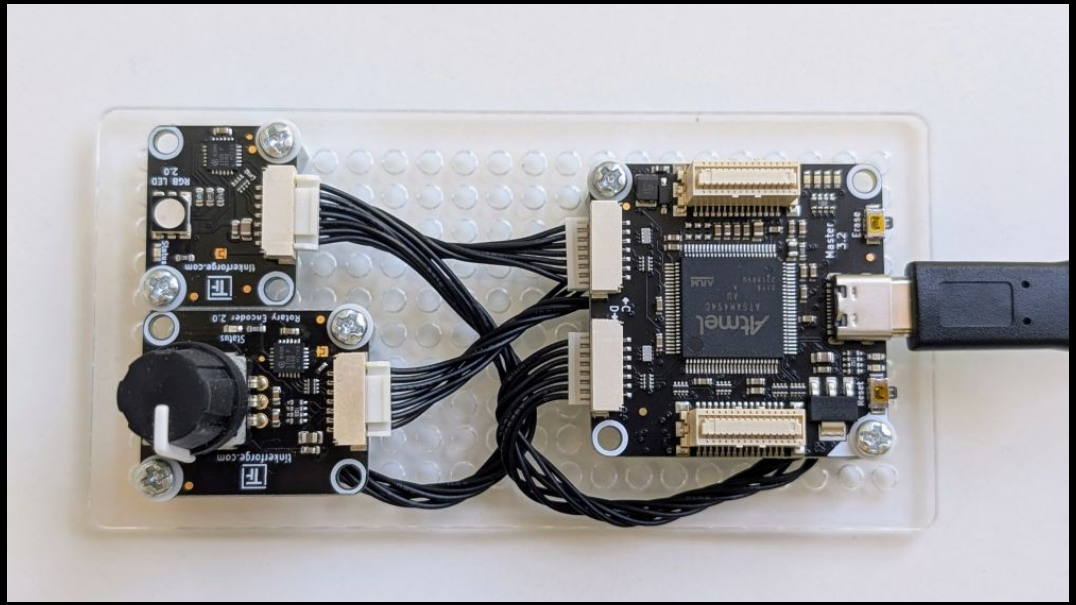
These slides serve as a visual aid for the lecture, not as a comprehensive document or script.

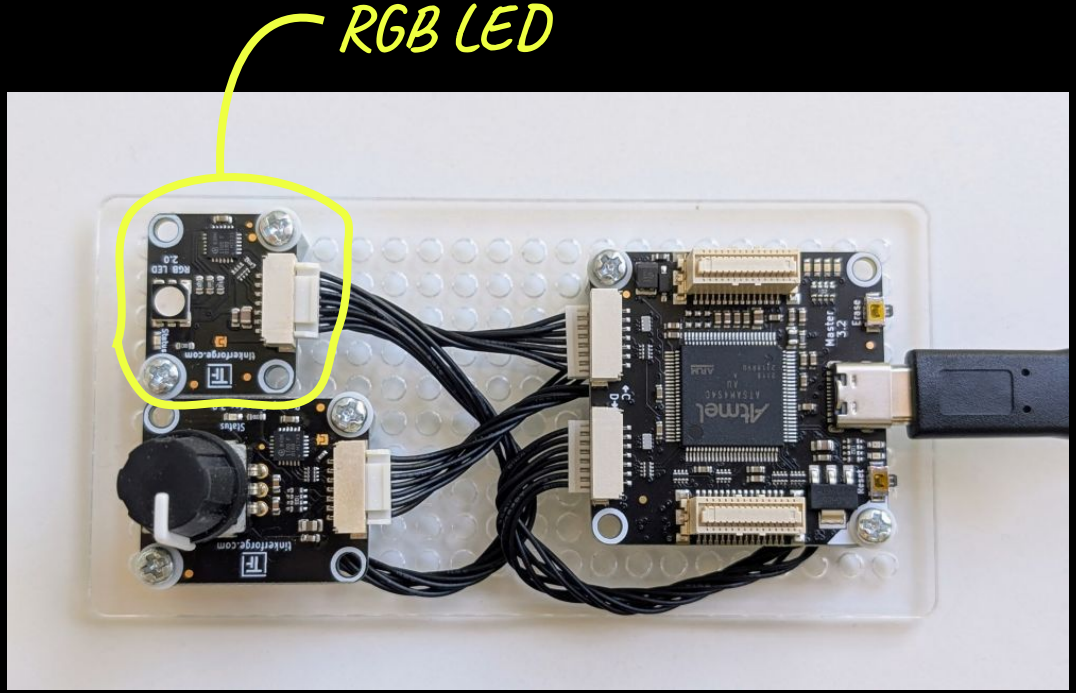
Please refrain from printing these slides to help protect the environment.

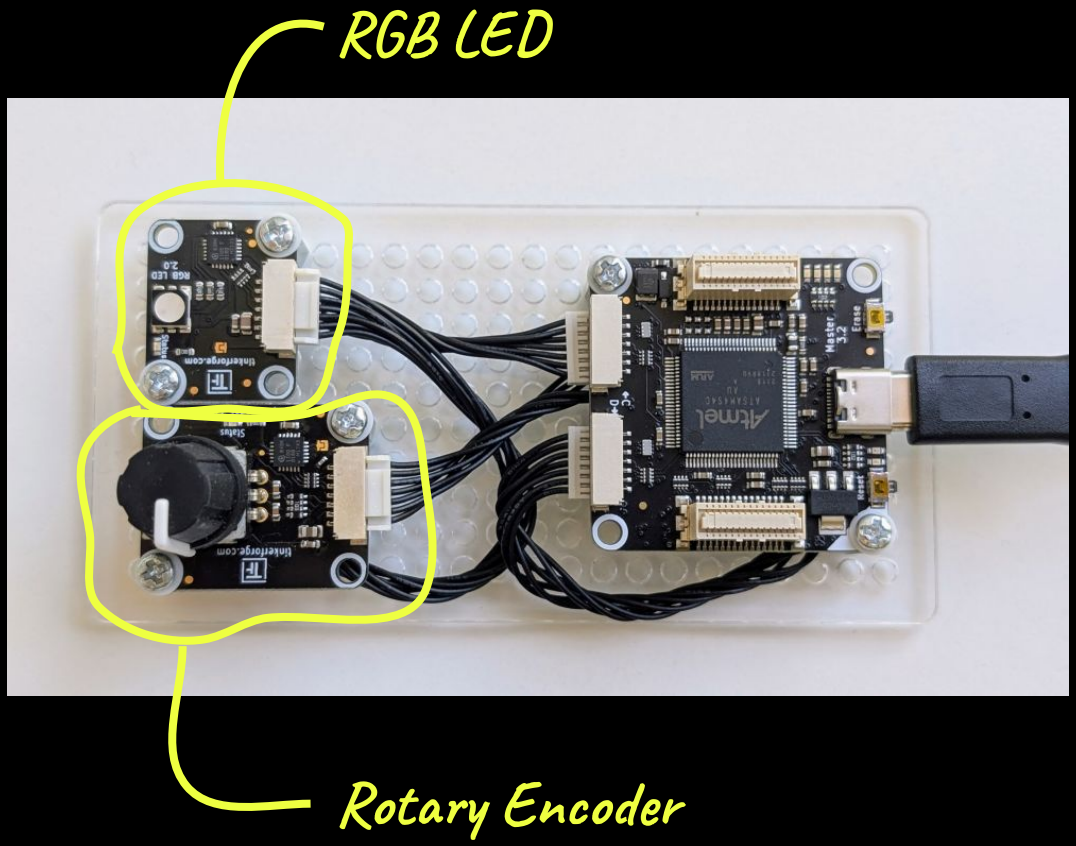
For any comments or feedback, please contact n.meseth@hs-osnabrueck.de.

