

# CSCI 3907

# Introduction to IoT and Edge Computing Applications

Prof. Kartik Bulusu, CS Dept.

Week 2 [01/27/2023]

- Overview of computing in IoT
- Differences in Cloud and Fog computing
- What is Edge Computing and Edge Compute device
- Setting up the Edge Lab
- Understanding the Python installation on the Raspberry Pi
- Some basic Python programming constructs
- First steps in Raspberry Pi programming [Blinking LEDs]



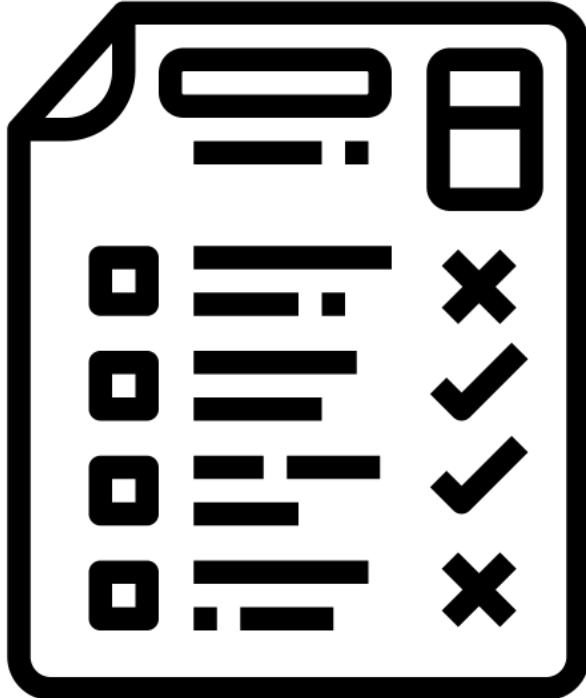
School of Engineering  
& Applied Science

Spring 2023

THE GEORGE WASHINGTON UNIVERSITY

Photo: Kartik Bulusu

# Democratizing the grading and other course policies



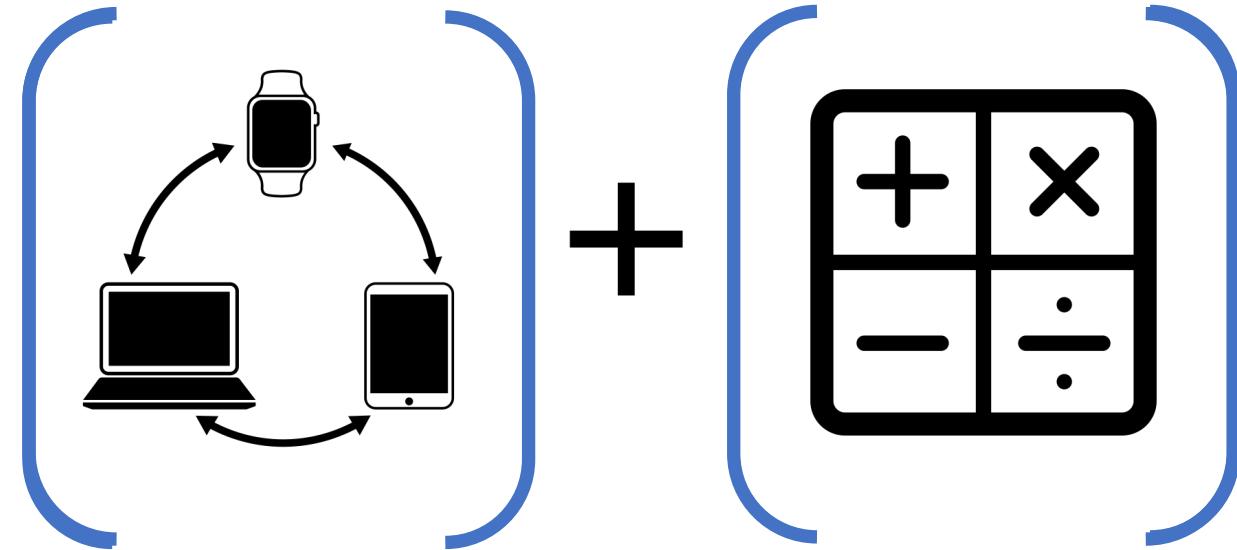
- Attendance and/or peer-to-peer and instructor interactions 10%
- In-class work and/or Weekly Quizzes 10%
- Project reports, paper reviews and in-class presentations 10%
- Python programming and other Homework 20%
- Projects (including code, demo, written report and presentation) 50%
  - 20% Individual midterm project
  - 30% Group final project

# What's the “thing”?

## Paradigm #1

- A **thing** is self-contained and only operates within the confines of its physical shell.
  - **Thing** carries out only those functions that its designer envisioned when it was fabricated.
- The **thing** contains a powerful computer inside but is completely hidden from the user.
- The **thing** has firmware (not called software).

## Paradigm #2



## Paradigm #3

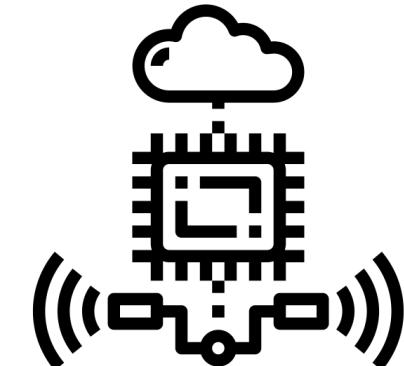
# What's an “edge” device?

## Paradigm #1

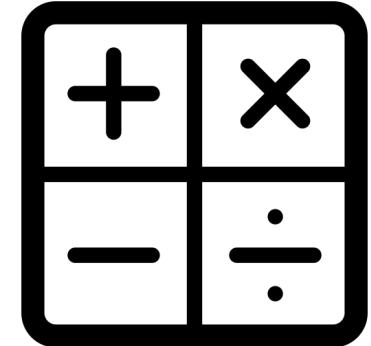
An **edge device** is any piece of hardware that controls data flow at the boundary between two networks.

- Essentially serve as network entry -- or exit -- points.
- Common functions of edge devices are the transmission, routing, processing, monitoring, filtering, translation and storage of data passing between networks.

## Paradigm #2



+



## Paradigm #3

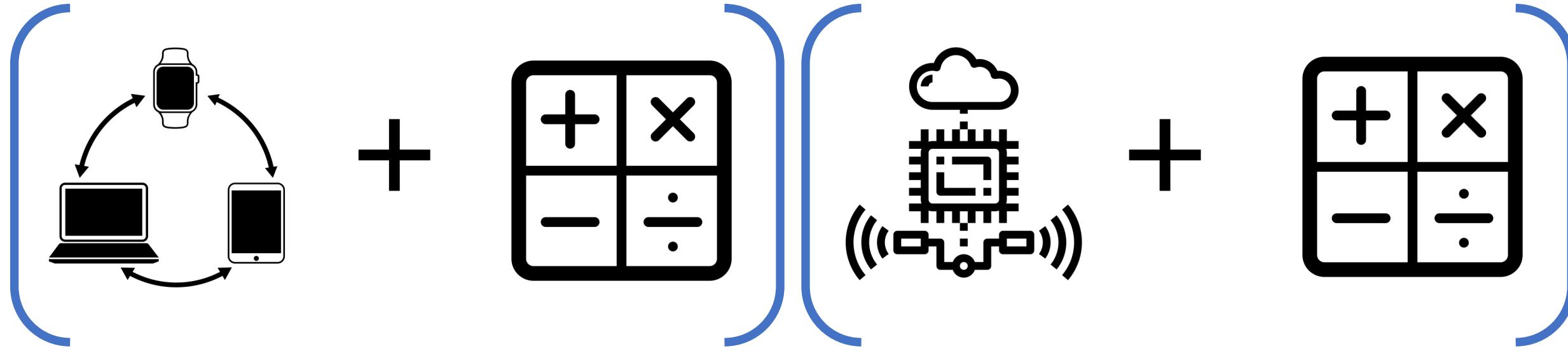
internet of things by Davo Sime from <https://thenounproject.com/browse/icons/term/internet-of-things/>

Calculator by Markus from <https://thenounproject.com/browse/icons/term/calculator/>

industrial transformation by dDara from <https://thenounproject.com/browse/icons/term/industrial-transformation>

What is IoT Edge computing?, <https://www.redhat.com/en/topics/edge-computing/iot-edge-computing-need-to-work-together>

# Overview of Computing in IoT



The Internet of Things (IoT) is made up of smart devices connected to a network—sending and receiving large amounts of data to and from other devices—which produces a large amount of data to be processed and analyzed.

