDs(one red, one green, and one blue) and a driver to control the color and the luminosity of the led.
- Each LED is individually defined with an integer from 0-255, where 0 is off and 255 is maximum brightness.
- A controller unit in the LED applies the appropriate value to each LED, and the composite of all 3 LEDs yields the desired color.



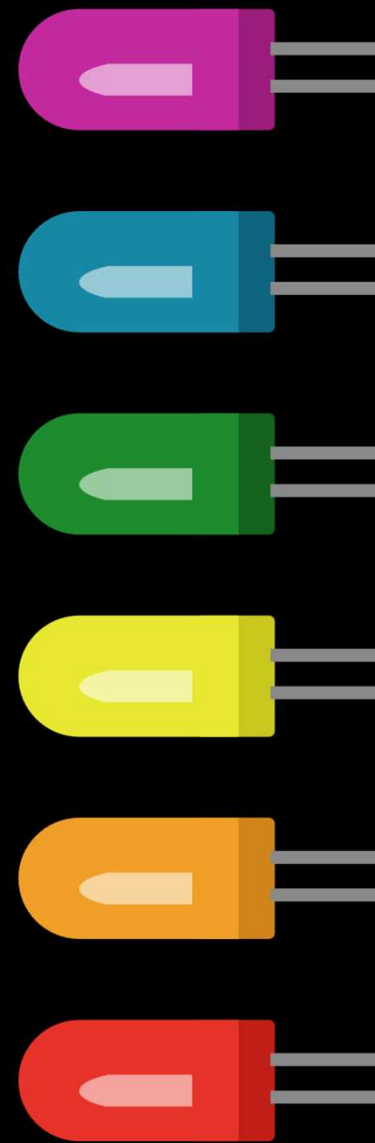
Brightness: Pulse-Width Modulation (PWM)

- Varies how much time a digital signal is “high” in to mimic an analog signal
- Signal can only be “high” (input voltage) or “low” (0V) at any time, but the proportion of time that the signal is “high” compared to when it is “low” over a time interval.
- **Duty Cycle:** Defines the amount of "on time", %
- **Switching Frequency:** The number of times a signal switches between high-low in one second, measured in hertz (Hz)
 - 100Hz (Flicker Threshold): Humans don't perceive flicker
 - 1000Hz (Dimming Resolution): Transitions are sharper

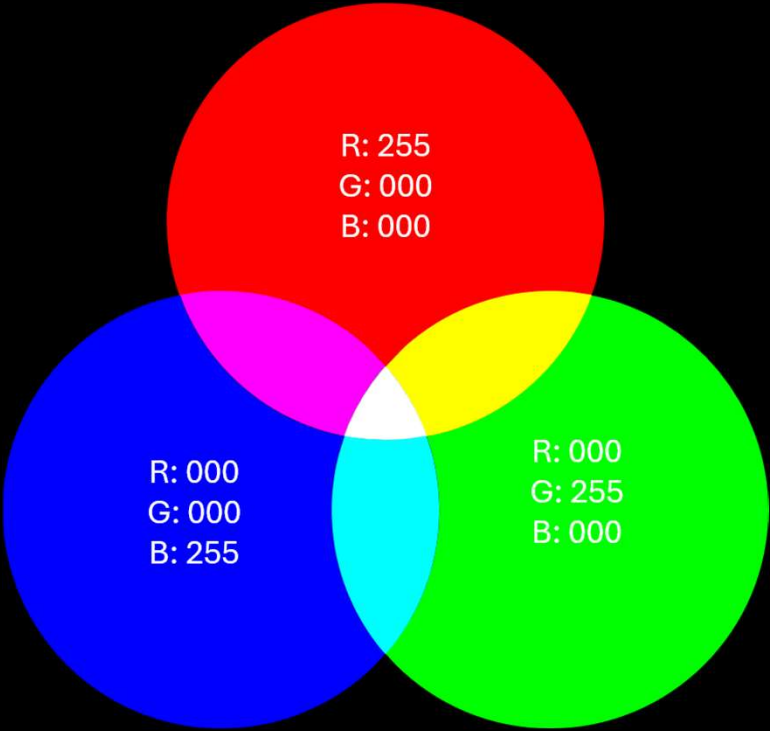


Color Theory

LED There Be Light!



