



PYTHON BASICS & ROBOT ASSEMBLY

Day 1: Code & Build

Tuesday, February 3, 2026



Today's Schedule

SESSION 1: 9:00 - 10:50 AM



9:00 - 9:20 • Setup & Hello World

Install Thonny, first program



9:20 - 10:00 • Python Fundamentals

Variables, data types, operations



10:00 - 10:50 • Control Flow

If statements, loops, functions + practice



BREAK: 10:50 - 11:10 AM

SESSION 2: 11:10 AM - 1:00 PM

What is Python?

Python is a programming language – a way to give instructions to computers.



Easy to Learn

Reads like English!



Powerful

Used by Netflix, Google,
NASA



Versatile

Games, websites,
robots, AI



In-Demand

Top programming
language



Access Course Materials on GitHub

All code files and presentations are available online:

1

Open Your Browser

Chrome, Safari, Firefox, Edge - any browser works

2

Go to GitHub Repository

Type this URL (case-sensitive!):

`github.com/Wisdomerh/IBM-REDHAT-TY-2026`

3

Bookmark This Page!

You'll use it throughout the week



Ask your instructor if you can't find the link!



What's in the Repository?

Everything you need for this week:

Debug_Challenges

3 broken code files + solutions

challenge1_BROKEN.py, challenge1_SOLUTION.py, etc.

Behavior_Challenges

5 robot behavior programs

square_pattern.py, figure8.py, spin.py, etc.

Main_Code

Full robot control program

main_cleaned.py (obstacle avoidance + WiFi)

Presentations

All PowerPoint slides

Tuesday.pptx, Thursday.pptx, etc.

To Download: Click green **Code** button → Download ZIP → Extract files



Setting Up Thonny

If you haven't installed Thonny yet:

1. Open your web browser
2. Go to thonny.org
3. Download for your OS (Windows/Mac/Linux)
4. Install and open Thonny



What is Thonny?

An IDE (Integrated Development Environment) – like Microsoft Word, but for writing code!



Need Help? Raise your hand – we'll come to you!



Your First Program: Hello World!

Let's write our first line of Python code!

```
print("Hello, World!")
```

Follow Along:

1. Type the code above into Thonny
2. Press the green ▶ Play button (or F5)
3. Watch your computer say hello! 🗣️



Congratulations! You're officially a programmer!





Variables: Storing Information

Think of a variable as a labeled box where you store information:

```
# Storing text (strings)
name = "Alice"
school = "Your School Name"

# Storing numbers
age = 15
height = 172.5

# Using variables
print("My name is " + name)
```




Data Types

String (str)

Text

Examples: "Hello", "Python"

Integer (int)

Whole numbers

Examples: 42, 2026

Float

Decimals

Examples: 3.14, 98.6

Boolean

True/False

Examples: True, False



Python as a Calculator

Try these operations:

+ Addition `10 + 5 → 15`

- Subtraction `10 - 5 → 5`

***** Multiplication `10 * 5 → 50`

/ Division `10 / 5 → 2.0`

****** Power `10 ** 2 → 100`

% Remainder `10 % 3 → 1`

 Try It: `age * 365` (your age in days!)



Making Decisions: If Statements

Programs make decisions based on conditions:

```
age = 16

if age >= 17:
    print("You can drive!")
elif age >= 16:
    print("You can get a permit!")
else:
    print("Soon!")
```

⚠ Indentation (spaces) matters in Python!

</> Loops: Repeating Actions

For Loop (count specific times):

```
for i in range(5):  
    print("Hello", i)
```

While Loop (repeat while condition is true):

```
count = 3  
while count > 0:  
    print(count)  
    count = count - 1
```



Functions: Reusable Code

Write once, use many times!

```
def greet(name):  
    print("Hello, " + name)  
  
# Use the function  
greet("Alice")  
greet("Bob")
```

- Write once, use many times
- Keep code organized
- Easy to debug

Practice Time!

Until 10:50 - try these:

1. Make a countdown from 10 to 1
2. Check if a number is even or odd (use %)
3. Create a function that converts Celsius to Fahrenheit

 **Break Time!** 

20 minutes - back at 11:10!

Get ready for ROBOT ASSEMBLY! 



ROBOT ASSEMBLY TIME!

Let's connect the wheels and sensor!



Your Robot Parts

Simple setup - just a few parts to connect:

- ✓ Raspberry Pi Pico (the brain)
- ✓ 4 wheels (2 front, 2 rear)
- ✓ 2 motor drivers (front & rear)
- ✓ 1 ultrasonic sensor
- ✓ Wires to connect everything



Keep all parts organized!