Edge computing, a strategy for computing on location where data is collected or used, allows IoT data to be gathered and processed at the edge, rather than sending the data back to a datacenter or cloud.

internet of things by Davo Sime from <https://thenounproject.com/browse/icons/term/internet-of-things/>

Calculator by Markus from <https://thenounproject.com/browse/icons/term/calculator/>

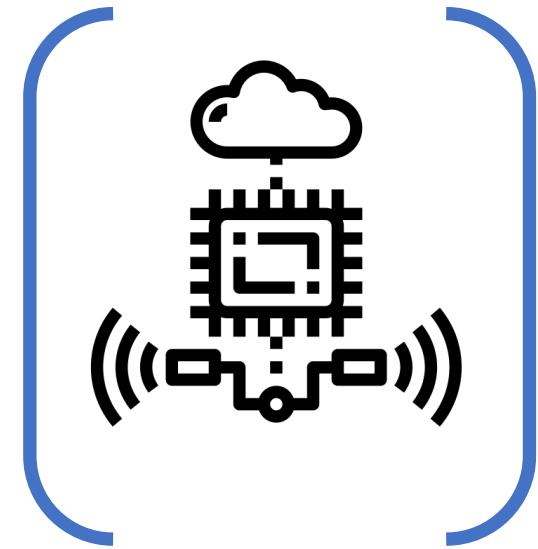
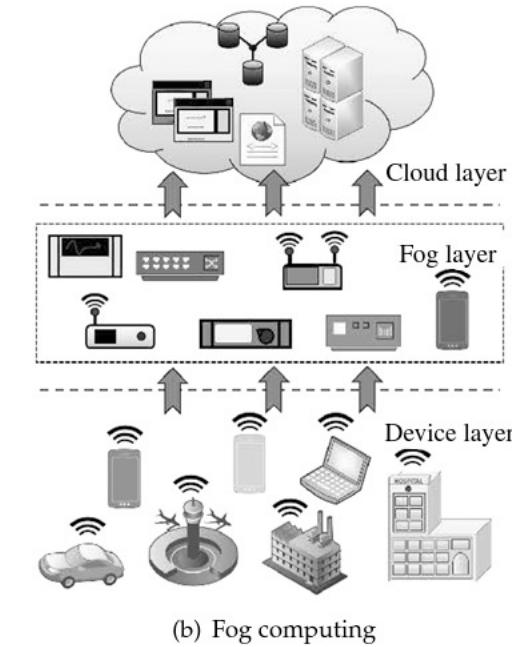
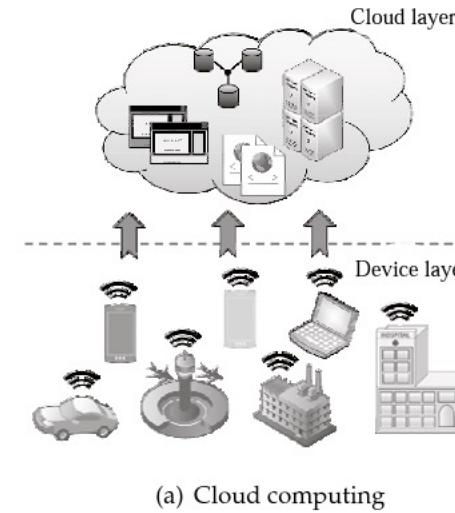
industrial transformation by dDara from <https://thenounproject.com/browse/icons/term/industrial-transformation/>

What is IoT Edge computing?, <https://www.redhat.com/en/topics/edge-computing/iot-edge-computing-need-to-work-together>

Cloud Computing by Slamet Ariyanto from <https://thenounproject.com/browse/icons/term/cloud-computing/>

S. Mishra, A. Mukherjee and A. Roy, Introduction to IoT

# Edge, Fog and Cloud Landscape



In a cloud computing model, compute resources and services are often centralized at large datacenters.

- Clouds often provide a portion of the network infrastructure required to connect IoT devices to the internet.

Edge devices require network connectivity to central locations for different purposes:

- To allow remote management,
- To receive automation instructions,
- To forward network telemetry traffic needed for analytics

## Summarizing the need for Edge Computing

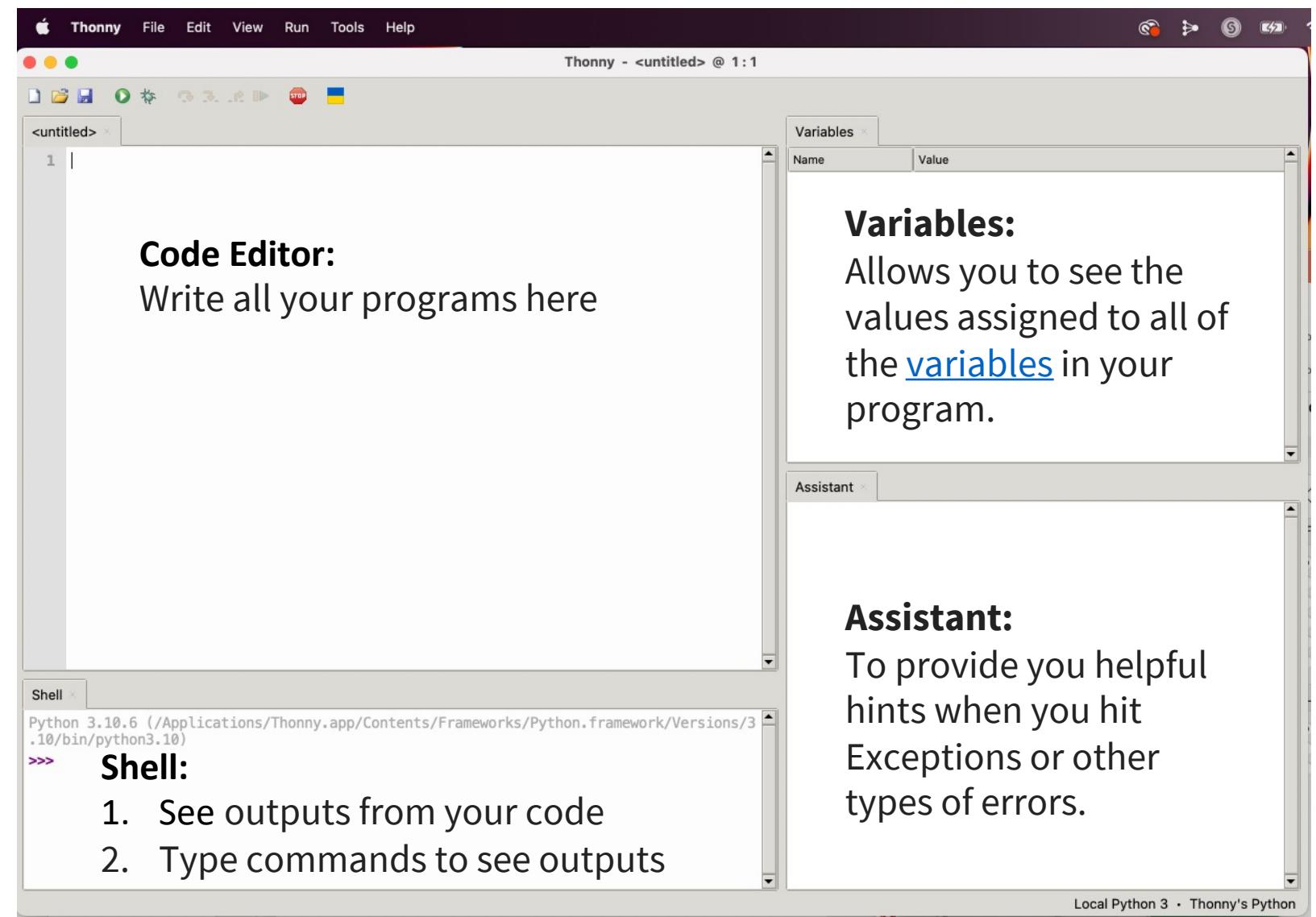
- IoT devices are low-power and getting smaller in size
- Formidable amount of information to be processed and handled.
- IoT devices need to depend on external platforms
  - Cloud
  - Fog
  - **Edge computing (more recent paradigm)**