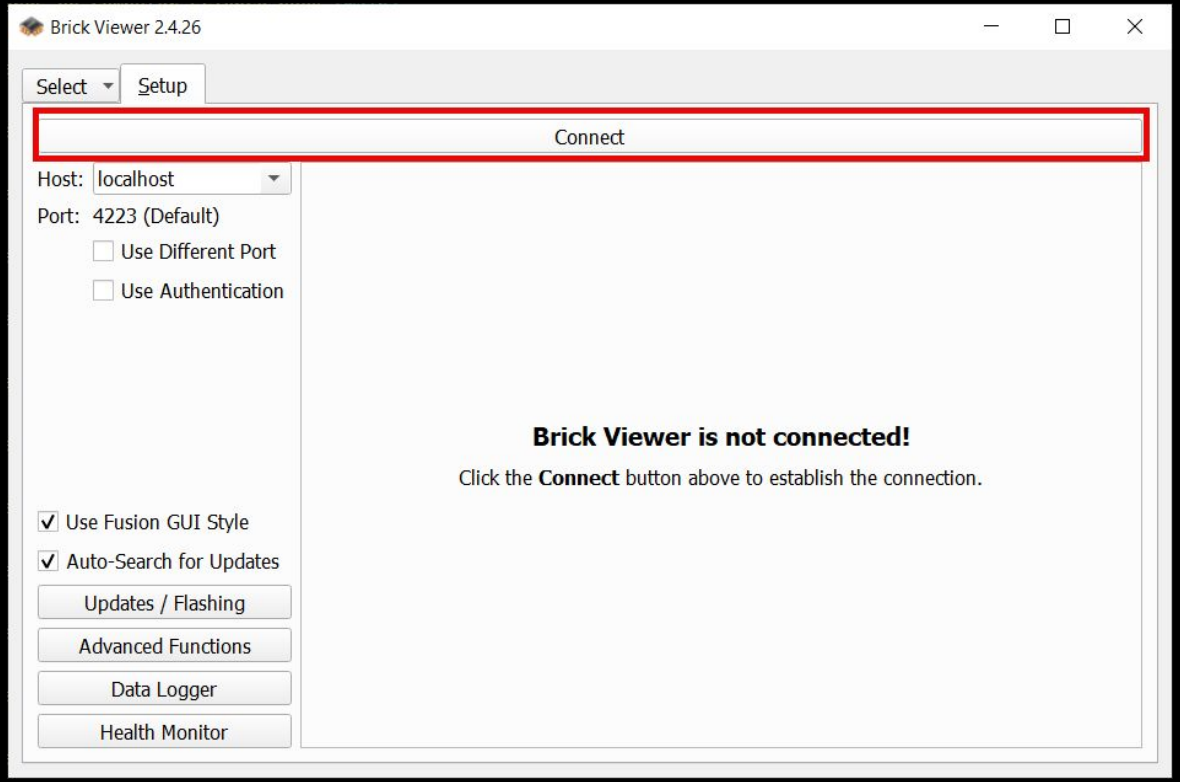
NUMBERS

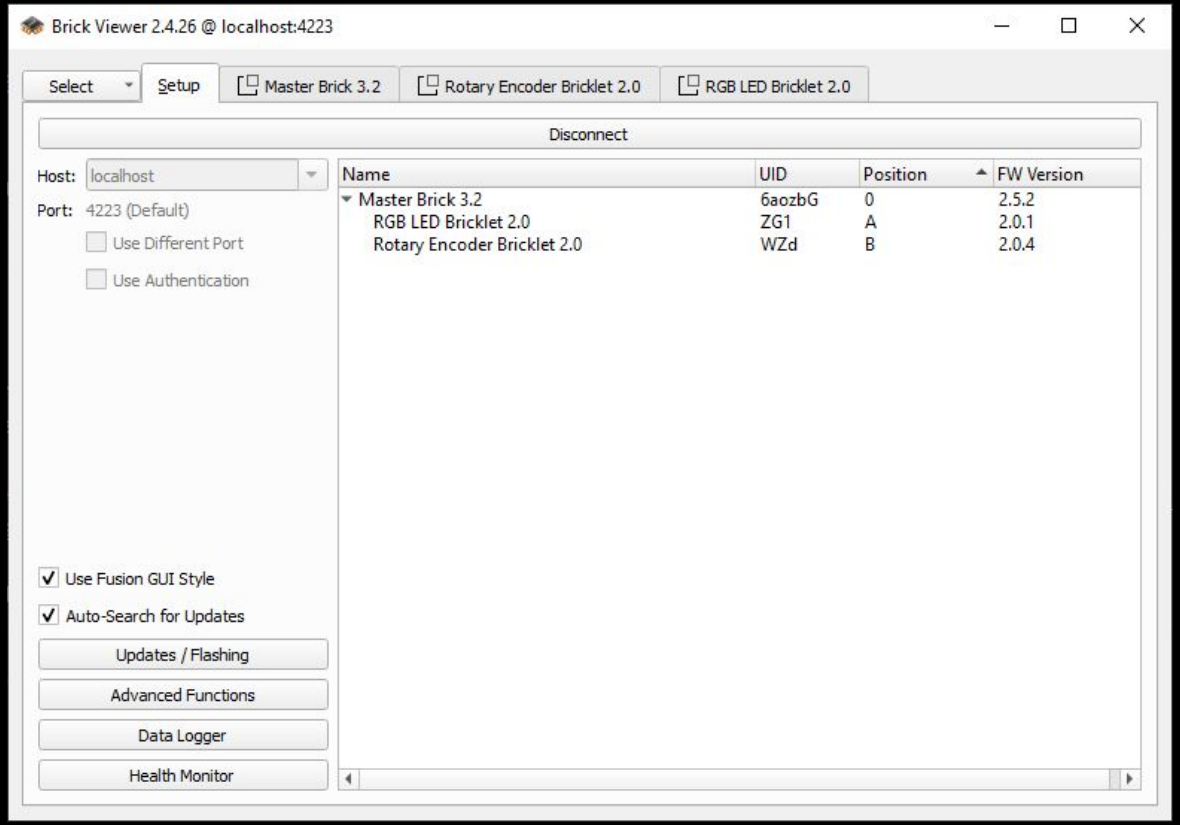
Supporting slides for chapter 2 of the book
Hands-On Computer Science

