



LE/ESSE 2220 Algorithmic and Computational Methods

Lab 1: Raspberry Pi + Python environment

(Fall 2025-2026)

Z. ARJMANDI (ZARARJ@YORKU.CA)

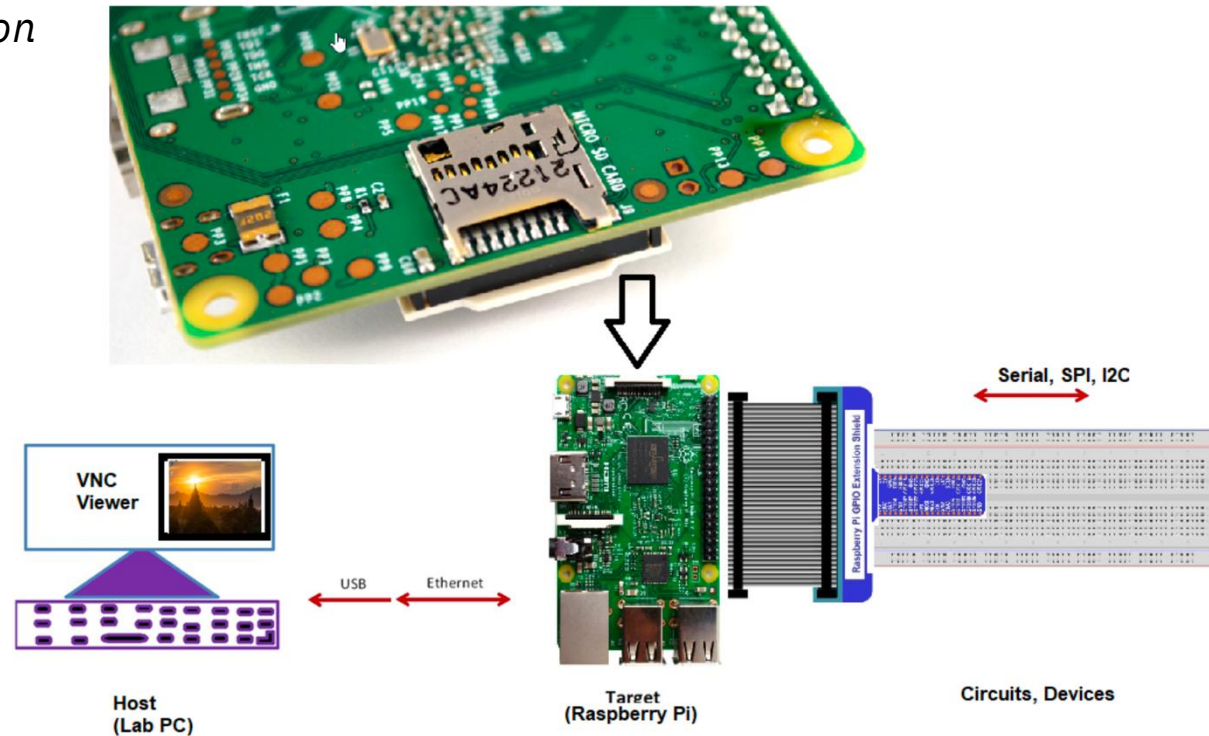
YORK 

Raspberry Pi

➤ Required components to connect a Raspberry Pi:

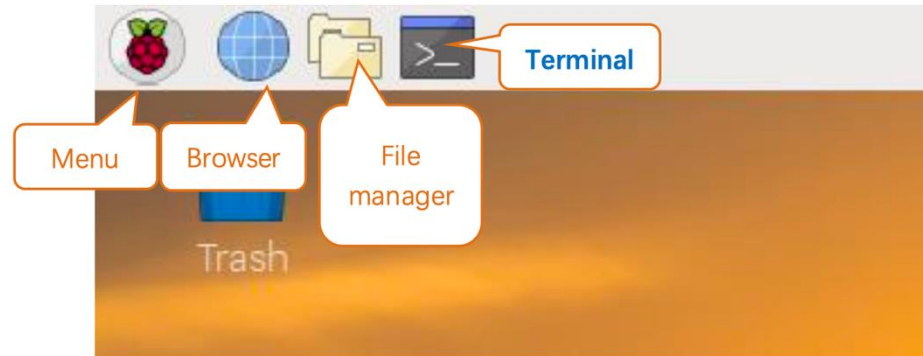
- Power cable to supply electricity and turn it on
- Ethernet cable to connect it to your PC (using VNC Viewer as the monitor*)
- 5V power supply for the cooling fan