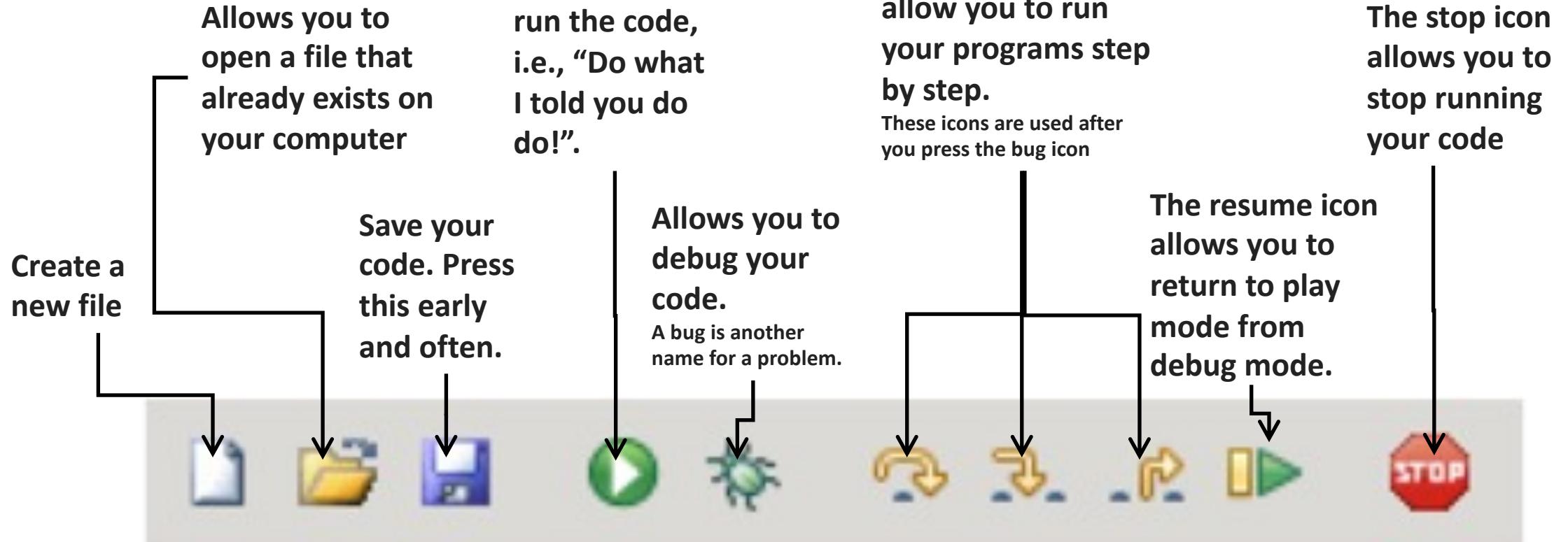
Install Thonny IDE on your laptops

## Quick peek at the Thonny-editor or integrated development environment (IDE)

Th

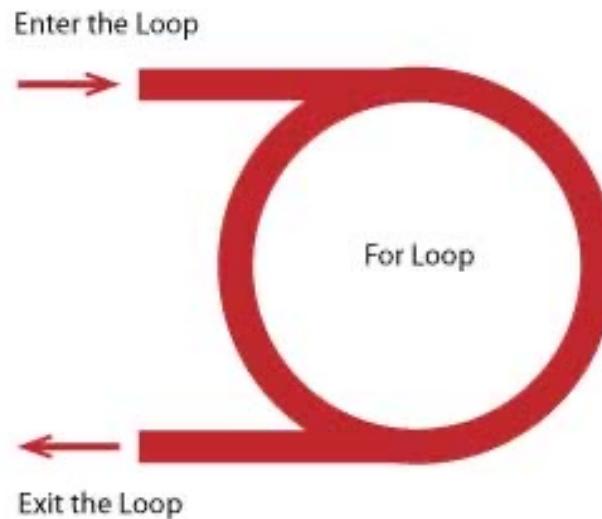


## More about Thonny-editor or integrated development environment (IDE)

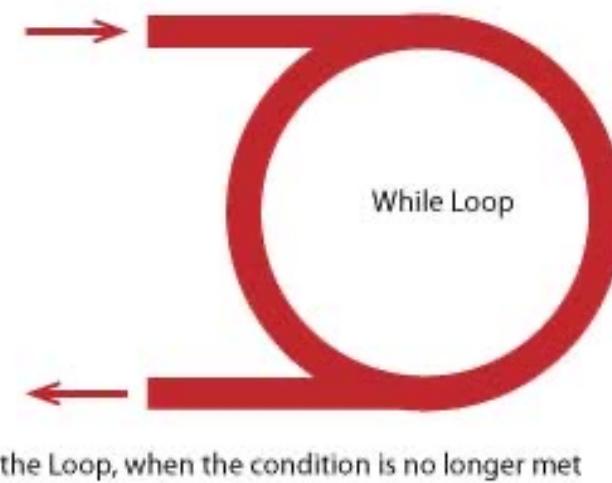


## Know some programming paradigms

(For) How many times do you want to execute a piece of code ?



(While) Some condition is met, Enter the Loop



Functions are blocks of reusable pieces of code

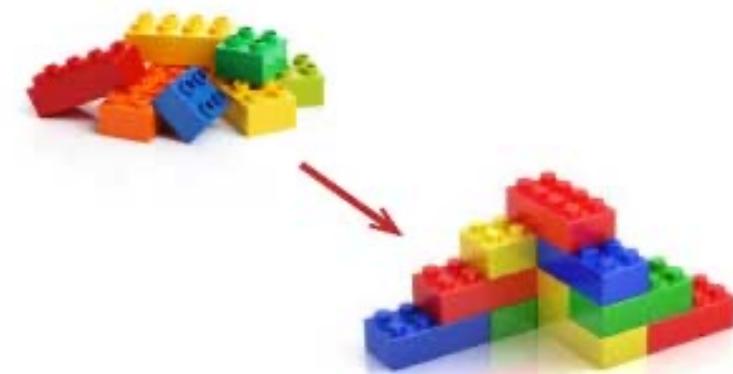


Image source:  
<https://www.thescopformommies.com/teaching-colors-to-preschoolers/lego-blocks-clip-art-ulodyk-clipart/>  
<https://www.123rf.com/clipart-vector/lego.html?utm1=tzlyzsh742&ux74>



Loops

Functions

A **function** is group of related statements that performs a specific task.

- functions**
- Break your program into smaller and modular chunks.
  - Make a program more manageable and modular as a program grows larger and larger
  - Avoid repetition and make the code reusable

- print()**
- Example of a function you just learned
  - Built-in function in Python