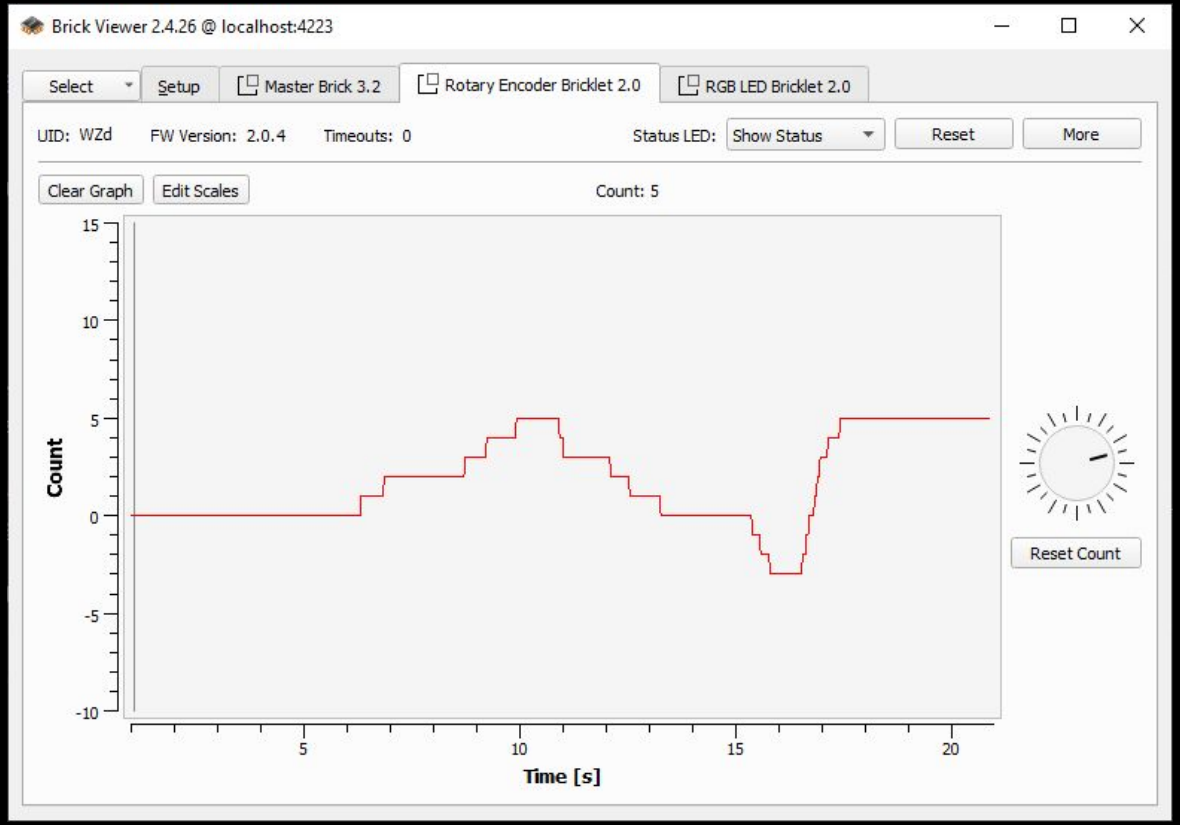

Rotary Encoder 2.0
Status
Master 3.2
Erase
Reset
atmel
ATSAMD21
ARM


RGB LED


RGB LED
Rotary Encoder


Brick Viewer 2.4.26
Select
Setup
Connect
Host: localhost
Port: 4223 (Default)
Use Different Port
Use Authentication
Brick Viewer is not connected!
Click the Connect button above to establish the connection.
Use Fusion GUI Style
Auto-Search for Updates
Updates / Flashing
Advanced Functions
Data Logger
Health Monitor


Brick Viewer 2.4.26 @ localhost:4223
Select
Setup
Master Brick 3.2
Rotary Encoder Bricklet 2.0
RGB LED Bricklet 2.0
Disconnect
Host:
localhost
Port:
4223 (Default)
Use Different Port
Use Authentication
Name
UID
Position
FW Version
Master Brick 3.2
6aozbG
0
2.5.2
RGB LED Bricklet 2.0
ZG1
A
2.0.1
Rotary Encoder Bricklet 2.0
WZd
B
2.0.4
Use Fusion GUI Style
Auto-Search for Updates
Updates / Flashing
Advanced Functions
Data Logger
Health Monitor


Brick Viewer 2.4.26 @ localhost:4223
Select
Setup
Master Brick 3.2
Rotary Encoder Bricklet 2.0
RGB LED Bricklet 2.0
UID: WZd
FW Version: 2.0.4
Timeouts: 0
Status LED:
Show Status
Reset
More
Clear Graph
Edit Scales
Count: 5
15
10
5
0
-5
-10
Count
5
10
15
20
Time [s]
Reset Count

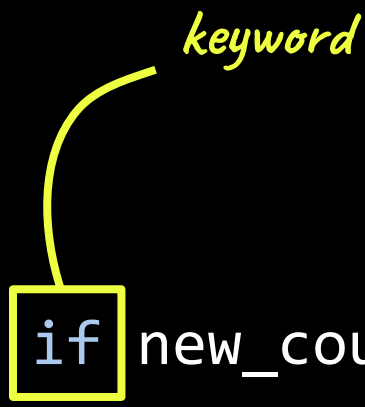
boilerplate code

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rotary_encoder_v2 import BrickletRotaryEncoderV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
knob = BrickletRotaryEncoderV2("WZd", ipcon)
```

reading the counter

