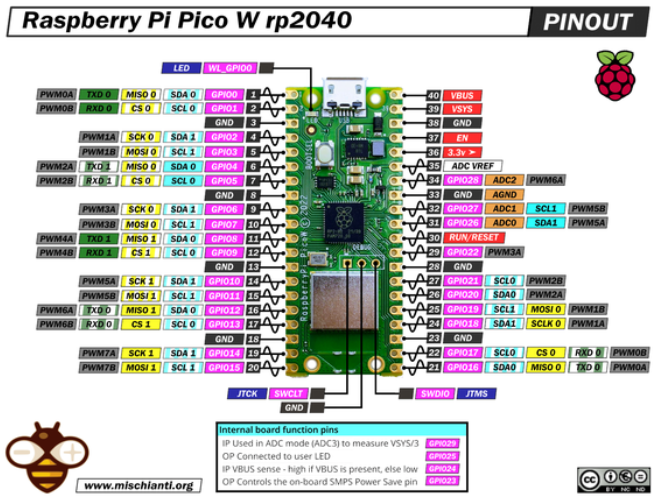
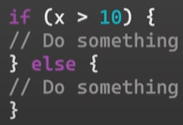
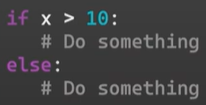
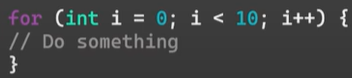
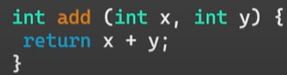
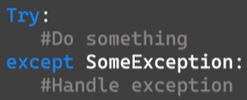
**Python Starter Kit**

**Day 0: Getting Started**

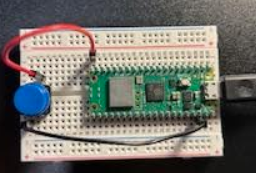
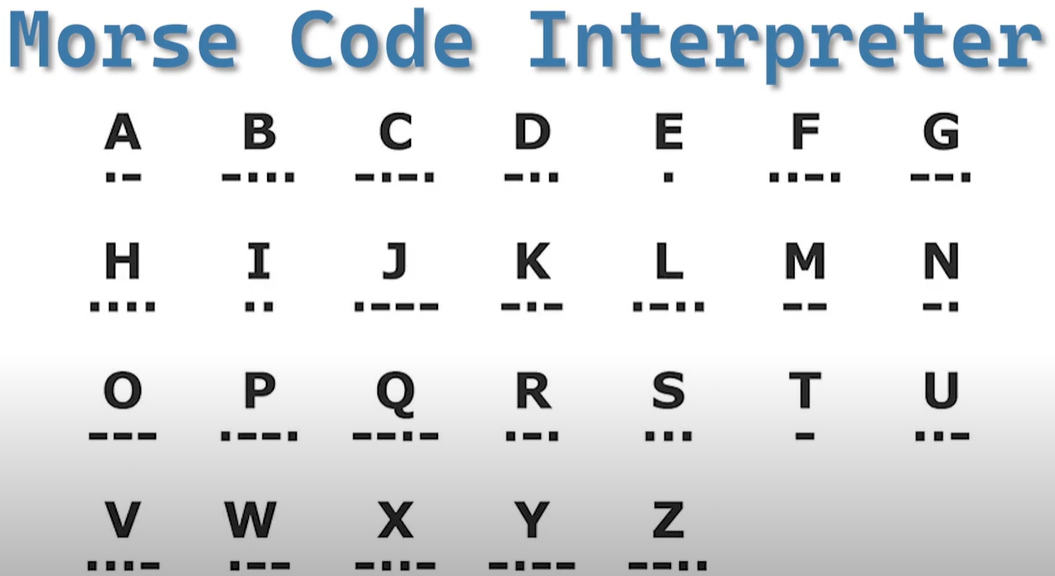
* This will be using Raspberry Pi Pico W micro controller
* ‘While True’ while loop is a loop to pretty much run indefinitely because True always equals True
* **IMPORTANT – PIN set in code is equal to the GP (pink) pin number, NOT the physical placement number**
  + **EX – GP15 (in pink in the diagram and pin 15 in code) is the same as physical pin 20 in the bottom, left corner**
  + **DO NOT USE PHYSICAL PIN NUMBER AS THE ASSIGNED PIN INPUT/OUTPUT**

