Sri Indu
College of Engineering & Technology
UGC Autonomous Institution
Recognized under 2(f) & 12(B) of UGC Act 1956,
NAAC, Approved by AICTE &
Permanently Affiliated to JNTUH
Estd 2001

NAAC
NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL

HANDOUT

First Year ECE Semester II

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ACADEMIC YEAR 2022-23

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

HANDOUT- INDEX

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, Affiliated to JNTUH)
Sheriguda (V), Ibrahimpatnam (M), R.R. Dist-501510

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTION VISION

To be a premier Institution in Engineering & Technology and Management with competency, values and social consciousness

INSTITUTION MISSION

IM$_1$ Provide high quality academic programs, training activities and research facilities.

IM$_2$ Promote Continuous Industry-Institute interaction for employability, Entrepreneurship, leadership and research aptitude amongstakeholders.

IM$_3$ Contribute to the economical and technological development of the region, state and nation.

DEPARTMENT VISION

To be a centre of excellence in Electronics and Communication Engineering Education to produce professionals for ever-growing needs of society

DEPARTMENT MISSION

The Department has following Missions:

DM1: To promote and facilitate student- centric learning.

DM2: To involve in activities that enable overall development of stakeholders.

DM3: To provide holistic environment with state-of-art facilities for students to develop solutions for various social needs.

DM4: Organize trainings in embedded systems with Industry interaction.

PROGRAM EDUCATIONAL OBJECTIVES(PEO'S)

PEO1:	**Higher Degrees & Professional Employment:** Graduates with ability to pursue career in core industries or higher studies in reputed institution
PEO2:	**Domain Knowledge:** Graduates with ability to apply professional knowledge/skills to design and develop product or process.
PEO3:	**Engineering Career:** Graduates with excellence in Electronics and Communication Engineering along with effective inter-personnel skills
PEO4:	**Lifelong Learning:** Graduates equipped with skills in recent technologies and be receptive to attain professional competence through life-long learning.

PROGRAM SPECIFIC OUTCOMES(PSO'S)

PSO1:	To nurture and empower the SICET-ECE students strong in practical, technical and research domains in the areas of Signal/Image processing, VLSI and wireless Communication.
PSO 2:	To design and develop a prototype system that will incorporate user requirements using modern devices and emerging technology for industry automations.
PSO 3:	To make the SICET-ECE students as successful **industry ready engineers** by impartingessential **interpersonal skillsand widespread exposure** on multi-disciplinary technologies.

PROGRAM OUTCOMES (PO'S)

PO1	**Engineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem Analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design / Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COs MAPPING WITH POs & PSOs

I YEAR SEMESTER – I (REGULATION – R22)

ACADEMIC YEAR: 2022-2023

Course Code & Name: **(R22CSE1224) APPLIED PYTHON PROGRAMMING LAB**

Course outcomes: After learning the contents of this paper the student must be able to,

Course Name	Course Outcomes
C126.1	Build basic programs using constructs fundamental programming(L3-Apply)
C126.2	Write and execute python codes for different applications.(L3-Apply)
C126.3	Illustrate the programs using modular approach file I/O ,python Standard library (L4-Analyze)
C126.4	Capable to implement on hardware boards. (L4-Analyze)

CO's	**Program Outcomes**														
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO-1	2	2	2	-	2	-	-	-	1	-	-	-	2	1	-
CO-2	1	2	2	-	2	-	-	-	-	-	-	-	2	1	-
CO-3	1	2	2	1	2	-	-	-	1	-	-	-	2	1	-
CO-4	1	2	2	1	1	-	-	-	1	-	-	-	2	1	-
C126															

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY
(An Autonomous Institution under UGC, New Delhi)
Recognized under 2(f) and 12(B) of UGC Act 1956
NBA Accredited. Approved by AICTE and Permanently affiliated to JNTUH
Sheriguda (V), Ibrahimpatnam, R.R.Dist, Hyderabad - 501 510

D4

BR-22

Lr.No.SICET/AUTO/DAE/BR-22/Academic Cal./655/2022 Date: 27.10.2022

I B.TECH. ACADEMIC CALENDAR
ACADEMIC YEAR : 2022-2023

Dr.G. SURESH,
Principal,

To,
All the HODs
Sir,

Sub: SICET (Autonomous) - Academic & Evaluation - Academic Calendar for **I B.Tech - I & II Semester** for the academic year **2022-23** – Reg.