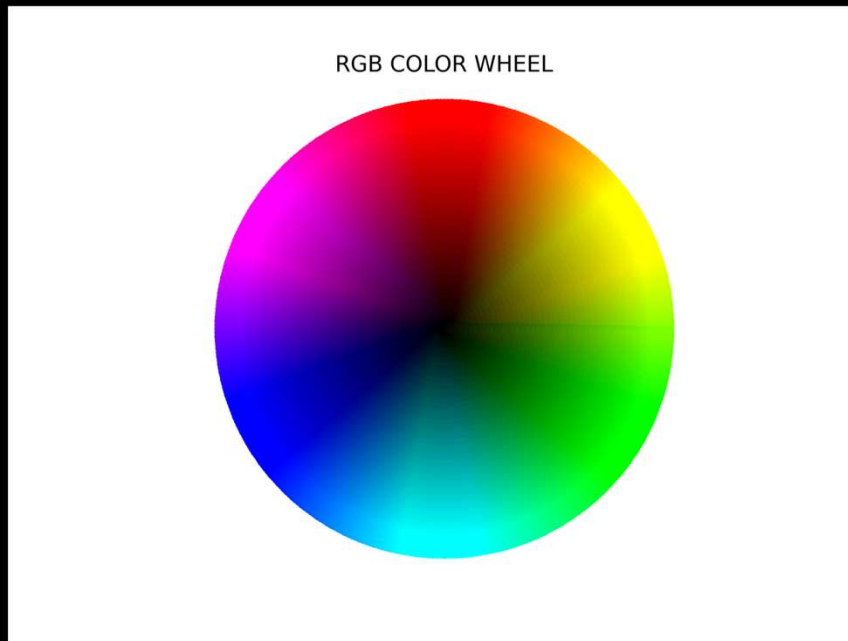
Additive Color Mixing



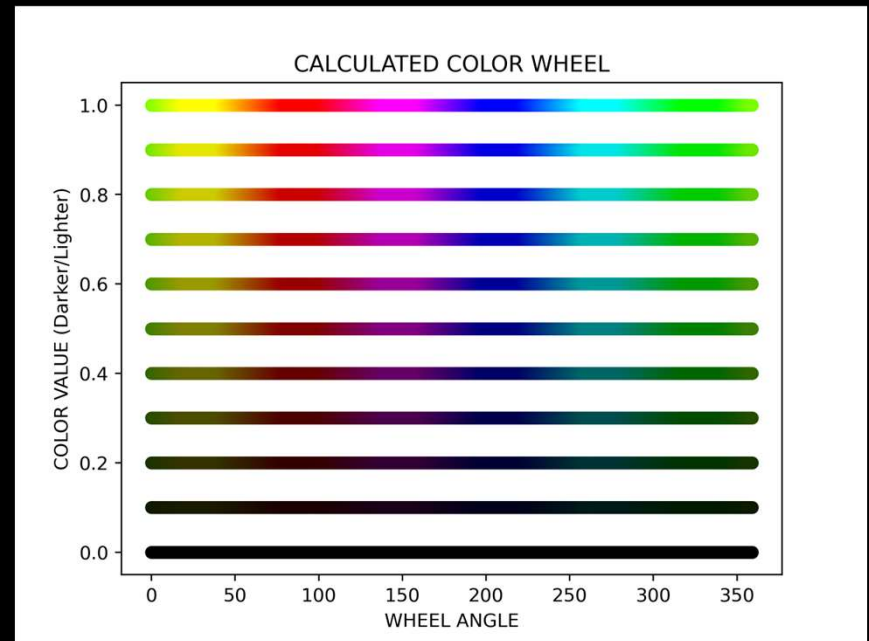
Red	Green	Blue	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
0	0	0	190	190	63	0	255	63	0	125	255	63	0	63
125	0	0	255	255	125	63	255	125	63	125	190	125	0	125
190	0	0	125	125	63	0	190	63	0	63	190	190	0	190
255	0	0	190	190	125	63	190	125	63	125	255	255	0	255
255	63	63	255	255	190	0	255	125	0	63	255	255	63	255
190	63	63	190	255	0	125	255	190	0	0	63	190	63	190
255	125	125	190	255	63	0	125	63	0	0	125	255	125	255
125	63	63	125	190	0	0	190	125	0	0	190	125	63	125
190	125	125	63	125	0	63	255	190	0	0	255	190	125	190
255	190	190	190	255	125	0	255	190	63	63	255	255	190	255
255	63	0	125	255	0	0	63	63	63	63	190	255	0	190
255	125	63	125	190	63	0	125	125	125	125	255	255	63	190
190	63	0	63	190	0	0	190	190	63	63	125	190	0	125
190	125	63	125	255	63	0	255	255	125	125	190	125	0	63
255	125	0	63	255	0	63	255	255	190	190	255	255	125	190
255	190	125	0	63	0	63	190	190	63	0	255	255	0	125
125	63	0	0	125	0	125	255	255	125	63	255	190	63	125
190	125	0	0	190	0	63	125	125	63	0	190	190	0	63
255	190	63	0	255	0	125	190	190	125	63	190	255	63	125
255	190	0	63	255	63	190	255	255	125	0	255	255	0	63
63	63	0	63	190	63	0	190	255	190	125	255	63	63	63
125	125	0	125	255	125	63	190	255	63	0	125	125	125	125
190	190	0	63	125	63	0	125	190	125	0	190	125	125	125
255	255	0	125	190	125	0	63	125	190	63	255	190	190	190
255	255	63	190	255	190	125	190	255	190	0	255	255	255	255