****

**Day 1: Traffic Light Simulator**

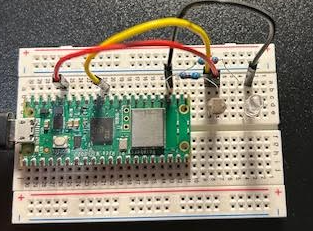
* **Differences between Python and C++ syntax**
  + Comments
    - C++ uses // for single line comment and /\* comment \*/ for multiline comments
    - Python uses # for a single line comment and ‘’’comment’’’ for multiline comments
  + Variable Declarations
    - C++ you must declare the data type before assigning a variable
      * Int x = 10 or float y = 20.5
    - Python no data type declaration is needed
      * X= 10 or y = 20.5
  + Printing
    - C++ uses Serial.prontln(“Hello\n”);
    - Python uses print(“Hello”)
  + Conditionals
    - C++
    - ****
    - Python
    - ****
  + Loops
    - C++
    - ****
    - Python
    - ****
  + Functions
    - C++
    - ****
    - Python
    - ****
  + Arrays and Lists
    - C++
    - ****
    - Python
    - ****
  + String Concatenation
    - C++ is more complex and usually requires a library ‘string.h’
    - Python is much simpler usually with no special library needed
  + Error Handling
    - C++ is done through return codes
    - Python uses Try-Except blocks
      * ****
  + Libraries
    - C++ uses #include <library.h>
    - Python uses import library
  + Misc
    - C++ uses {} to show the start and end of a code block
    - C++ uses ; at the end of lines
    - Python does not use {} or ;
    - Python does use indenting and spaces rather
    - C++ has setup and loop as a general setup
    - Python we just define out pins, create our function, call that function
* Import machine
  + Library we use to talk to our ‘machine’ in this case our Pico board
  + We use to establish output pins, read inputs
  + Used for a lot of commonly used machines
* Import time
  + Library to import timers, pauses, anything timing related
  + Similar to delay, Python uses sleep

**Day 2: Morse Code Machine**

* Morse code is a method used in telecom to encode text characters as a standardized sequence of two different signal durations called dots and dashes
* ****

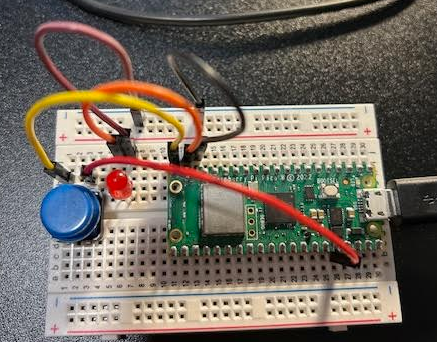
**Day 3: Night Light Simulator**

* The Photoresistor can only use a specific pin type – ADC
  + ADC = analog to digital converter
  + Allows us to read a range of values rather than just 1 and 0
* Adjusted night light threshold to under 63000 as light in room was 65000



**Day 4: Reaction Time**

* Same wiring as in the video, not the diagram with the code
* No additional concepts



**Day 5: Simon Says Game**

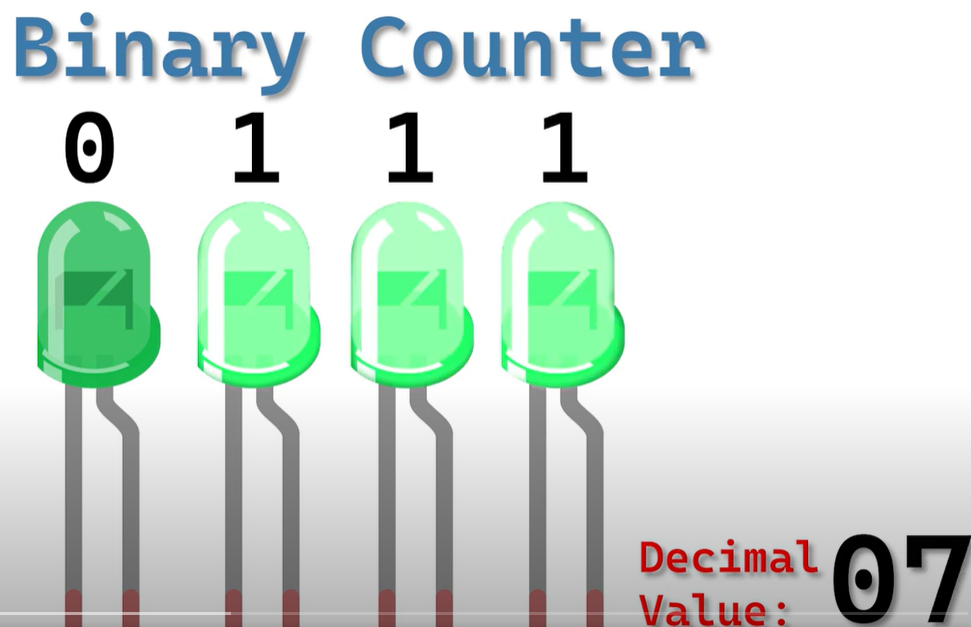
When adding a for loop in defining the OUT pin