****

The approved Academic Calendar for **I B.Tech – I & II Semester** for the academic year **2022-23** is given below:

I SEMESTER

S.NO.	EVENT	PERIOD	DURATION
1.	Induction & Orientation Programme	**03.11.2022**	
2.	1St Spell of Instructions for covering First Two and a half Units	03.11.2022 – 28.12.2022	8 Weeks
3.	I Mid Examinations	29.12.2022 – 04.01.2023	1 Week
4.	Submission of I Mid Term Examination Marks to the Autonomous Section on or before	10.01.2023	
5.	2nd Spell of Instructions for covering Remaining Two and a half Units	05.01.2023 – 02.03.2023	8 Weeks
6.	II Mid Examinations	03.03.2023 – 09.03.2023	1 Week
7.	Preparation & Practical Examinations and Remidial Mid Test (RMT)	10.03.2023 – 16.03.2023	1 Week
8.	Submission of II Mid Term Examination Marks Marks to the Autonomous Section on or before	16.03.2023	
9.	I Semester End Examinations	17.03.2023 – 01.04.2023	2 Weeks
Commencement of Class-Work for I B.Tech - II Semester 03.04.2023			

II SEMESTER

S.NO.	EVENT	PERIOD	DURATION
1.	Commencement of II Sem Class Work	**03.04.2023**	
2.	1st Spell of Instructions for covering First Two and a half Units (Including Summer Vacation)	03.04.2023 – 10.06.2023	10 Weeks
	Summer Vacation	15.05.2023 – 27.05.2023	2 Weeks
3.	I Mid Examinations	12.06.2023 – 17.06.2023	1 Week
4.	Submission of I Mid Term Examination Marks to the Autonomous Section on or before	23.06.2023	
5.	2nd Spell of Instructions for covering Remaining Two and a half Units	19.06.2023 – 12.08.2023	8 Weeks
6.	II Mid Examinations	14.08.2023 – 19.08.2023	1 Week
7.	Preparation & Practical Examinations and Remidial Mid Test (RMT)	21.08.2023 – 26.08.2023	1 Week
8.	Submission of II Mid Term Examination Marks Marks to the Autonomous Section on or before	26.08.2023	
9.	II Semester End Examinations	28.08.2023 – 09.09.2023	2 Weeks
Commencement of Class Work for II B.Tech – I Semester - 11.09.2023			

ACE **CE** **DEAN** **PRINCIPAL**

Copy to all the Heads of the Depts. and AO.

CONTROLLER OF EXAMINATIONS
Sri Indu College of Engineering & Technology
(An Autonomous Institution under JNTUH)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510.

DIRECTOR
(Academic Audit)
Sri Indu College of Engineering & Technology
Sheriguda, IBP, R.R. Dist-501510.

PRINCIPAL
Sri Indu College of Engineering & Technology
(An Autonomous Institution Under JNTUH)
Sheriguda (V), Ibrahimpatnam, R.R. Dist-501510.

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)

DEPARTMENT OF HUMANITIES & SCIENCES

DEPARTMENT CALENDAR – 2022-2023 (I B.Tech – II SEMESTER)

DAYS	APRIL - 23						JULY-2023					
SATURDAY	1						1					
SUNDAY	2		MAY-2023				2					
MONDAY	3	COMMENCEMENT OF II SEM CLASS WORK	1				3		AUGUST-2023			
TUESDAY	4		2				4		1			
WEDNESDAY	5		3			JUNE-2023	5		2			
THURSDAY	6		4		1		6		3		SEPTEMBER-2023	
FRIDAY	7		5		2		7		4		1	II SEMESTER END EXAMINATIONS
SATURDAY	8		6		3		8		5		2	
SUNDAY	9	HOLIDAY	7	HOLIDAY	4	HOLIDAY	9	HOLIDAY	6	HOLIDAY	3	HOLIDAY
MONDAY	10		8		5		10		7		4	II SEMESTER END EXAMINATIONS
TUESDAY	11		9		6		11		8		5	
WEDNESDAY	12		10		7		12		9		6	
THURSDAY	13		11		8		13		10		7	
FRIDAY	14		12		9		14		11		8	
SATURDAY	15		13		10		15		12		9	
SUNDAY	16	HOLIDAY	14	HOLIDAY	11	HOLIDAY	16	HOLIDAY	13	PREPARATION & LAB END EXAMINATIONS	10	HOLIDAY
MONDAY	17		15		12	MID I EXAM	17		14	MID II EXAM	11	
TUESDAY	18		16		13	MID I EXAM	18		15	MID II EXAM	12	
WEDNESDAY	19		17		14	MID I EXAM	19		16	MID II EXAM	13	
THURSDAY	20		18		15	MID I EXAM	20		17	MID II EXAM	14	
FRIDAY	21		19		16	MID I EXAM	21		18	MID II EXAM	15	
SATURDAY	22		20		17	MID I EXAM	22		19	MID II EXAM	16	
SUNDAY	23	HOLIDAY	21	HOLIDAY	18	HOLIDAY	23	HOLIDAY	20	HOLIDAY	17	HOLIDAY
MONDAY	24		22		19		24		21		18	
TUESDAY	25		23		20		25		22		19	
WEDNESDAY	26		24		21		26		23		20	
THURSDAY	27		25		22		27		24		21	
FRIDAY	28		26		23	I MID MARKS SUBMISSION	28		25		22	
SATURDAY	29		27		24		29		26	II MID MARKS SUBMISSION	23	
SUNDAY	30	HOLIDAY	28	HOLIDAY	25	HOLIDAY	30	HOLIDAY	27	HOLIDAY	24	HOLIDAY
MONDAY	29		29		26		31		28	II SEMESTER END EXAMINATIONS	25	
TUESDAY	30		30		27				29		26	
WEDNESDAY			31		28				30		27	
THURSDAY					29				31		28	
FRIDAY					30						29	
SATURDAY											30	