Functions are blocks of reusable pieces of code

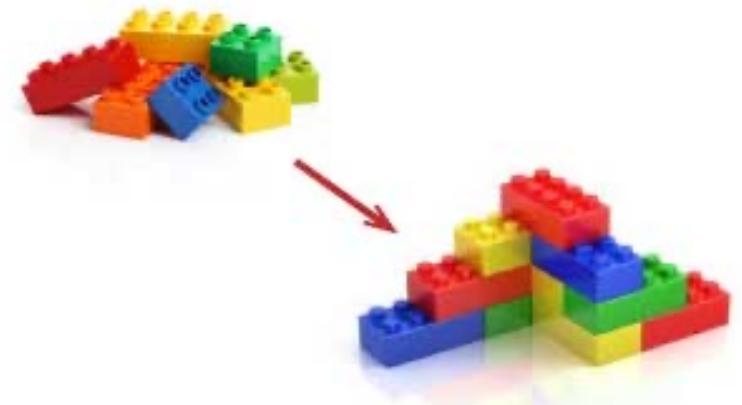
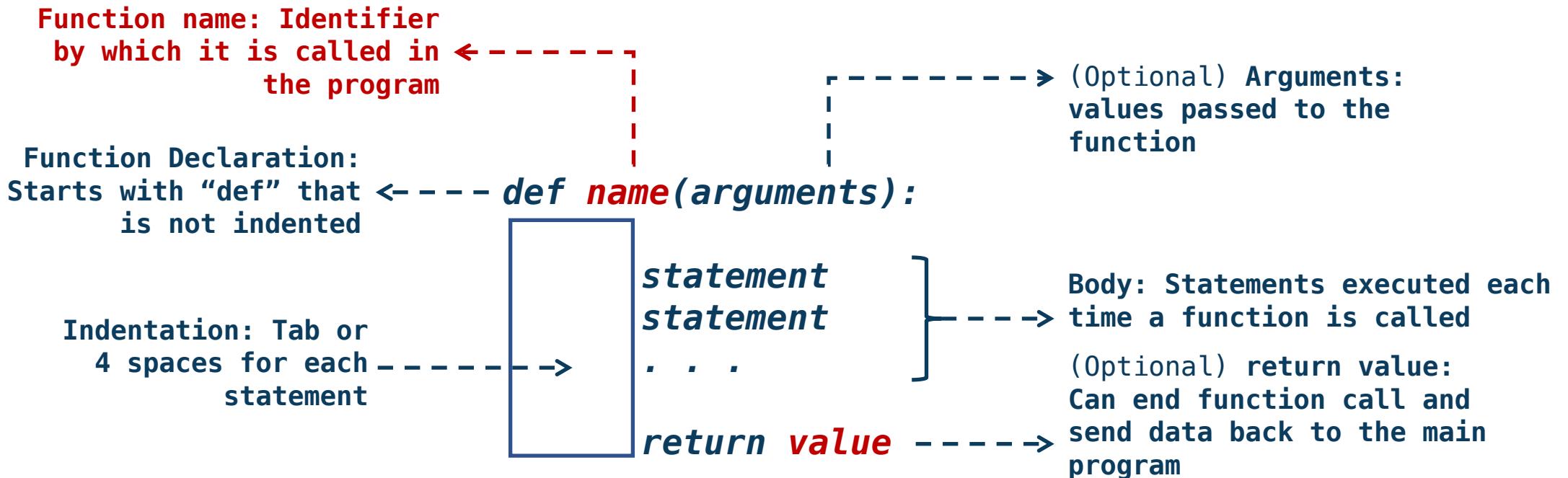


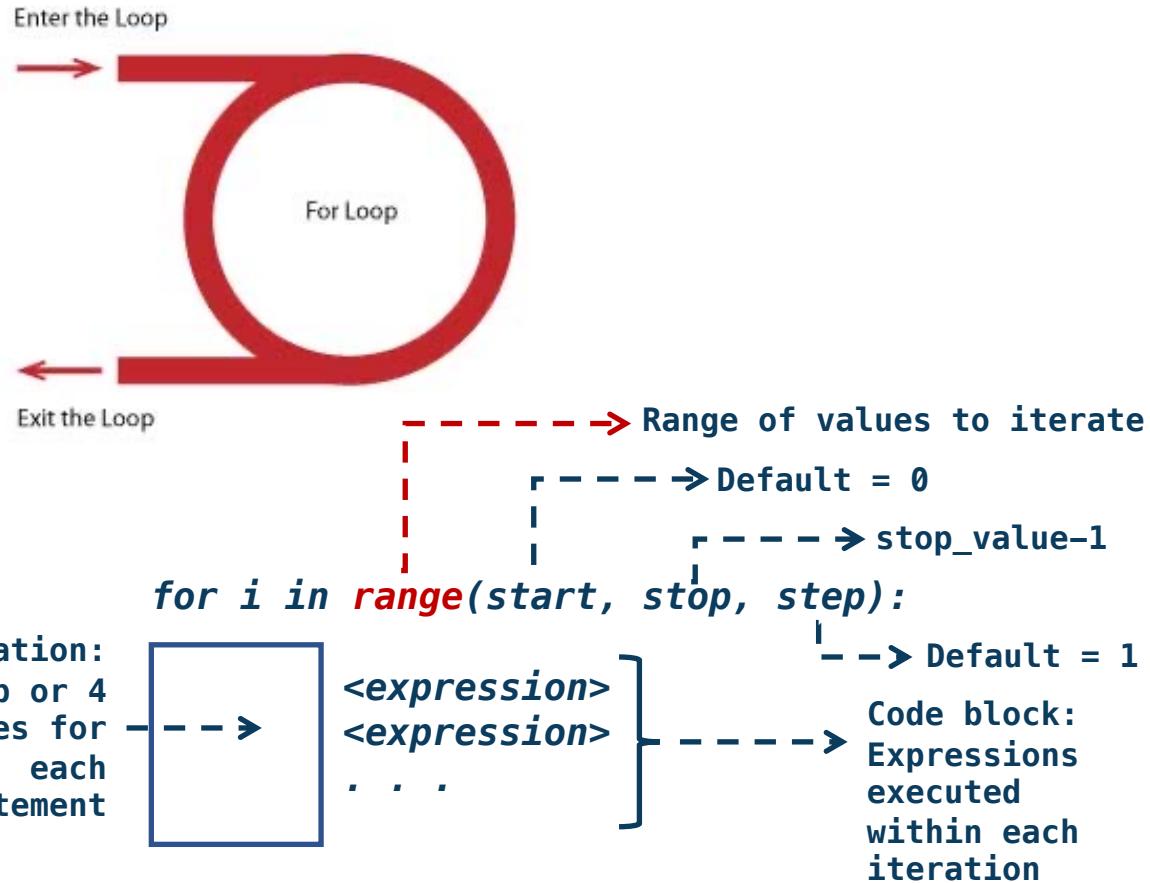
Image source:  
<https://www.thescoopformommies.com/teaching-colors-to-preschoolers/lego-blocks-clip-art-ulodyc-clipart/>  
<https://www.123rf.com/clipart-vector/lego.html?ttm=1zlyzh7426ux74>

## Skeleton of a user-defined function

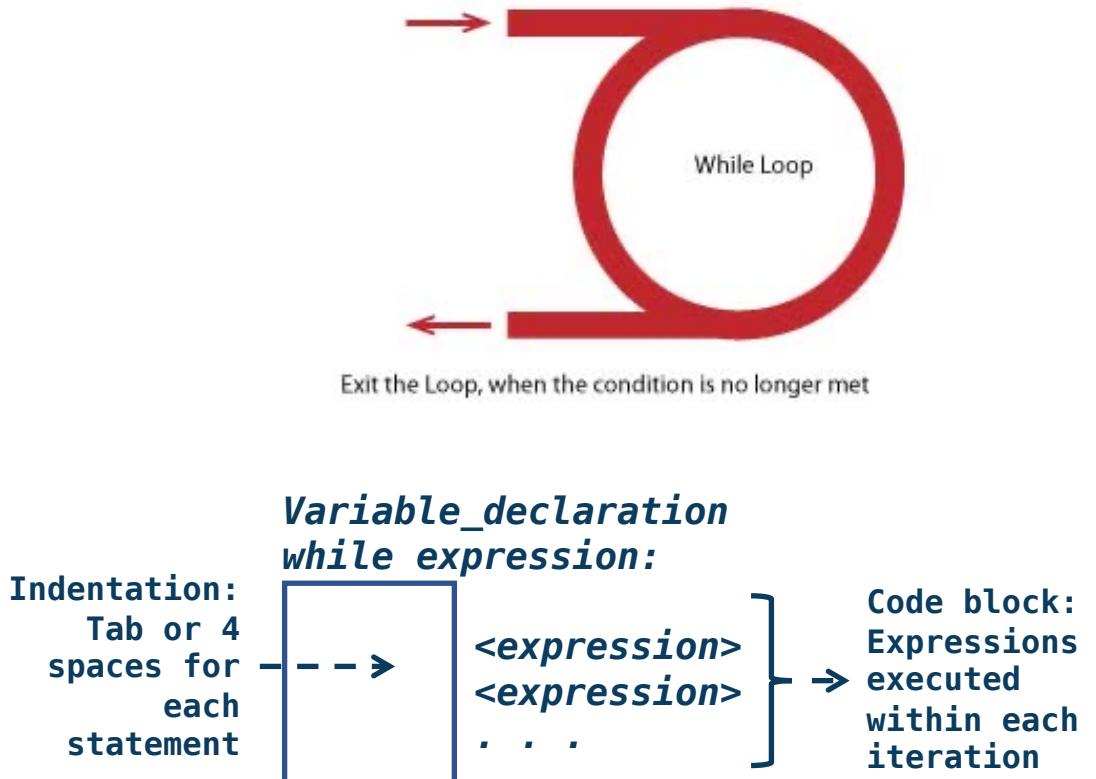


# Loops: A first look at the skeletons

(For) How many times do you want to execute a piece of code ?