* *Follow the tutorial to set up the software connection*

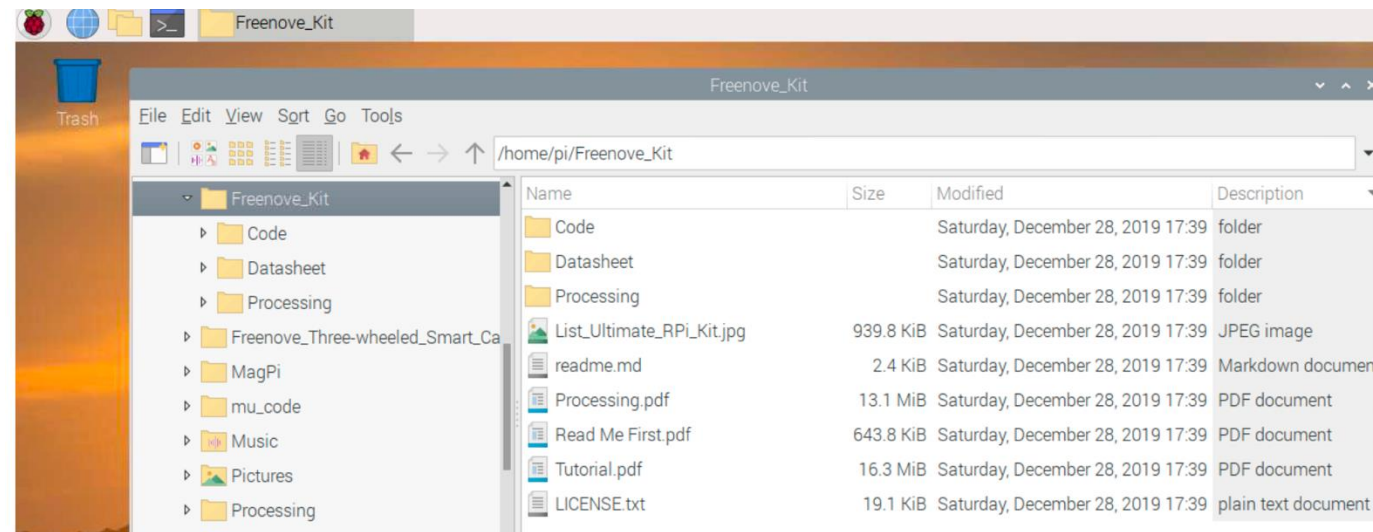


Raspberry Pi OS

- Raspberry Pi OS is built on the Linux operating system, which means you can use standard Linux commands and follow its rules.



- Freenove prewritten codes



Structure of a Python Program

➤ A Python program is made of **blocks of code** that work together:

- **Imports (libraries)** extra tools you can use
- **Functions** reusable instructions
 - **functions must be defined before they're called** in the main code.
- **Main Code** where the program starts running

```
# 1. Import libraries
import random
```

```
# 2. Define functions
def greet(planet):
    print("Hello,", planet, "!")

def countdown(n):
    for i in range(n, 0, -1):
        print(i)
    print("Liftoff!")

def choose_planet():
    planets = ["Mars", "Venus", "Jupiter"]
    return random.choice(planets)
```

```
# 3. Main code
countdown(5)
planet = choose_planet()
greet(planet)
```

if, elif, and else

- if → first condition
- elif → more conditions (“else if”)
- else → runs if none of the above are true

```
planet = "Mars"

if planet == "Earth":
    print("Hello, Earth!")
elif planet == "Mars":
    print("Welcome to Mars!")
else:
    print("Unknown world")
```

For Loops

› What is a for loop?