APPLIED PYTHON LABORATORY

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - I Year – II Semester	**L**	**T**	**P**	**C**
	0	**1**	**2**	**2**

(R22CSE1224) APPLIED PYTHON PROGRAMMING LABORATORY

Course Outcomes: Upon completing this course, the students will be able to

1. Build basic programs using fundamental programming constructs
2. Write and execute python codes for different applications
3. Capable to implement on hardware boards

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	1	1	1	-	-	1	-	1	1
CO2	2	3	2	1	1	2	-	-	1	-	1	1
CO3	2	3	2	1	1	2	-	-	1	-	1	1

LIST OF EXPERIMENTS:

Cycle - 1

1. Downloading and Installing Python and Modules
   a) Python 3 on Linux
   Follow the instructions given in the URL https://docs.python-guide.org/starting/install3/linux/
   b) Python 3 on Windows
   Follow the instructions given in the URL https://docs.python.org/3/using/windows.html (Please remember that Windows installation of Python is harder!)
   c) pip3 on Windows and Linux
   Install the Python package installer by following the instructions given in the URL https://www.activestate.com/resources/quick-reads/how-to-install-and-use-pip3/
   d) Installing numpy and scipy
   You can install any python3 package using the command pip3 install <packagename>
   e) Installing jupyterlab
   Install from pip using the command pip install jupyterlab
2. Introduction to Python3
   a) Printing your biodata on the screen
   b) Printing all the primes less than a given number
   c) Finding all the factors of a number and show whether it is a *perfect* number, i.e., the sumof all its factors (excluding the number itself) is equal to the number itself
3. Defining and Using Functions
   a) Write a function to read data from a file and display it on the screen
   b) Define a boolean function *is palindrome*(<input>)
   c) Write a function *collatz(x)* which does the following: if x is odd, $x = 3x + 1$; if x is even, then $x = x/2$. Return the number of steps it takes for $x = 1$
   d) Write a function $N(m, s) = exp(-(x-m)^2/(2s^2))/sqrt(2\pi)s$ that computes the Normaldistribution

4. The package numpy
   a) Creating a matrix of given order *m x n* containing *random numbers* in the range 1 to 99999
   b) Write a program that adds, subtracts and multiplies two matrices. Provide an interfacesuch that, based on the prompt, the function (addition, subtraction, multiplication) should be performed
   c) Write a program to solve a system of *n* linear equations in *n* variables using matrix inverse

5.The package scipy and pyplot

   a) Finding if two sets of data have the same *mean* value
   b) Plotting data read from a file
   c) Fitting a function through a set a data points using *polyfit* function
   d) Plotting a histogram of a given data set

6. The strings package
   1. Read text from a file and print the number of lines, words and characters
   2. Read text from a file and return a list of all *n* letter words beginning with a vowel
   3. Finding a secret message hidden in a paragraph of text
   4. Plot a histogram of words according to their length from text read from a file

Cycle -2

7. Installing OS on Raspberry Pi
   1. Installation using PiImager
   2. Installation using image file
      - Downloading an Image
      - Writing the image to an SD card
        - using Linux
        - using Windows
      - Booting up

   Follow the instructions given in the URL
   https://www.raspberrypi.com/documentation/computers/getting-started.html

8. Accessing GPIO pins using Python
   a. Installing GPIO Zero library.
      First, update your repositories list:
      sudo apt update
      Then install the package for Python 3:
      sudo apt install python3-gpiozero
   b. Blinking an LED connected to one of the GPIO pin
   c. Adjusting the brightness of an LED
   d. Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) usingthe in-built PWM wavelength.