(While) Some condition is met, Enter the Loop



## Skeleton of a typical Python program written for Raspberry Pi

```
import library1 as name1  
import RPi.GPIO as GPIO  
import time
```

} --> Import libraries that are relevant for interaction with the Raspberry Pi hardware such as GPIO pins, camera ports etc.

```
GPIO.setmode(GPIO.BOARD)  
GPIO.setup(12, GPIO.OUT)
```

} --> Set up GPIO pins as data outputs or inputs for sensors and actuators

```
def name(arguments):  
    statement  
    statement  
    ...  
    return value
```

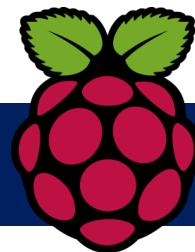
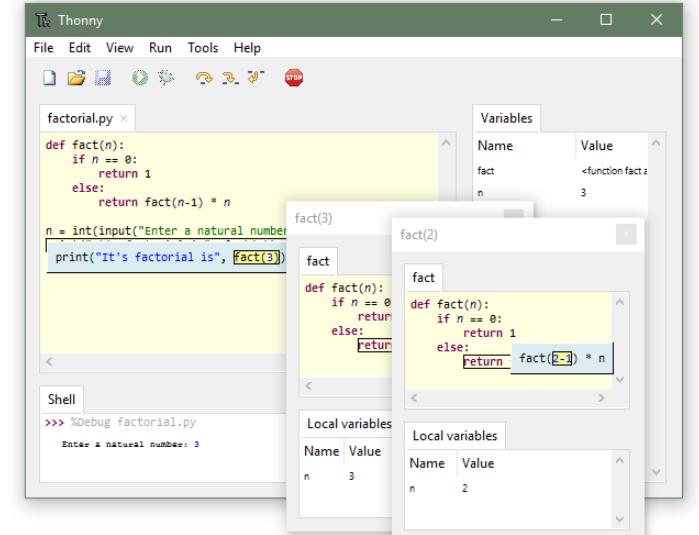
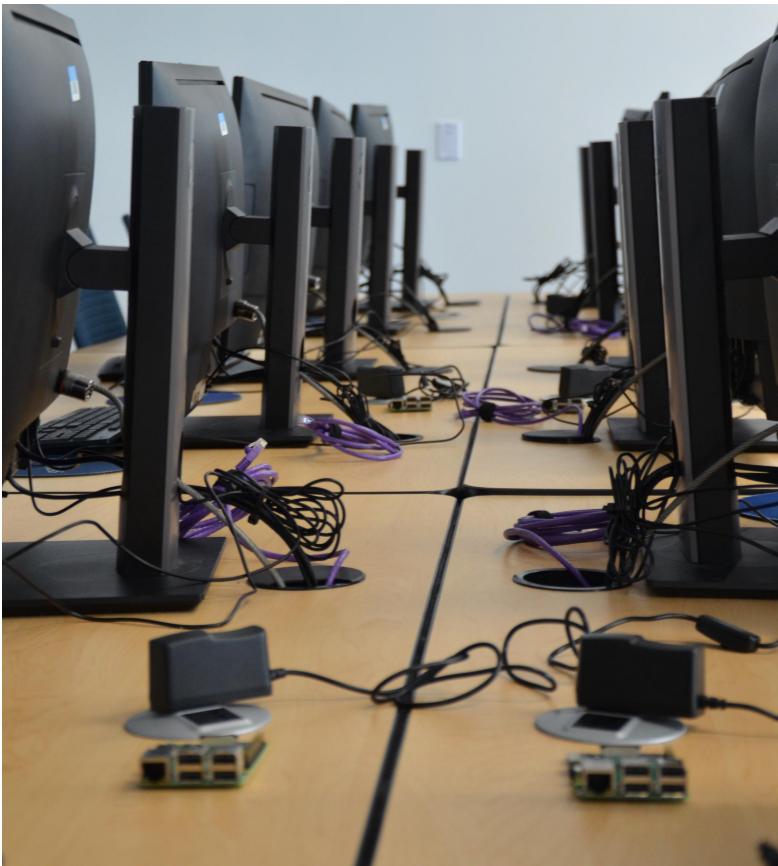
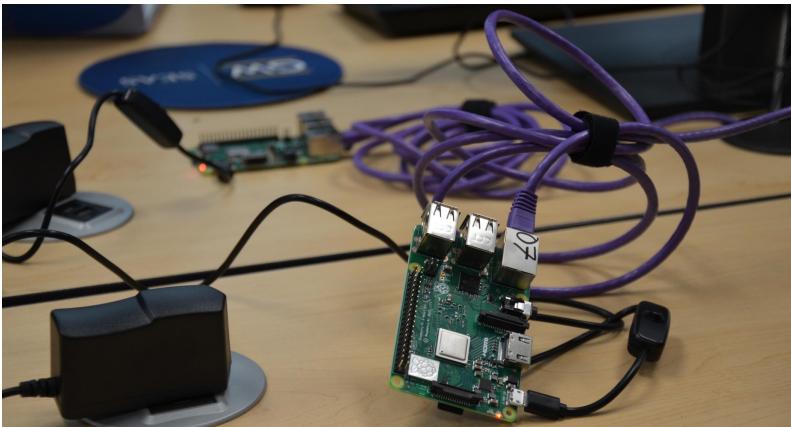
} --> Create user-defined functions to modularize your code and make it easy to work.  
Create user-defined functions to release the GPIO pins

## Setting up the Edge-lab each week

Sources:

Thonny (IDE): <https://en.wikipedia.org/wiki/Thonny>

Thonny: <https://thonny.org>



# Tompkins 201: Laboratory set up

## STEP [-1]:

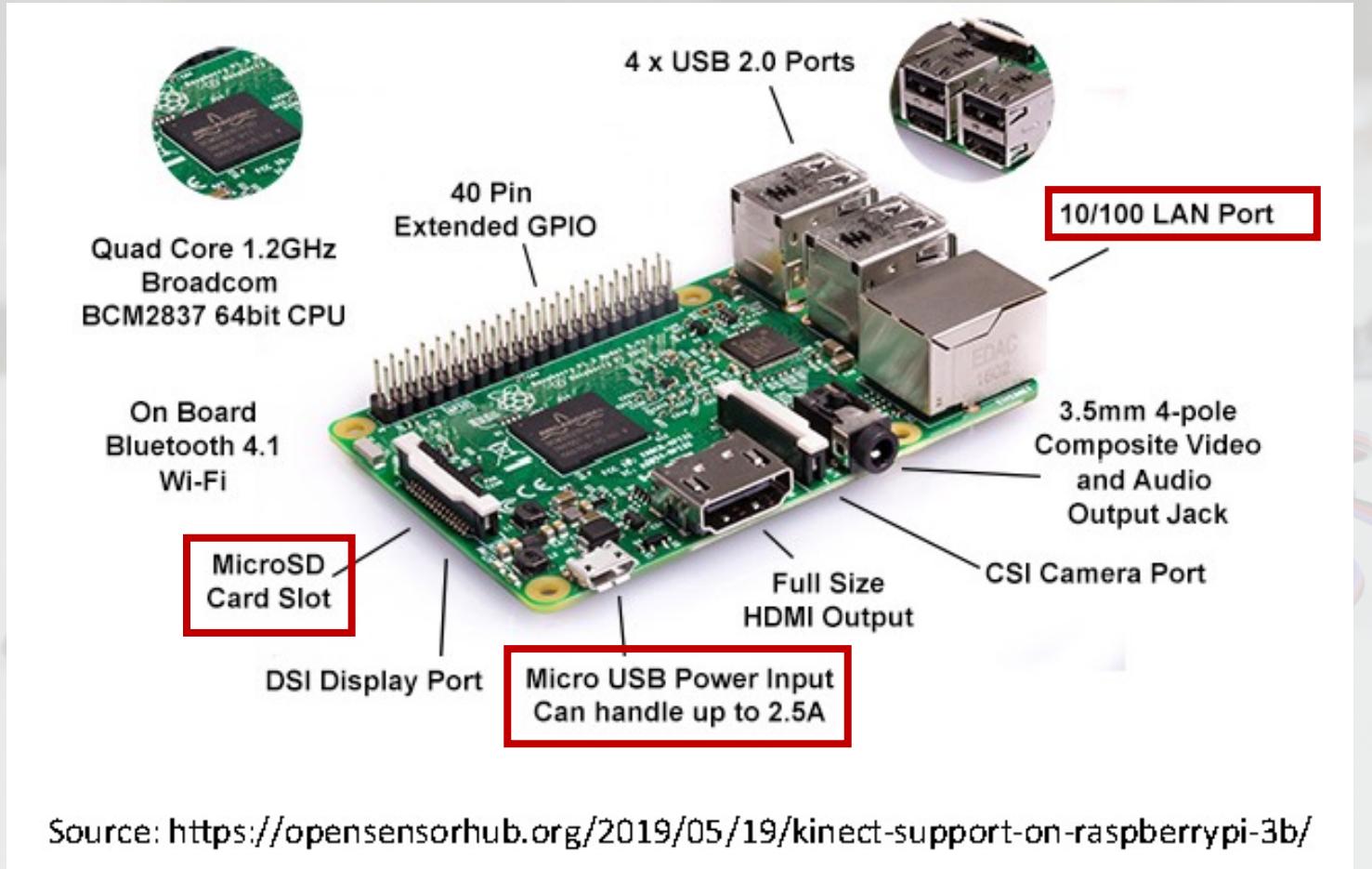
Familiarize yourself with TH201 classroom layout