Additive Color Mixing

Color Wheel

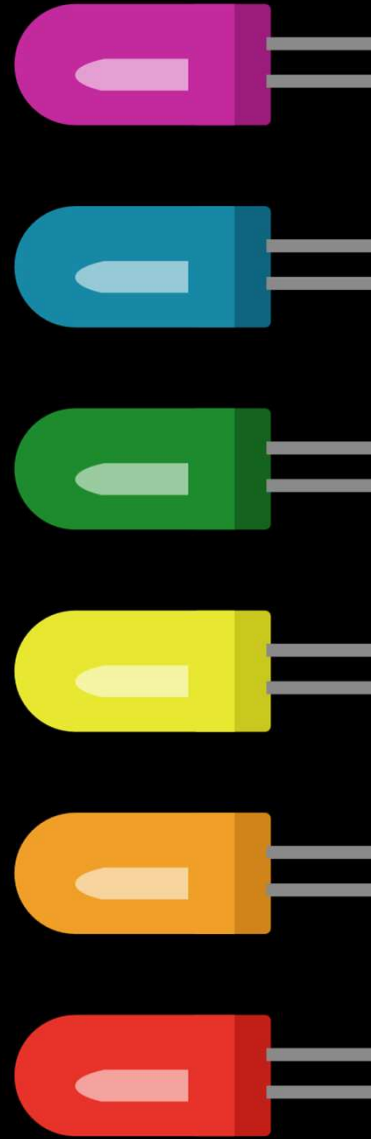


Calculated Color Values



Programming Your LED Strip

LED There Be Light!



Python Data Structures

Lists

An ordered sequence structure for storing a collection of items in a single variable

Syntax: [`<val_0>`, `<val_1>`, ..., `<val_n>`]

Ordered: YES

Mutable: YES

Heterogeneous: YES

Duplicates: YES

Indexable: YES (integer)

Tuples

An ordered sequence structure for storing a collection of items in a single variable

Syntax: (`<val_0>`, `<val_1>`, ..., `<val_n>`)

Ordered: YES

Mutable: NO

Heterogeneous: YES

Duplicates: YES

Indexable: YES (integer)

Python Indexing

- **Index:** An object's numerical position (location) in a data structure
- **Indexing:** Use the object index to access a specific value at a defined position in a data structure
- **Zero-Based Indexing:** Counting begins at 0
- **Forward Indexing:** Access value at the indicated index, counted from the beginning of the structure
 - `<data_structure_obj>[index]`
 - Ex: `<data_structure_obj>[0]` to get the **FIRST** value in a structure
 - Ex: `<data_structure_obj>[4]` to get the **FIFTH** value in a structure
- **Backward Indexing:** Access value at the indicated index, counted from the end of the structure
 - `<data_structure_obj>[-index]`
 - Ex: `<data_structure_obj>[-1]` to get the **LAST** value in a structure
 - Ex: `<data_structure_obj>[-3]` to get the **THIRD FROM LAST** value in a structure