```
from tinkerforge.ip_connection import IPConnection
from tinkerforge.bricklet_rotary_encoder_v2 import BrickletRotaryEncoderV2

ipcon = IPConnection()
ipcon.connect("localhost", 4223)
knob = BrickletRotaryEncoderV2("WZd", ipcon)

count = knob.get_count(reset=False)
```

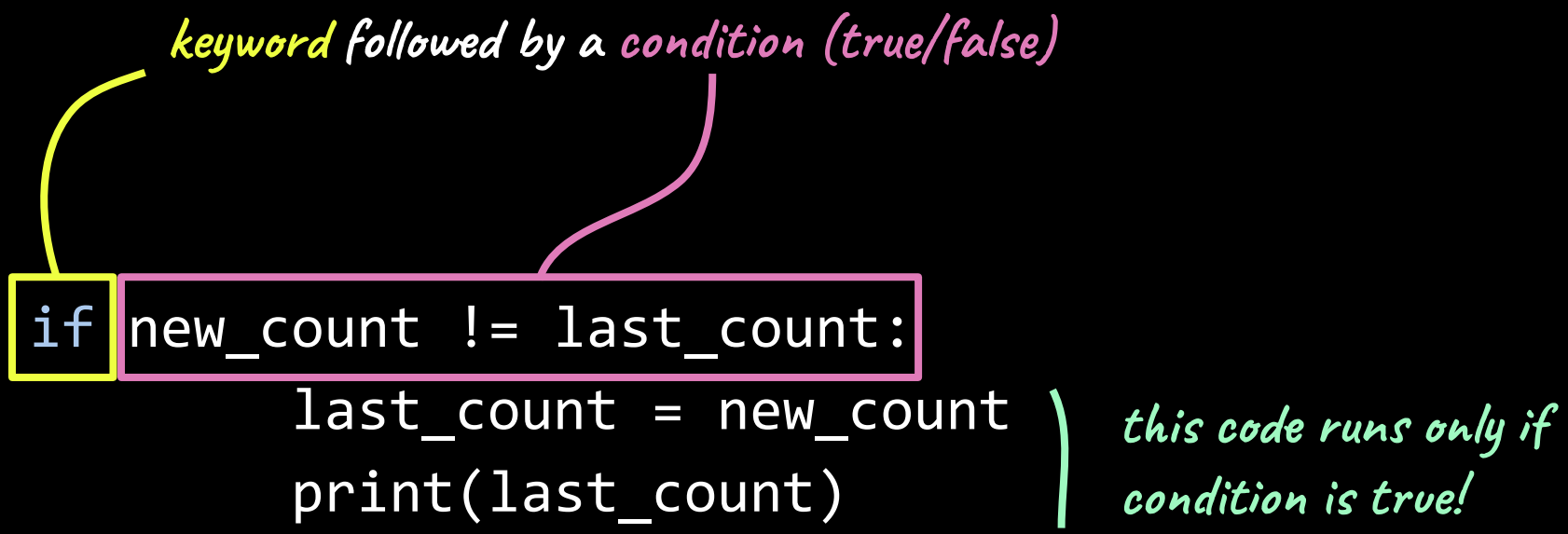
CONTROL STRUCTURES




```
if new_count != last_count:
        last_count = new_count
        print(last_count)
```

keyword followed by a condition (true/false)

```
if new_count != last_count:
        last_count = new_count
        print(last_count)
```


keyword followed by a condition (true/false)
if new_count != last_count:
last_count = new_count
print(last_count)
this code runs only if condition is true!

LED DIMMER V1

```
knob.reset()
last_count = 0

while True:
    new_count = knob.get_count(reset=False)

    if new_count != last_count:
        last_count = new_count
        led.set_rgb_value(last_count, last_count, last_count)
```

```
knob.reset()
last_count = 0

while True:
    new_count = knob.get_count(reset=False)

    if new_count != last_count:
        last_count = new_count
        led.set_rgb_value(last_count, last_count, last_count)
```