- 6 clusters of desks
- Each cluster 2-3 laptop power cables (Dell Inspiron)
- Each cluster has 2-3 blue colored ethernet cables specifically connectivity
- Each cluster should have surge protector (on the desk)

## STEP [0]:

Familiarize yourself with the Raspberry Pi 3B+ (RPi) layout

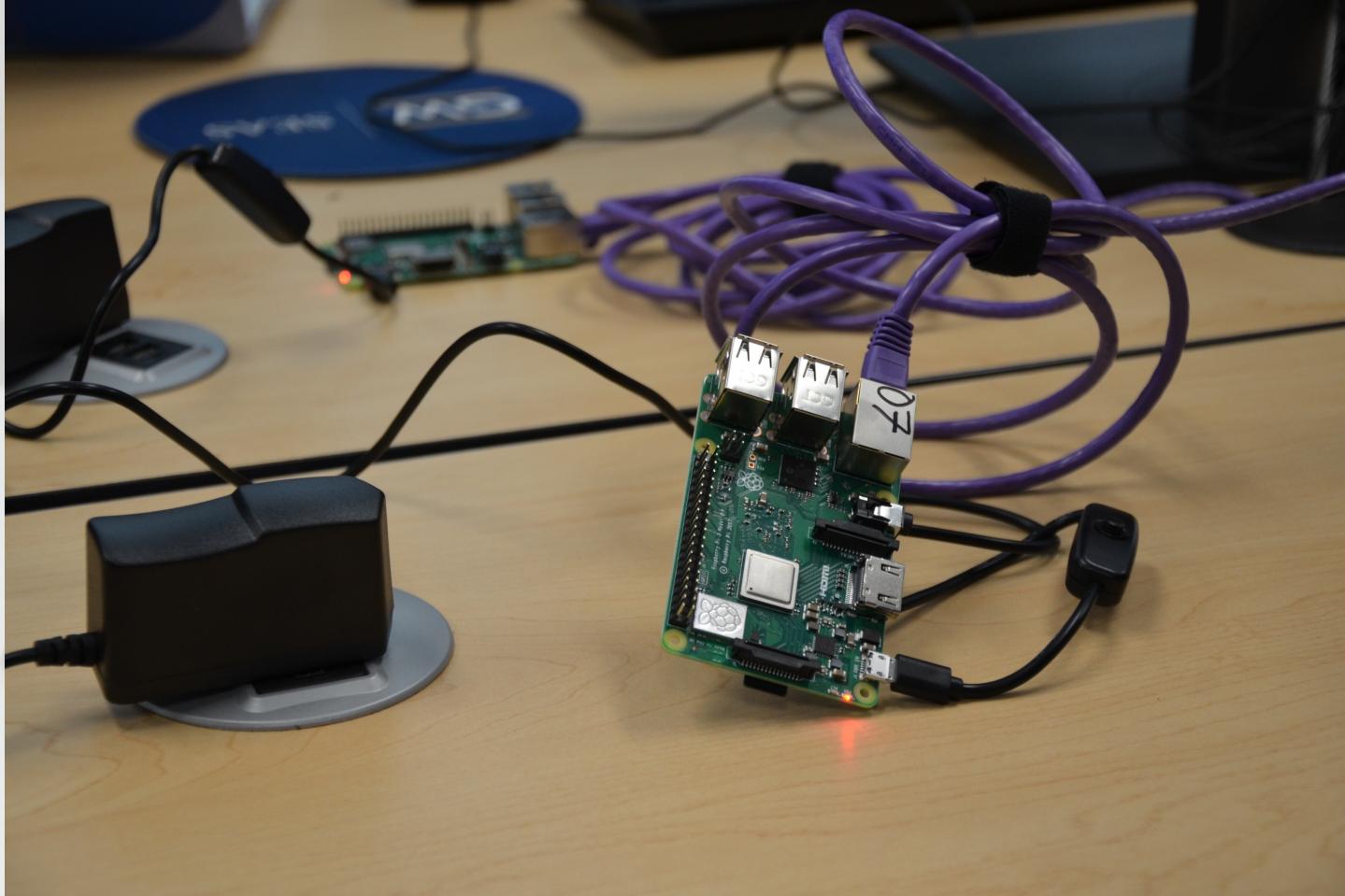


Source: <https://opensensorhub.org/2019/05/19/kinect-support-on-raspberrypi-3b/>

- Each RPi will come with a preconfigured OS in a microSD card
- Each RPi will have a microSD card already installed
- Each RPi is assigned a unique alpha-numeric name (e.g., Pi07, Pi152 etc)
- To power, boot up and test the RPi you will need
  - MicroUSB power cord
  - Ethernet cord for 10/100 LAN port
  - IP address
  - Login and password for the RPi
  - Remote desktop feature enabled in your computer

## STEP [1]:

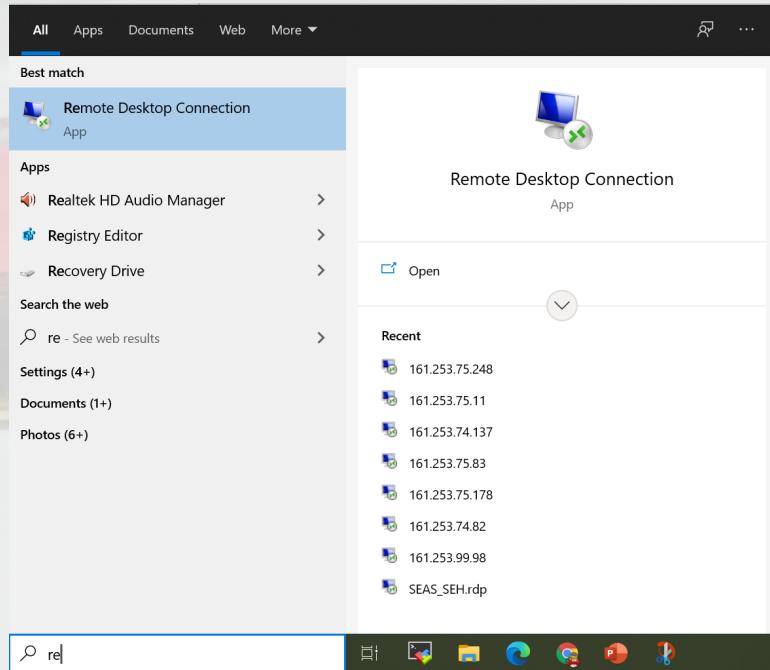
Connect the RPis to each desk power outlet as shown