Connecting Your Robot

This is what we'll be doing:

- 1** Connect FRONT DRIVER to Pico (4 wires)
- 2** Connect REAR DRIVER to Pico (4 wires)
- 3** Connect ULTRASONIC SENSOR to Pico (3 Wires)
- 4** Connect BATTERY PACK to power
- 5** Test the connections

We'll do each step together!



Safety First!

Important Rules:

- ⚡ Turn OFF power before connecting wires
- 📌 Each wire goes to a specific pin on the Pico
- 📌 We'll show you exactly where each wire goes
- 📌 Double-check before turning power on
- 📌 Make sure not to pull on the wires too hard!!!!

🙋 **ALWAYS ask if you're unsure!**



Raspberry Pi Pico Wiring Diagram

⚠ **IMPORTANT:** Power (5V) and Ground already connected to Pico

REAR DRIVER

● Yellow wire → GP 2

● Blue wire → GP 3

● Black wire → GP 5

● Red wire → GP 6

FRONT DRIVER

● Yellow wire → GP 8 (*next to GP 9*)

● Blue wire → GP 9

● Black wire → GP 11

● Red wire → GP 12

ULTRASONIC SENSOR

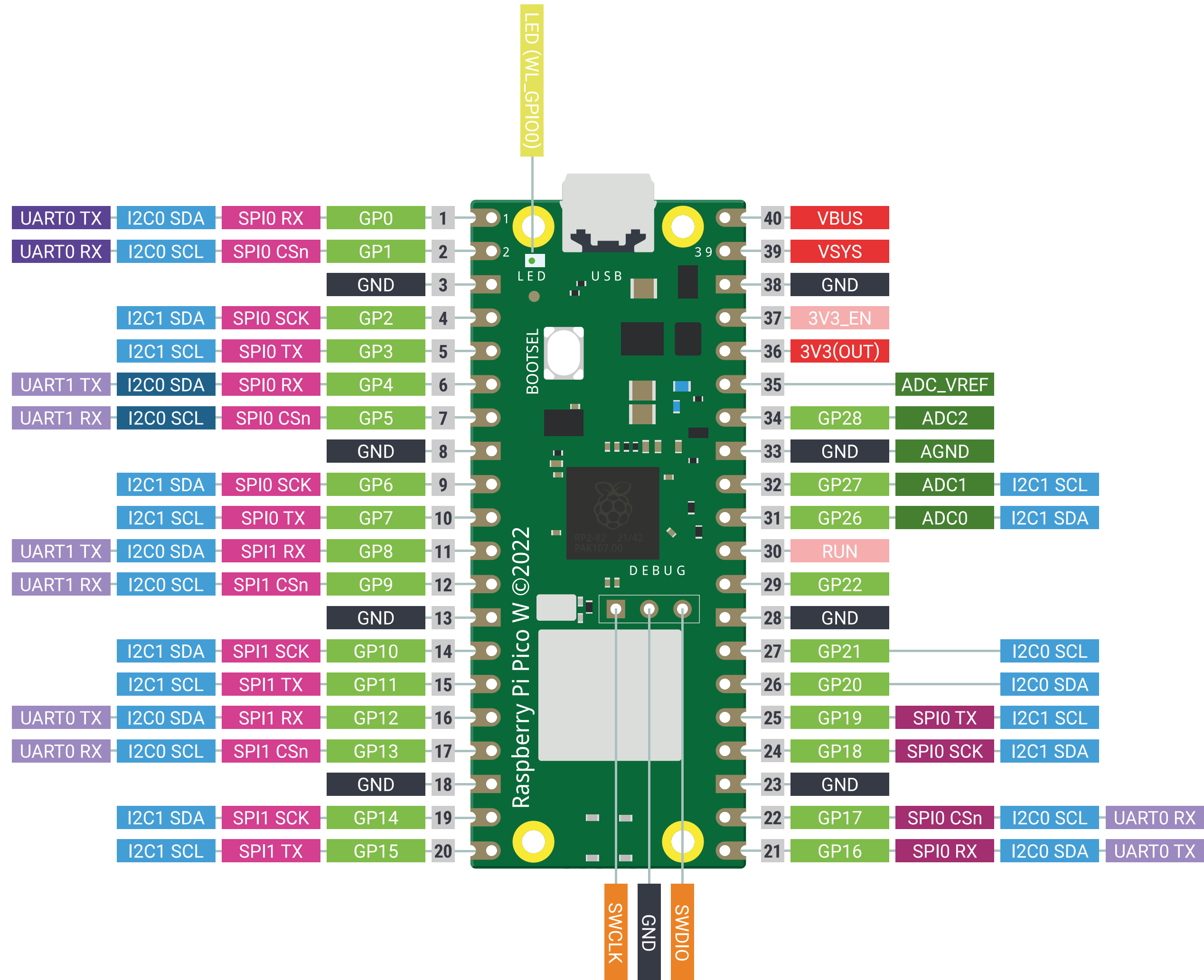
● Black wire (Trig) → GP 14

● Blue wire (Echo) → GP 15

● Yellow wire → Any **GROUND** pin (GND)