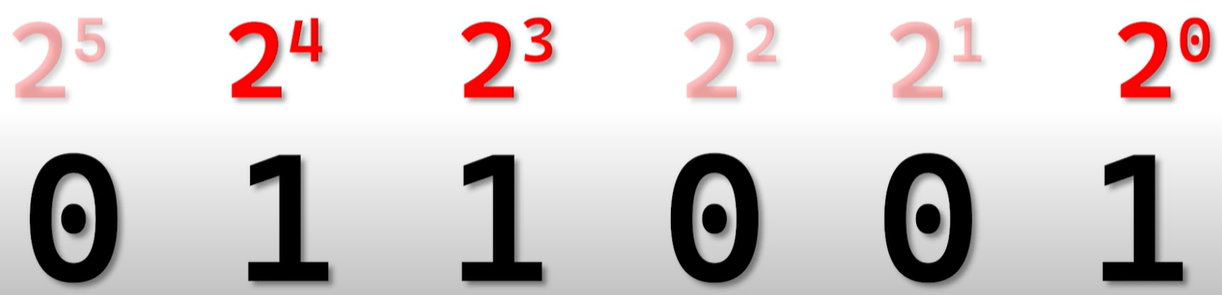
Led\_pins = [12,13]

leds = [machine.Pin(pin, machine.Pin.OUT) for pin in led\_pins]

* this for just sets everything in the led pins as an output

**Day 6: Binary LED Counter**

* Using binary code – we will represent a counter going from 1-15 using 4 LEDS
* Since binary only uses 1s and 0s, using 4 binary bits can only represent numbers 1-15
* EX
  + 0,0,0,1 = 1
  + 0,0,1,0 = 2
  + 0,1,1,1 = 7
* 
* How binary works every binary bit is to the power of 2
* EX



* And when each place is set to 1 or ON, take that number it represents and sum to get the number being representing
* In the example we have **(2^4) + (2^3) + (2^0)** = **16 + 8 + 1** = **25**
* **FUNCTION ENUMERATE**
  + Turns values of an array into an index to easily call later
  + EX led\_pins = [pin1, pin2, pin3, pin4]
  + Enumerate(led\_pins[2]) would be pin3

**Day 7: LED with Brightness Control**

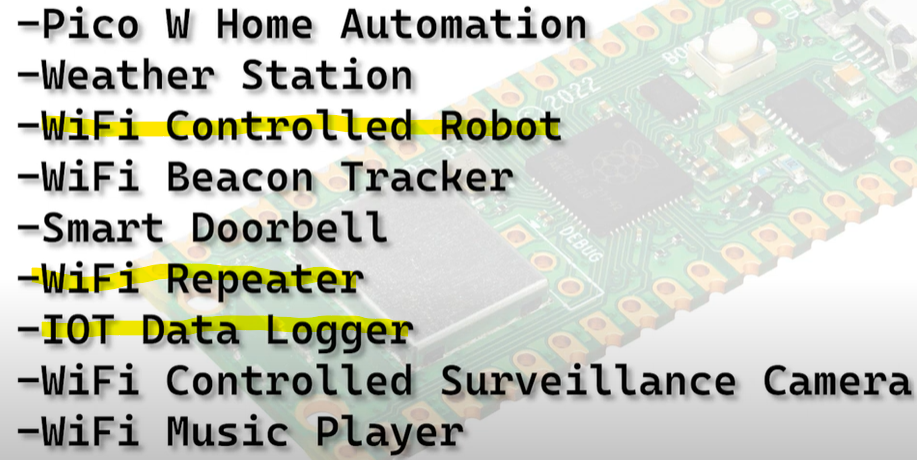
* Pulse with modulation – will allow us to change the brightness level of our LED

**Day 8: Built in Temp Reader**

* The board has a built-in temp reader in pin 4!

**Day 9: Wifi Temp Reader**

* W in Pico stands for Wifi
* Other Projects:

****