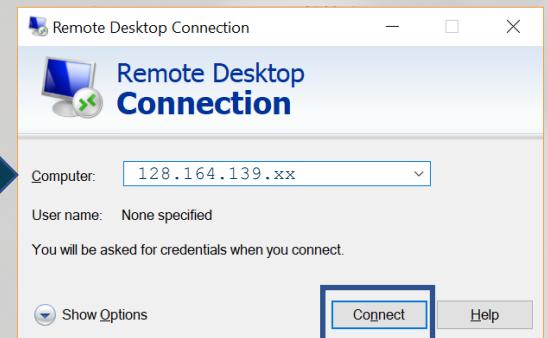
- Make sure there is a microSD card installed in the RPi
- Connect the RPi using the microUSB cable provided
- Connect the purple colored ethernet cables specifically for RPi connections
- LEDs on the RPi will start blinking indicating that it is booting up

## STEP [2]: Access the RPi in the Edge-lab

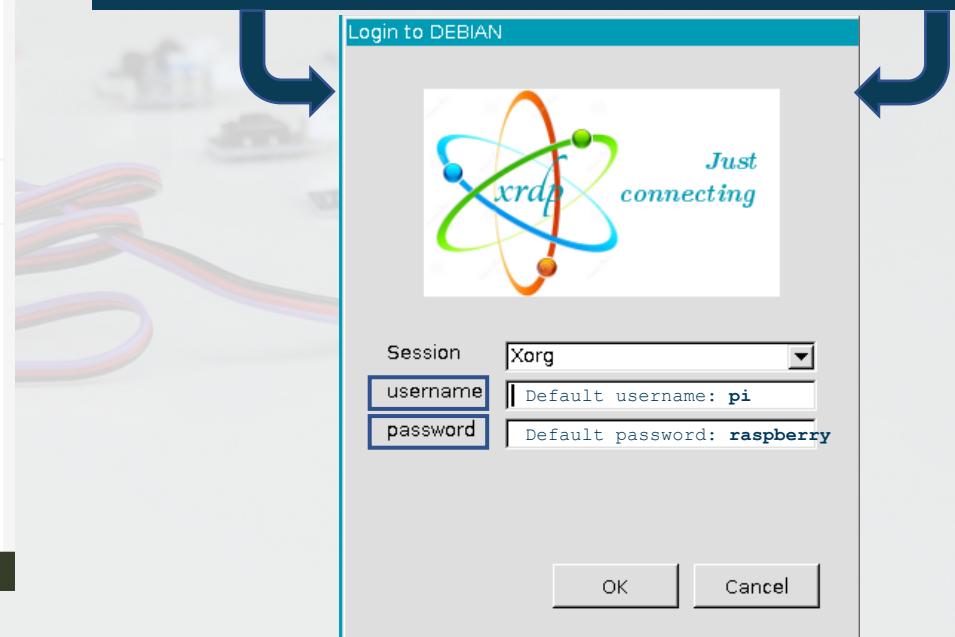
2.1 Open up remote desktop connection  
(Windows 10 OS shown below)



2.2 Enter the IP address and hit  
“Connect”  
(Windows 10 OS shown here)



2.3 Once you are connected you will  
• See xrdp-interface  
• Type in the username and password

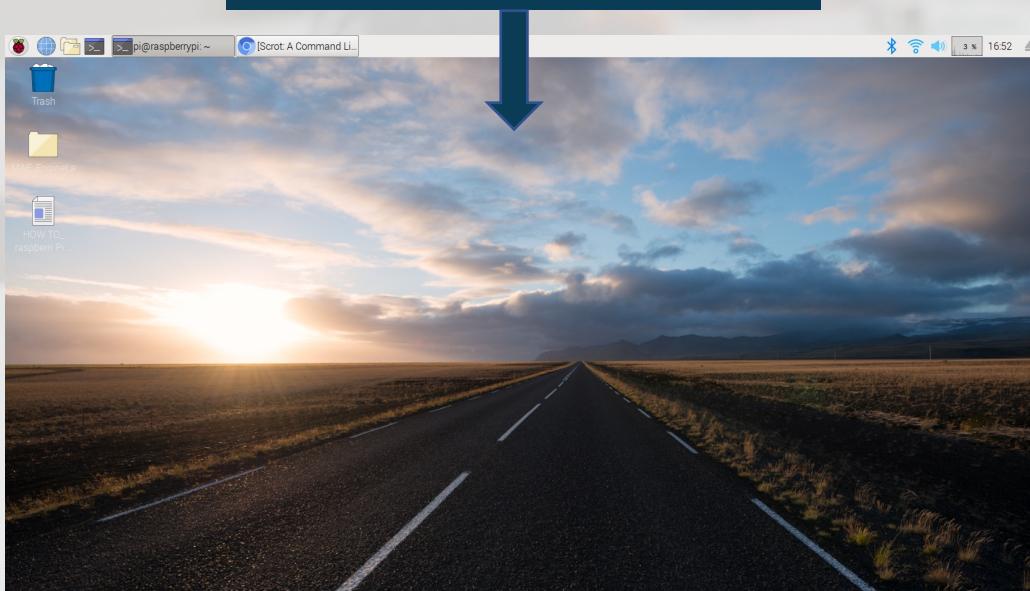


Source: [https://upload.wikimedia.org/wikipedia/commons/f/f1/XRDP\\_Screenshot.png](https://upload.wikimedia.org/wikipedia/commons/f/f1/XRDP_Screenshot.png)

## STEP [3]:

Now that you accessed the RPi...

You will see a screen like the one shown below

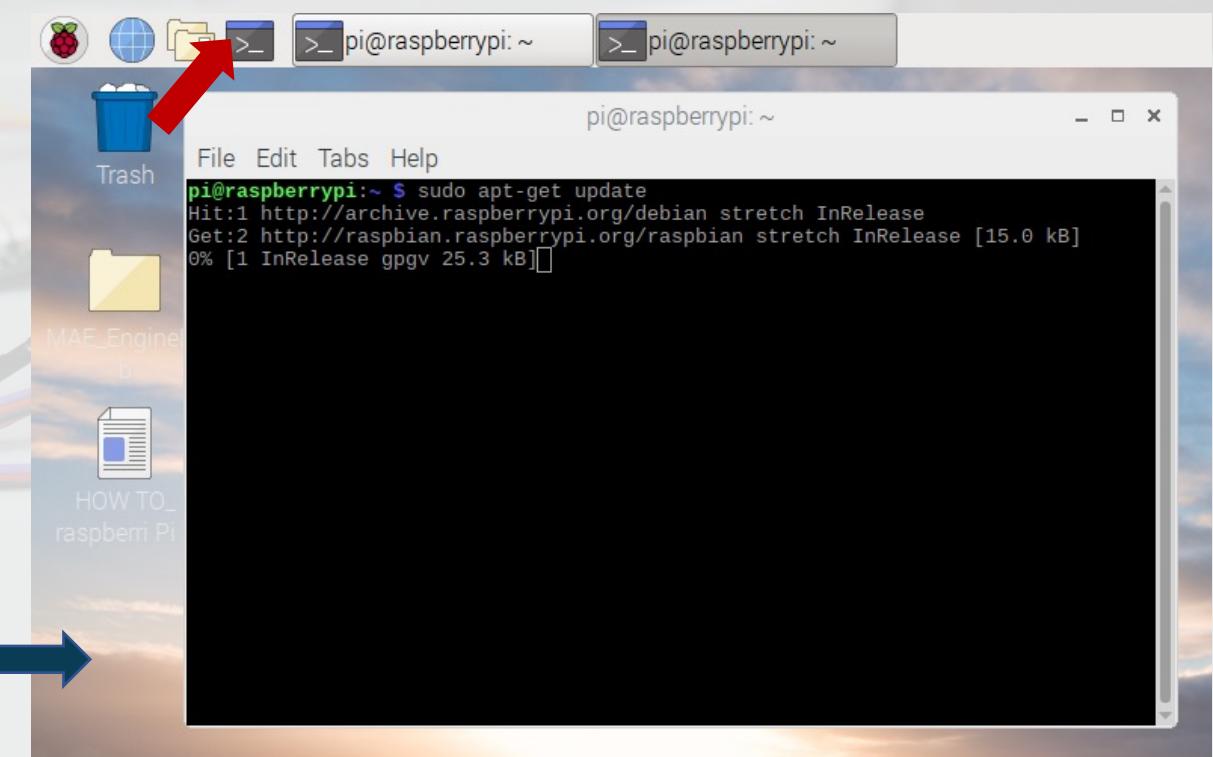


### 3.2 Testing is complete when you get to this step.

- Students should get the RPis to this step before the laboratory modules begin.