```
struct.error: ubyte format requires 0 <= number <= 255
```

NUMBER SYSTEMS

1 2 3

1	2	3
10^2	10^1	10^0

1	2	3
10^2	10^1	10^0
100	10	1

1	2	3
10^2	10^1	10^0

$$= 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

$$= 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$= 123$$

4	1	2	3
?	10^2	10^1	10^0

4	1	2	3
10^3	10^2	10^1	10^0

$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

4	1	2	3
10^3	10^2	10^1	10^0

$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

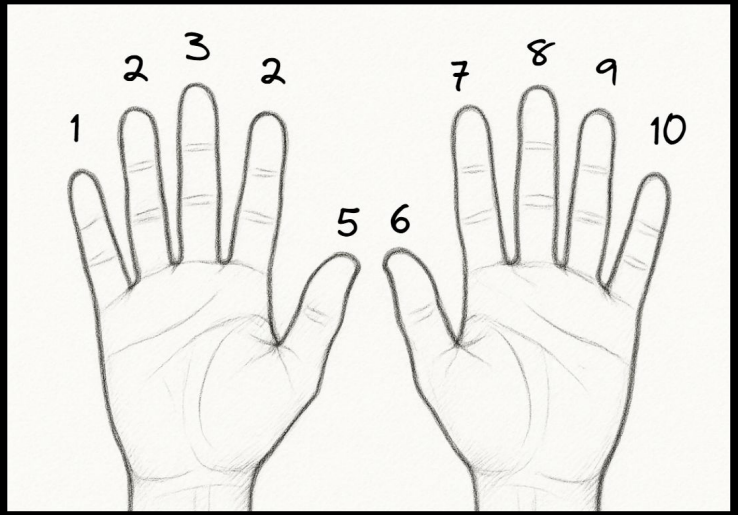
$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

4	1	2	3
10^3	10^2	10^1	10^0

$$= 4 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$$

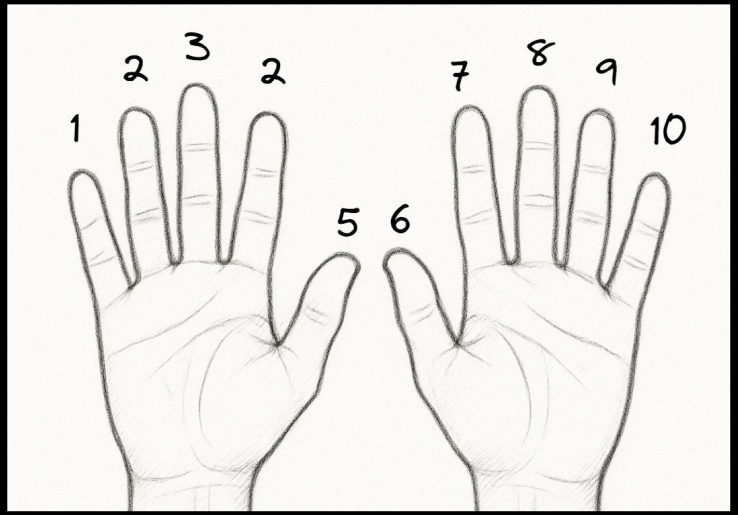
$$= 4 \times 1000 + 1 \times 100 + 2 \times 10 + 3 \times 1$$

$$= 4123$$

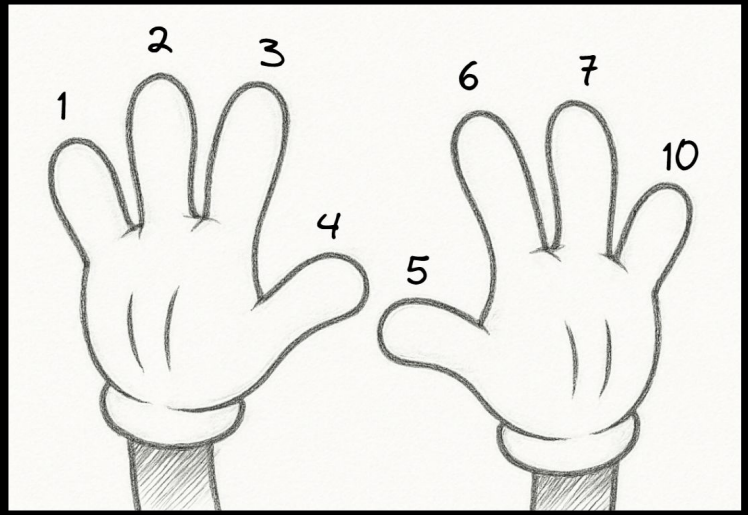


human hand

Images created with ChatGPT following: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.



human hand



cartoon character’s hand

Images created with ChatGPT following: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.

1 2 3 (octal)

1	2	3	(octal)
8^2	8^1	8^0	

1	2	3
8^2	8^1	8^0

(octal)

$$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$$

1	2	3
8^2	8^1	8^0

(octal)