Key Reminders:

- GP 8 label may be hidden under black frame - it's **next to GP 9**
- GP 15 is **on the mounting hole**
- Ask for help before connecting if unsure! 🙋



RP2040

	Power
	Ground
	UART / UART (default)
	GPIO, PIO, and PWM
	ADC
	SPI / SPI (default)
	I2C / I2C (default)
	System Control
	Debugging

Infineon 43439

	GPIO
--	------

</> Testing Your Robot - First Wheel

Here's code to test ONE wheel. Use what you learned to test the others!

```
from machine import Pin
import time

# Rear right wheel (GP2 and GP3)
rear_right_forward = Pin(2, Pin.OUT)
rear_right_backward = Pin(3, Pin.OUT)

# Test: Spin wheel forward for 2 seconds
rear_right_forward.value(1)
rear_right_backward.value(0)
time.sleep(2)

# Stop the wheel
rear_right_forward.value(0)
rear_right_backward.value(0)
```



YOUR CHALLENGE: Test the other 3 wheels!

Hints:

- Rear Left: GP 5 & GP 6
- Front left: GP 8 & GP 9
- Front right: GP 11 & GP 12

Testing the Ultrasonic Sensor

This sensor measures distance - let's test it with your hand!

```
from machine import Pin, time_pulse_us
import time

# Ultrasonic sensor pins
trig = Pin(14, Pin.OUT)
echo = Pin(15, Pin.IN)

def get_distance():
    trig.value(0)
    time.sleep_us(2)
    trig.value(1)
    time.sleep_us(10)
    trig.value(0)

    pulse_time = time_pulse_us(echo, 1, 30000)
    distance = (pulse_time / 2) / 29.1 # in cm
    return distance

# Continuous testing loop
while True:
    dist = get_distance()
    print("Distance:", dist, "cm")
    time.sleep(0.5)
```

 **TRY IT: Move your hand closer and further from the sensor!**

What You'll See:

Numbers updating in real-time showing distance in centimeters

- Close hand: ~5-10 cm
- Arm's length: ~50-70 cm
- Far away: 100+ cm

Python Controls Your Robot

On Thursday, you'll see code like this:

```
# Set motor speed
speed = 50

# Check for obstacles
distance = sensor.read()
if distance < 20:
    turn_right()
else:
    move_forward()

# Make it drive in a square
for i in range(4):
    move_forward()
    turn_left()
```




What's Tomorrow?

Wednesday: SETU Visit

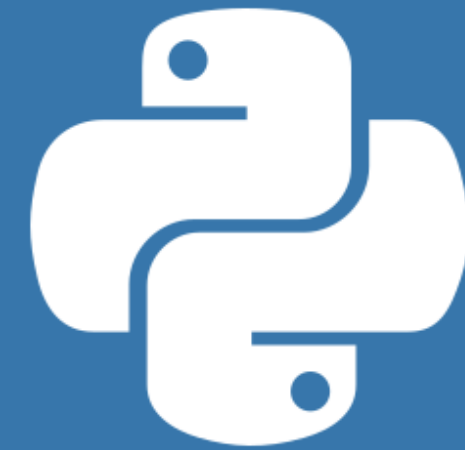
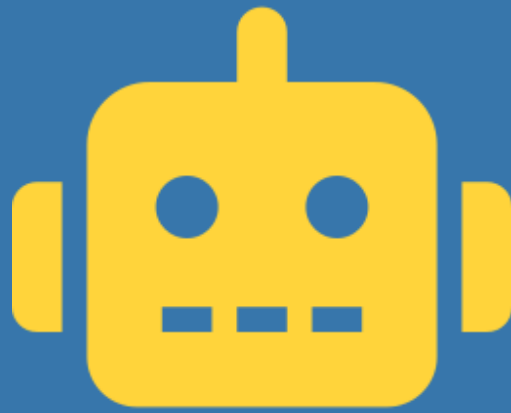
- A Visit from an SETU student/lecturer
- Learn about computer science courses
- Ask questions about college life

Thursday: Advanced Python + Robot Programming

- Write Python to control your robot
- Fix Bugs
- Program obstacle avoidance
- Make it drive autonomously



Review Python basics tonight!



Amazing Work Today!

You're a Python programmer
AND a robot builder!

See you tomorrow! 🙌