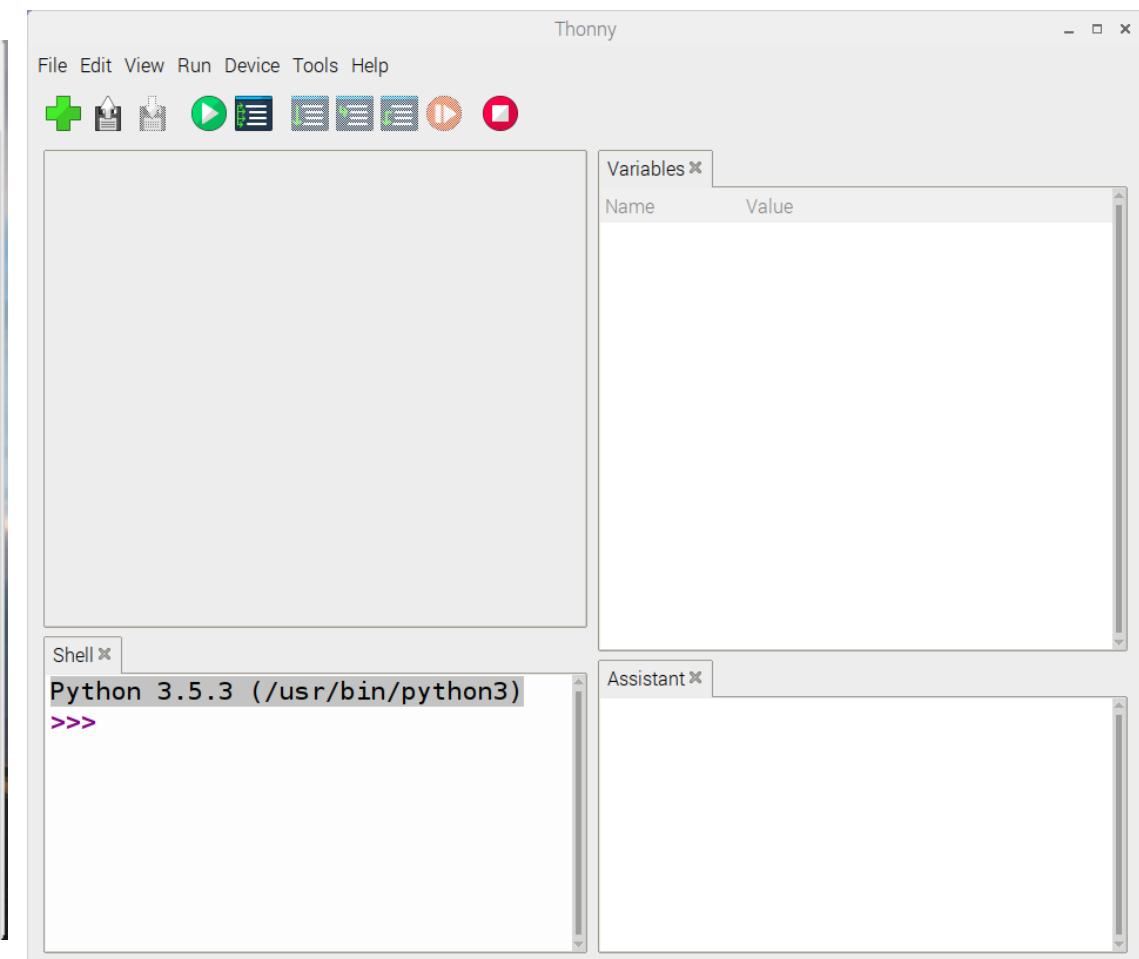
### 3.1 Click on terminal(shown with a red arrow below)

- At the prompt type: **sudo apt-get update**
- Wait for the updates to complete
- Then type: **sudo apt-get upgrade**
- If you get the following prompt
  - **Do you want to continue [Y/n]**
  - Type: **y**
  - And hit “Enter” on your keyboard and let the upgrades complete



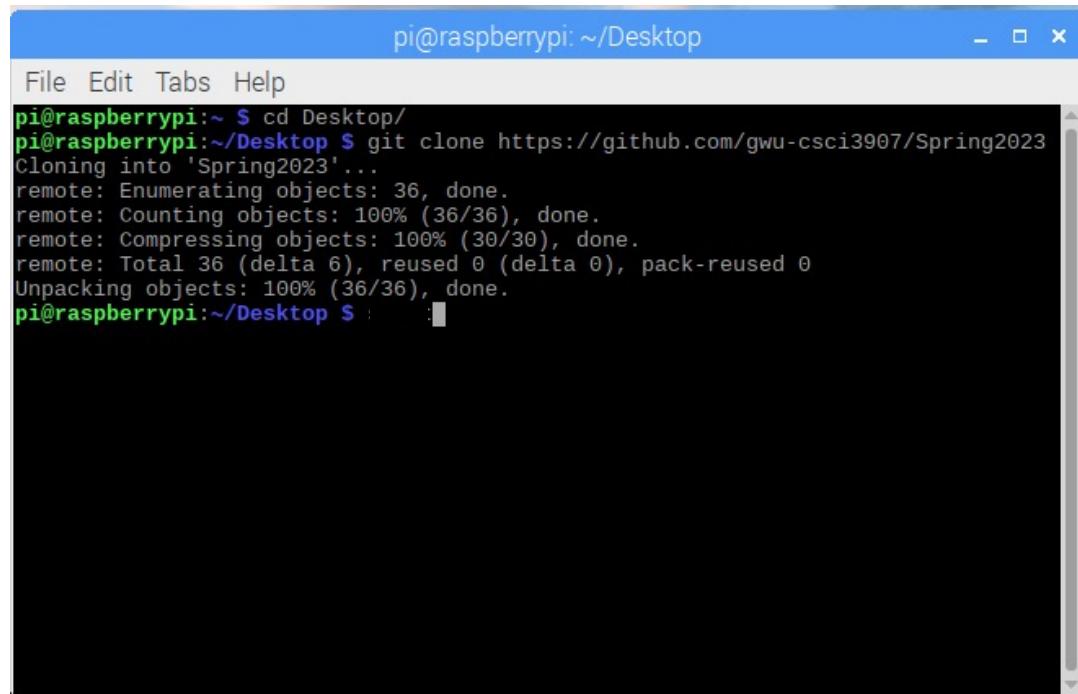
# Understanding your Python installation on the Raspberry Pi

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 9 (stretch)"
NAME="Raspbian GNU/Linux"
VERSION_ID="9"
VERSION="9 (stretch)"
VERSION_CODENAME=stretch
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $ python --version
Python 2.7.13
pi@raspberrypi:~ $ python3 --version
Python 3.5.3
pi@raspberrypi:~ $
```



## Downloading folders from course git-repository

```
git clone https://github.com/gwu-cssi3907.github.io/Spring2023.git
```



A terminal window titled "pi@raspberrypi: ~/Desktop". The command "git clone https://github.com/gwu-cssi3907/Spring2023" is run, cloning the repository into the "Spring2023" directory. The output shows the progress of cloning, including object enumeration, counting, compressing, and unpacking.

```
pi@raspberrypi:~ $ cd Desktop/  
pi@raspberrypi:~/Desktop $ git clone https://github.com/gwu-cssi3907/Spring2023  
Cloning into 'Spring2023'...  
remote: Enumerating objects: 36, done.  
remote: Counting objects: 100% (36/36), done.  
remote: Compressing objects: 100% (30/30), done.  
remote: Total 36 (delta 6), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (36/36), done.  
pi@raspberrypi:~/Desktop $ :
```