- Repeats a block of code a set number of times
- Often used to go through numbers, lists, or text

```
for variable in sequence:  
    # code to repeat
```

1. Counting with range()

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

2. Countdown with range(start, stop, step)

```
for i in range(5, 0, -1):  
    print(i)  
    print("Liftoff!")
```

3. Looping through a List

```
planets = ["Mercury", "Venus", "Earth", "Mars"]  
for planet in planets:  
    print("Hello,", planet)
```

4. Looping through Characters in a String

```
for letter in "NASA":  
    print(letter)
```


While Loops

- Repeats a block of code **as long as a condition is true**

1. Basic Countdown

```
count = 5
while count > 0:
    print(count)
    count -= 1
print("Liftoff!")
```

2. Waiting for a Condition

```
planet = ""
while planet != "Mars":
    planet = input("Enter a planet: ")
print("Welcome to Mars!")
```

3. Infinite Loop (be careful!)

```
while True:
    print("Orbiting Earth...")
```

break and continue in Loops

➤ **break** → Stop the loop early

```
for planet in ["Mercury", "Venus", "Earth", "Mars"]:  
    if planet == "Earth":  
        break  
    print("Exploring", planet)
```

➤ **continue** → Skip one step and go on

```
for planet in ["Mercury", "Venus", "Earth", "Mars"]:  
    if planet == "Earth":  
        continue  
    print("Exploring", planet)
```


Data Types in Python

› Common Data Types:

- **String (text):** "Mars"
- **Integer (whole number):** 225
- **Float (decimal number):** 3.14
- **Boolean (True/False):** True

› Python automatically figures out the data type when you assign a value

```
x = "Mars"      # String
y = 225         # Integer
z = 3.14        # Float
alive = True    # Boolean
```

› Defining Explicit Data Types:

```
x = str(123)    # "123"
y = int("456")  # 456
z = float("3.14") # 3.14
```

› Most Common Data Types

- **str** → String (text) → "Hello"
- **int** → Integer (whole numbers) → 42
- **float** → Decimal numbers → 3.14
- **bool** → Boolean (True/False) → True
- **list** → Ordered collection → [1, 2, 3]
- **dict** → Key-value pairs → {"planet": "Mars", "moons": 2}
- **NoneType** → Represents nothing → None

Lab 0 Assignment: “Where is the ISS right now?”

➤ Step 1 – Visit the Website

- Where is the ISS right now? <https://www.n2yo.com/?s=25544>
- Look at the live values:
 - Latitude
 - Longitude
 - Altitude
 - Speed
 - Visibility (Daylight / Shadow)

➤ Step 2 – Copy Values into Code

- Everyone **must copy the latitude**.
- Then choose **at least one more** value (your group decides).

Lab 0 Assignment: “Where is the ISS right now?”

➤ Step 3 – Write Your Program

➤ Your program must include:

- **Variables**

- Copy the **latitude** from the ISS website.
- Copy **one more value of your choice** (longitude, speed, altitude, or visibility).

- **A Condition**

- Use latitude to print “**Hello from the Northern Hemisphere!**” if latitude > 0,
- otherwise print “**Hello from the Southern Hemisphere!**”.

- **At Least One Function**

- Define a function of your choice that does something useful with your data.
- Examples:
 - `def check_speed(speed):` → prints “Super fast!” if speed > 27000
 - `def show_location(lat, lon):` → prints both coordinates

- **A Loop**

- Use a for or while loop to repeat your function call multiple times (e.g., 3 times or countdown style).

Report Format (short, personal, verifiable)

› 1. Names & Group Info

- Group number, member names, date

› 2. Data Copied from Website

- Write down the exact values you copied (latitude + at least one more, e.g., altitude, speed, etc.)
- Write 1–2 sentences in your own words: *“At the time we checked, the ISS was at latitude __ and longitude __, which means it was above ___ (roughly northern/southern hemisphere, ocean/continent, etc.).”*

› 3. Code (with comments)

- Paste your **final Python code**
- Add at least 2 **comments** explaining what each block does in *your own words* (e.g., `# this loop prints the greeting three times`).

› 4. Output (your run)

- Copy-paste your program’s actual output and its screenshot.
- At the bottom, write one short sentence: *“When we ran it, we got ___ because ____.”*

› 5. Reflection (3–4 sentences)

- Answer briefly in your own words:
 - What did you learn about Python from this task?
 - What part was easiest? What part was hardest?
 - If you had more time, what’s one improvement you’d add?

Rubric

➤ Rubric (10 points)

- Correct data (latitude + one more value, explained in own words) – 1 pts
- Code (works + comments show understanding) – 4 pts
- Condition (hemisphere check + one extra) – 2 pts
- Function & loop (defined and used correctly) – 2 pts
- Reflection (original, not copy-paste) – 1 pt