$$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$$

$$= 1 \times 64 + 2 \times 8 + 3 \times 1$$

1	2	3
8^2	8^1	8^0

(octal)

$= 1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$

$= 1 \times 64 + 2 \times 8 + 3 \times 1$

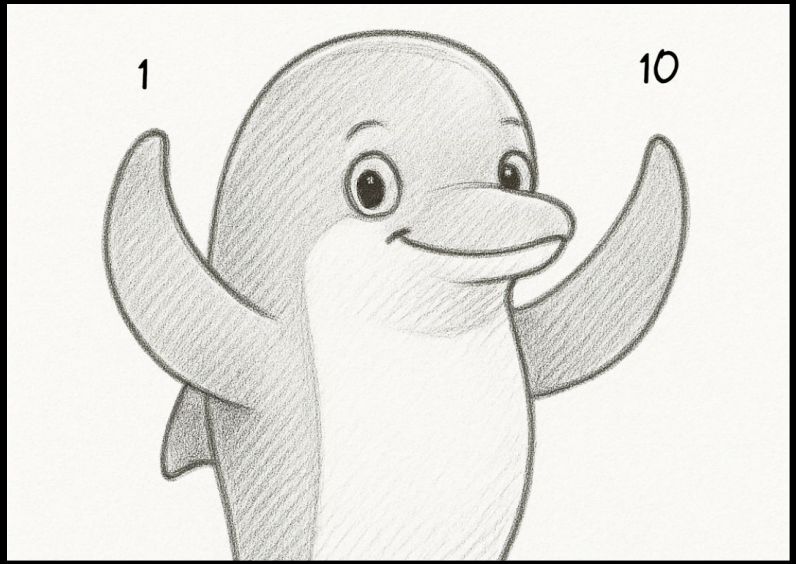
$= 83$ (decimal)


decimal
octal
8
?


decimal
octal
?
7


decimal
octal
16
?

decimal		octal
?		100



what now?

Image created with ChatGPT following: Petzold, Charles. Code: The Hidden Language of Computer Hardware and Software. 2nd ed., Microsoft Press, 2022.

0, 1, …

0, 1, 10, …

0, 1, 10, 11, …

0, 1, 10, 11, 100, …

0, 1, 10, 11, 100, 101, …

0, 1, 10, 11, 100, 101, 110

1 1 0 (binary)

1	1	0	(binary)
2^2	2^1	2^0	

1 1 0 (binary)

2^2 2^1 2^0

$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

1	1	0
2^2	2^1	2^0

(binary)

$$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 1 \times 4 + 1 \times 2 + 0 \times 1$$

1	1	0	(binary)
2^2	2^1	2^0	

$= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$= 1 \times 4 + 1 \times 2 + 0 \times 1$

$= 6$ (decimal)

2 3 4 5 6

0, 1, 10, 11, 100, 101, 110

place value systems

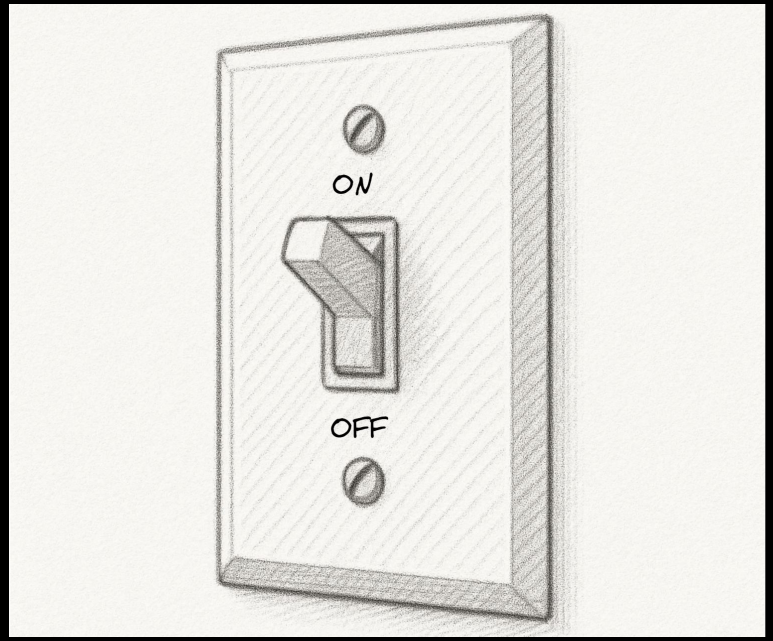
$$N = d_n * R^{n-1} + \ldots + d_2 * R^1 + d_1 * R^0$$