Index number	val_1	val_2	val_3	val_4	val_5	val_6
Forward	0	1	2	3	4	5
Backward	-6	-5	-4	-3	-2	-1

<data_structure>[1] = <data_structure>[-5] = val_2

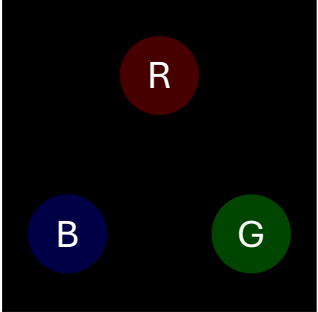
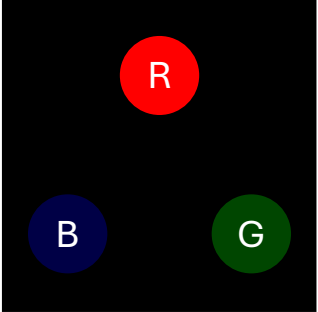
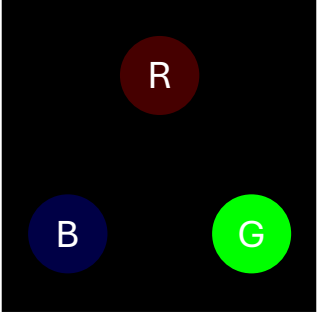
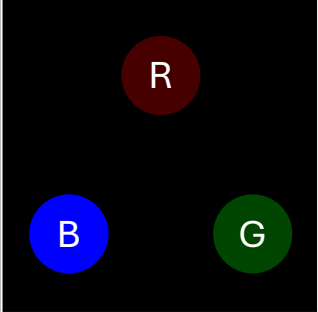
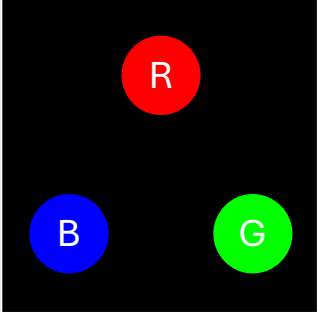
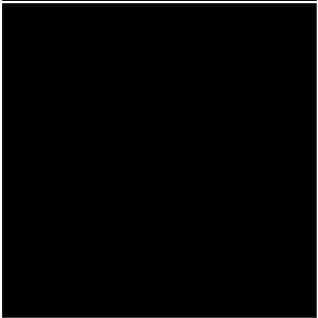
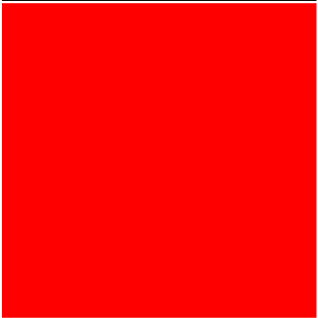
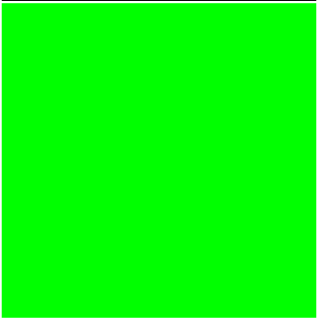
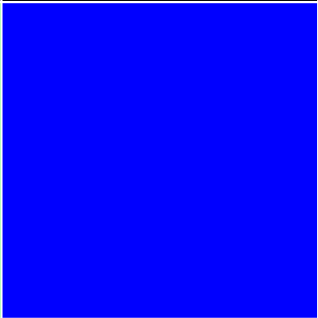
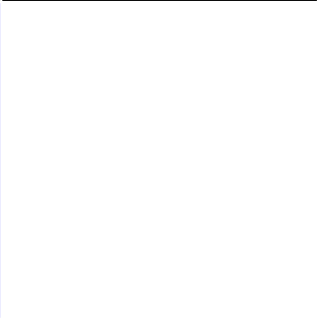
Functions

- A reusable block of code that only runs when it is called and can return objects to the code that calls the function
- Avoids repetitive code and facilitates code maintenance