$d \in \{ 0, 1, \ldots R-1 \}$

n = number of digits
R = base

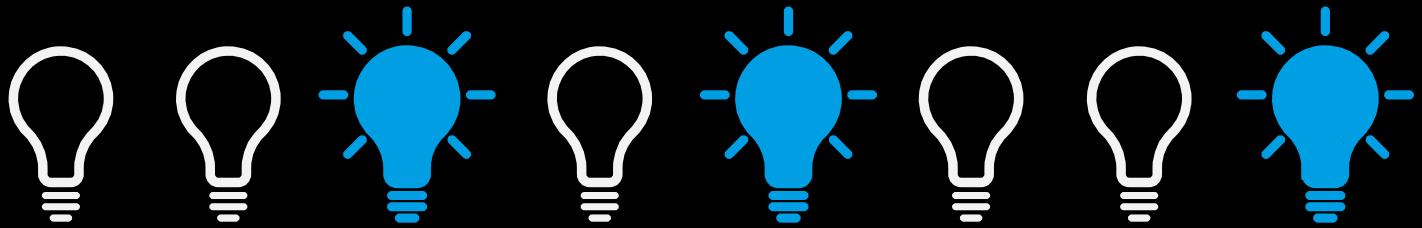
$$R \geq 2$$

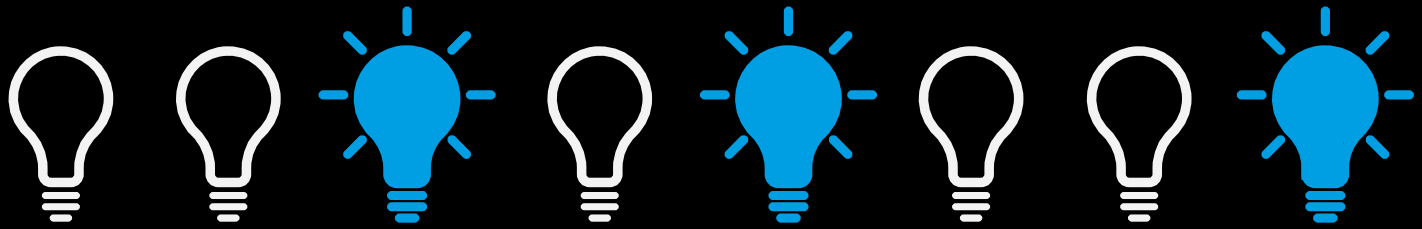
BITS & BYTES

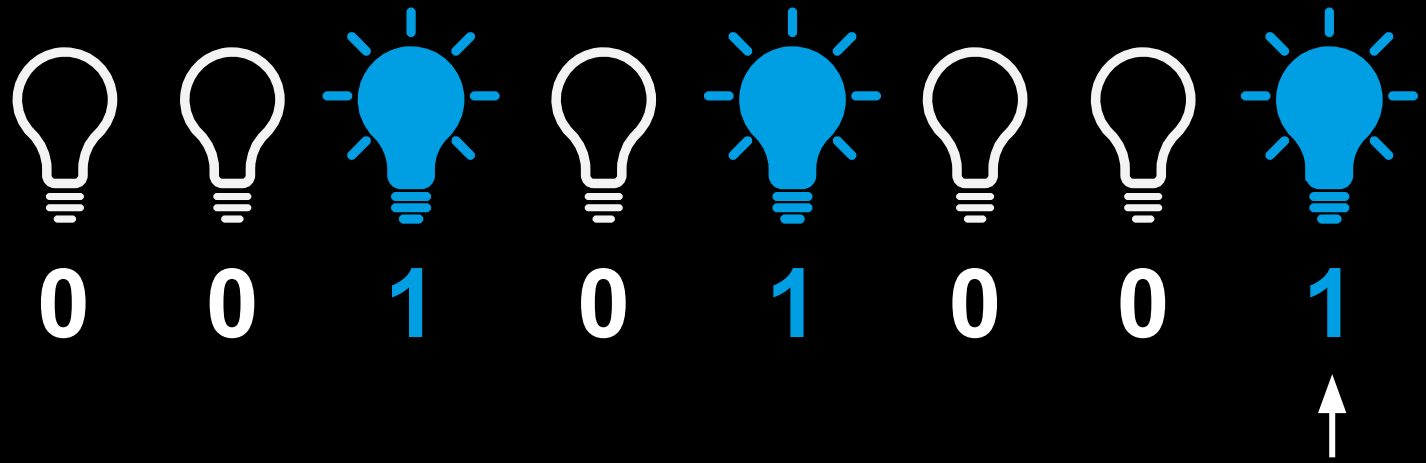


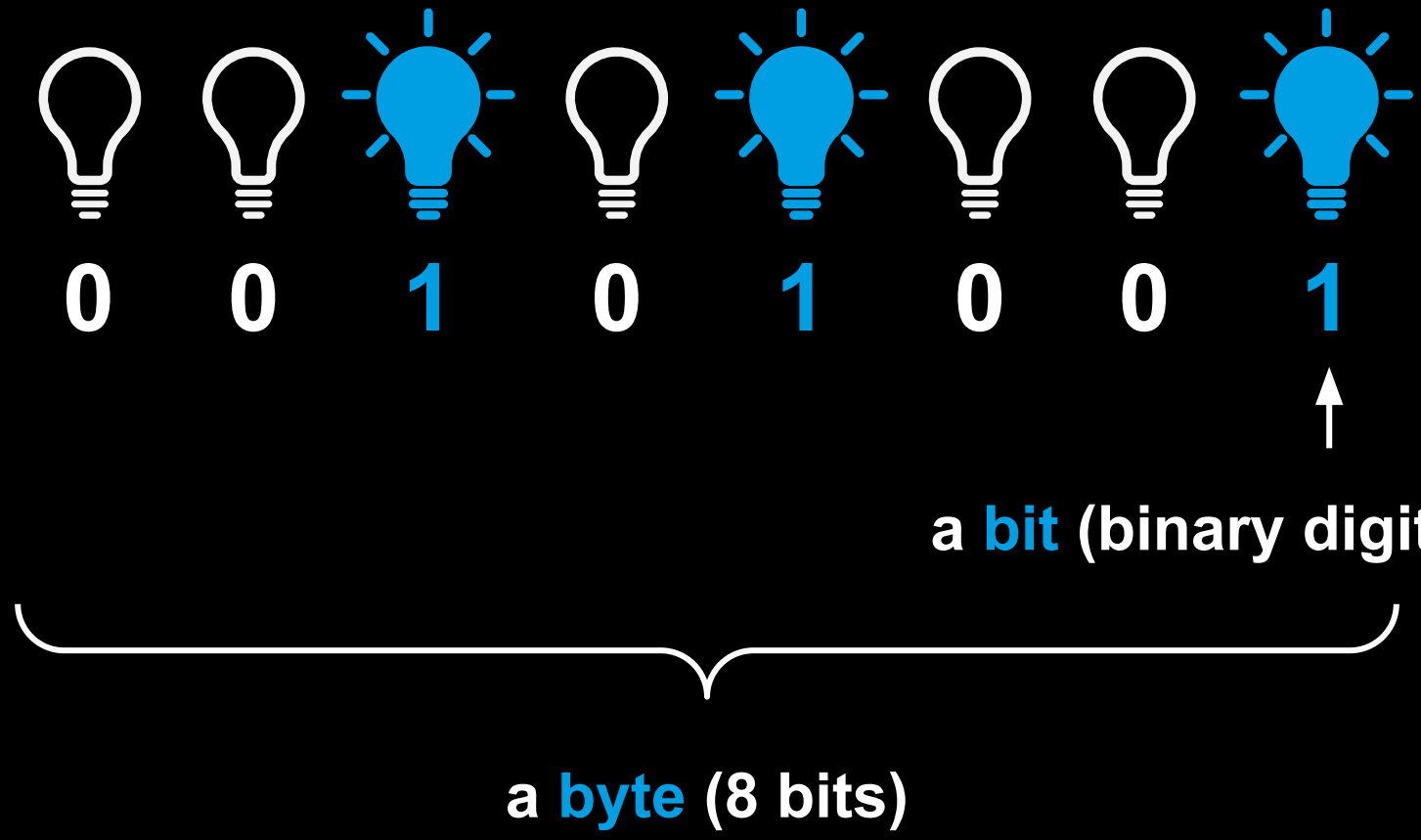
a binary number is like a switch

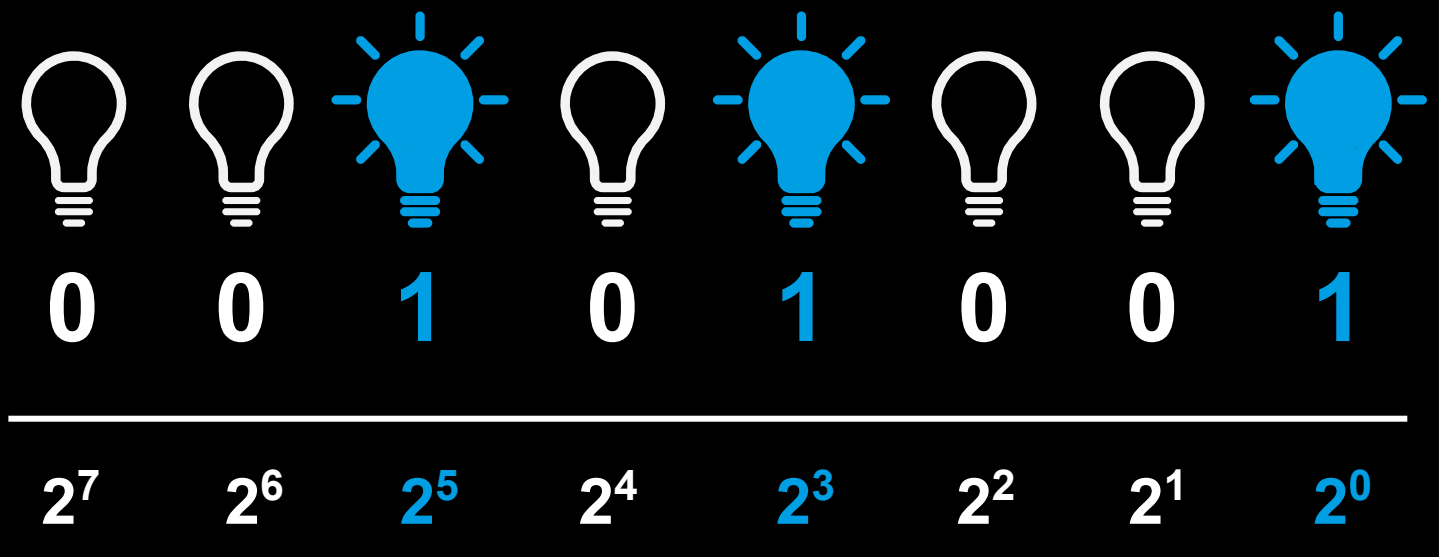
Image created with ChatGPT.

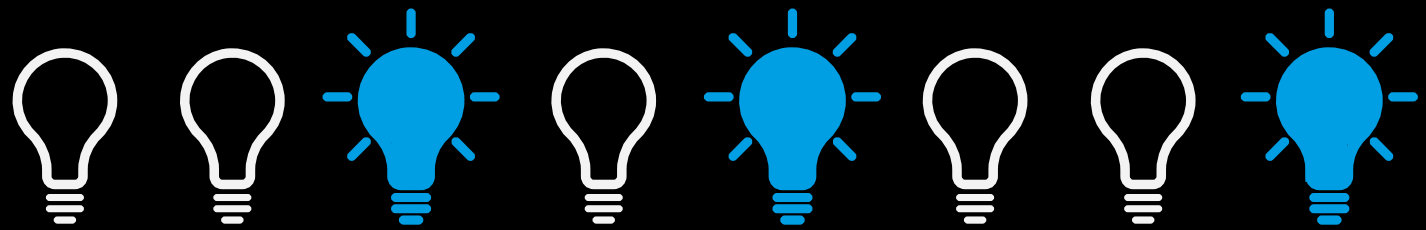



0 0 1 0 1 0 0 1


0 0 1 0 1 0 0 1
a bit (binary digit)

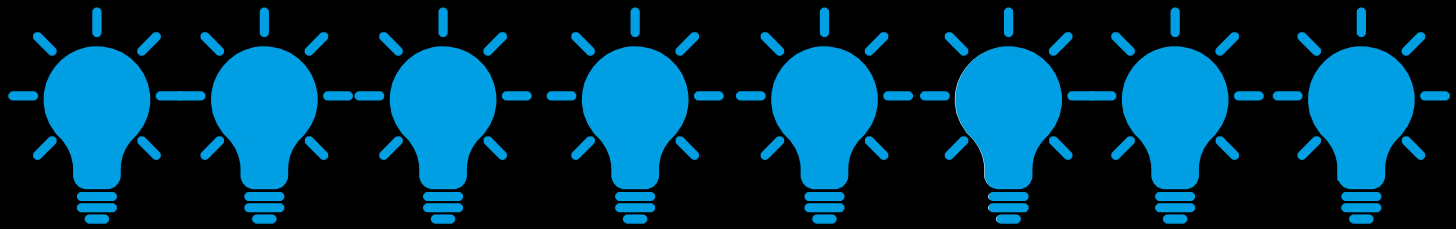

0 0 1 0 1 0 0 1
a bit (binary digit)
a byte (8 bits)


0 0 1 0 1 0 0 1
2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0



0	0	1	0	1	0	0	1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1



what can we store in one byte?

what comes after the byte?

10^3 bytes = 1.000 bytes = 1 Kilobyte (KB)

10^6 bytes = 1.000.000 bytes = 1 Megabyte (MB)

10^9 bytes = 1.000.000.000 bytes = 1 Gigabyte (GB)

10^{12} bytes = ?

https://www.matheretter.de/wiki/zehnerpotenzen-namen

2^{10} bytes = 1.024 bytes = 1 Kibibyte (KiB)

2^{20} bytes = 1.048.576 bytes = 1 Mebibyte (MiB)

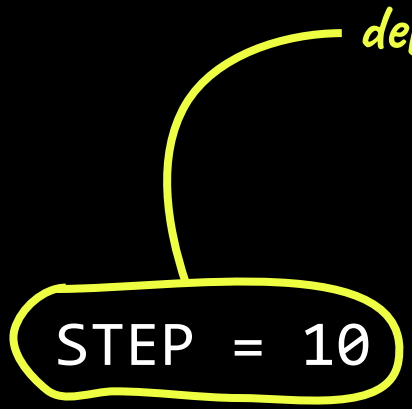
2^{30} bytes = 1.073.741.824 bytes = 1 Gibibyte (GiB)

how many bits are on a DVD with 4.7 GB capacity?

LED DIMMER V2