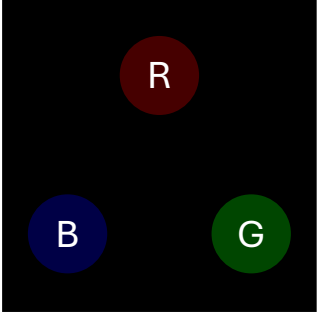
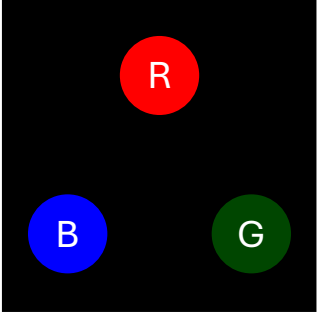
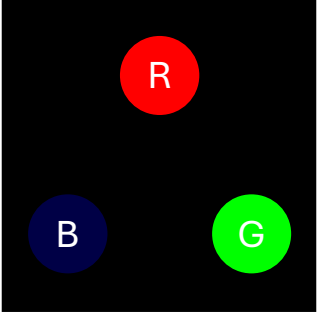
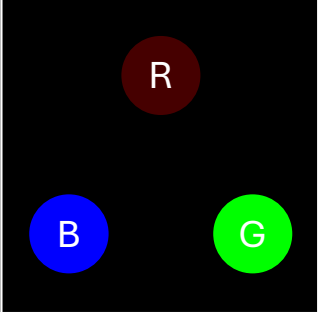
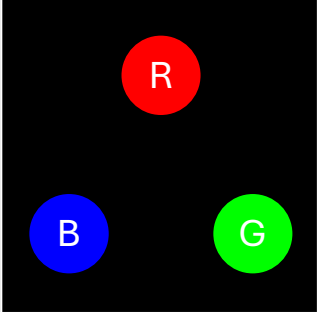
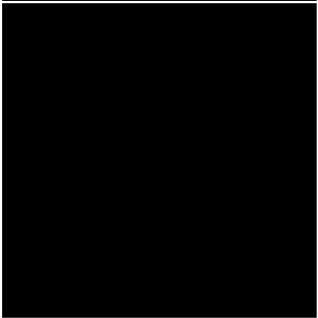
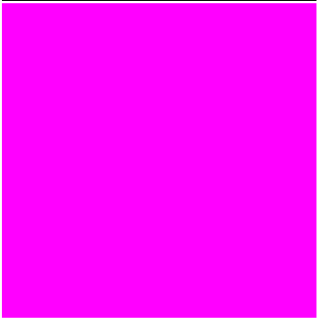
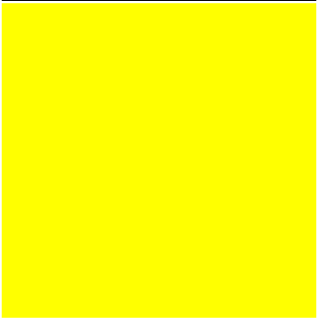
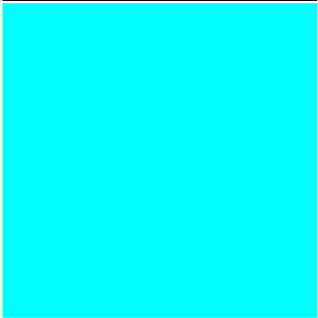
```
def <func_name>(<param>, <param>...):  
    <func_code>  
    return <object>
```

```
<func_name>(<param>, <param>...)
```

Sample LED Strip with 5 RGB LED Units

data = [(0,0,0), (255,0,0), (0,255,0), (0,0,255), (255,255,255)]				
Index = 0	Index = 1	Index = 2	Index = 3	Index = 4
(0,0,0)	(255,0,0)	(0,255,0)	(0,0,255)	(255,255,255)
				
				

Sample LED Strip with 5 RGB LED Units

data = [(0,0,0), (255,0,255), (255,255,0), (0,255,255), (255,255,255)]				
Index = 0	Index = 1	Index = 2	Index = 3	Index = 4
(0,0,0)	(255,0,255)	(255,255,0)	(0,255,255)	(255,255,255)
				
				

Think of an LED strip as a list of tuples

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

What index and RGB do you need to define to achieve this configuration?

Loops

For Loops

- Iteratively execute a block of code *a number of times* or over objects in an iterable structure.

```
for <iterator> in <iterable>:  
    <CODE>
```

While Loops

- Iteratively execute a block of code *indefinitely* until a condition is met (i.e. condition is *True*).

```
while <true_condition>:  
    <CODE>
```

LED Strip Implementation