```
brightness = 0
...
diff = new_count - last_count
brightness += diff
```

```
brightness = max(0, min(255, brightness))
```




```
STEP = 10
...
brightness += diff * STEP
brightness = max(0, min(255, brightness))
```

READING THE BUTTON

```
while True:
    if knob.is_pressed():
        print("Button pressed")
    else:
        print("Button not pressed")
```

LED DIMMER V3

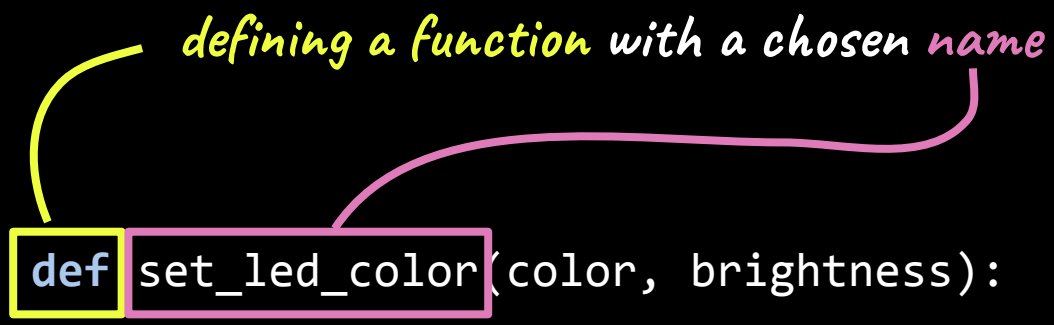
```
if color == "white":
    led.set_rgb_value(brightness, brightness, brightness)
if color == "yellow":
    led.set_rgb_value(brightness, brightness, 0)
if color == "green":
    led.set_rgb_value(0, brightness, 0)
```

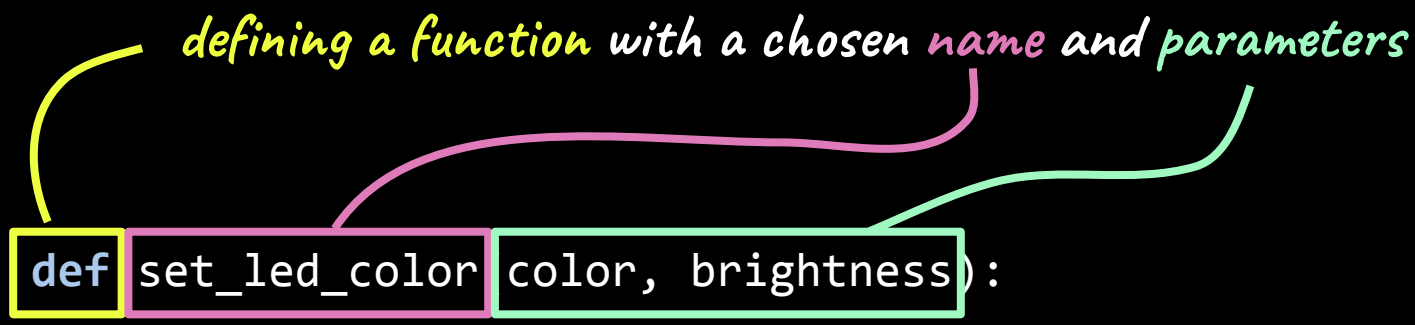
```
button_pressed_before = False
while True:
    button_pressed_after = knob.is_pressed()

    if button_pressed_before == True and button_pressed_after == False:
        if color == "white":
            color = "yellow"
        elif color == "yellow":
            color = "green"
        elif color == "green":
            color = "white"

    button_pressed_before = button_pressed_after
```

FUNCTIONS

defining a function

```
def set_led_color(color, brightness):
    if color == "white":
        led.set_rgb_value(brightness, brightness, brightness)
    if color == "yellow":
        led.set_rgb_value(brightness, brightness, 0)
    if color == "green":
        led.set_rgb_value(0, brightness, 0)
```


defining a function with a chosen name
def set_led_color(color, brightness):
if color == "white":
led.set_rgb_value(brightness, brightness, brightness)
if color == "yellow":
led.set_rgb_value(brightness, brightness, 0)
if color == "green":
led.set_rgb_value(0, brightness, 0)


defining a function with a chosen name and parameters
def set_led_color color, brightness):
if color == "white":
led.set_rgb_value(brightness, brightness, brightness)
if color == "yellow":
led.set_rgb_value(brightness, brightness, 0)
if color == "green":
led.set_rgb_value(0, brightness, 0)