9. Collecting Sensor Data
   a) DHT Sensor interface
      - Connect the terminals of DHT GPIO pins of Raspberry Pi.
      - Import the DHT library using *import Adafruit_DHT*
      - Read sensor data and display it on screen.

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

B.TECH – DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Academic Year: 2022-2023 B.Tech I Year II Sem

(R22CSE1224) APPLIED PYTHON PROGRAMMING LABORATORY

LIST OF PROGRAMS

S.No	Name Of The Experiment
1	Downloading and Installing Python and Modules a) Python 3 on Linux Follow the instructions given in the URL https://docs.python-guide.org/starting/install3/linux/ b) Python 3 on Windows Follow the instructions given in the URL https://docs.python.org/3/using/windows.html(Please remember that Windows installation of Python is harder!) c) pip3 on Windows and Linux Install the Python package installer by following the instructions given in the URLhttps://www.activestate.com/resources/quick-reads/how-to-install-and-use-pip3/ d) Installing numpy and scipy You can install any python3 package using the command pip3 install <packagename> e) Installing jupyterlab Install from pip using the command pip install jupyterlab
2	Introduction to Python3 a) Printing your biodata on the screen b) Printing all the primes less than a given number Finding all the factors of a number and show whether it is a *perfect* number, i.e., the sumof all its factors (excluding the number itself) is equal to the number itself
3	Defining and Using Functions a) Write a function to read data from a file and display it on the screen b) Define a boolean function *is palindrome*(<input>) c) Write a function *collatz(x)* which does the following: if *x* is odd, $x = 3x + 1$; if *x* is even,then $x = x/2$. Return the number of steps it takes for $x = 1$ Write a function $N(m, s) = exp(-(x-m)^2/(2s^2))/sqrt(2\pi)s$ that computes the Normal distribution
4	The package numpy a) Creating a matrix of given order *m x n* containing *random numbers* in the range 1 to 99999 b) Write a program that adds, subtracts and multiplies two matrices. Provide an interfacesuch that, based on the prompt, the function (addition, subtraction,

	multiplication) should be performed Write a program to solve a system of *n* linear equations in *n* variables using matrix
5	The package scipy and pyplot a) Finding if two sets of data have the same *mean* value b) Plotting data read from a file c) Fitting a function through a set a data points using *polyfit* function d) Plotting a histogram of a given data set
6	The strings package a) Read text from a file and print the number of lines, words and characters b) Read text from a file and return a list of all *n* letter words beginning with a vowel c) Finding a secret message hidden in a paragraph of text d) Plot a histogram of words according to their length from text read from a file
7	Installing OS on Raspberry Pi 1. Installation using PiImager 2. Installation using image file ▪ Downloading an Image ▪ Writing the image to an SD card • using Linux • using Windows ▪ Booting up Follow the instructions given in the URL https://www.raspberrypi.com/documentation/computers/getting-started.html
8	Accessing GPIO pins using Python a. Installing GPIO Zero library. First, update your repositories list: *sudo apt update* Then install the package for Python 3: *sudo apt install python3-gpiozero* b. Blinking an LED connected to one of the GPIO pin c. Adjusting the brightness of an LED d. Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) usingthe in-built PWM wavelength.
9	Collecting Sensor Data a) DHT Sensor interface ▪ Connect the terminals of DHT GPIO pins of Raspberry Pi. ▪ Import the DHT library using *import Adafruit_DHT* Read sensor data and display it on screen

SRI INDU COLLEGE OF ENGG & TECHNOLOGYLESSON PLAN

(Regulation :R22)

Department of Electronics and Communication Engineering

preparedon:2.2.2022

Sub. Code & Title	**(R22CSE1127)** C PROGRAMMING FOR ENGINEERS LABORATORY		
Academic Year:2022-23		**Year/Sem./Section**	**I-I/ECE**
Faculty Name & Designation	**G.Anitha, Asst.Professor**		

S.No.	Topic (s)	Book Reference	Page (s)		Teaching Methodology	Proposed No. of hours	Actual Date of Handle d	CO/RBT
			From	To				
						33		
1	Downloading and Installing Python and Modules Python 3 on Linux Follow the instructions given in the URL https://docs.python-guide.org/starting/install3/linux/	Manual			BlackBoard & Demonstration On Projector	3		CO1/L3
2	Introduction to Python3 Printing your bio data on the screen	Manual			Demonstration On Projector			CO1/L3
3	Printing all the primes less than a given number	Manual			Demonstration On Projector	3		CO1/L3
4	Defining and Using Functions Write a function to read data from a file and display it on the screen	Manual			Demonstration On Projector			CO1/L3
5	Define a boolean function *is palindrome*(<input>)	Manual			Demonstration On Projector			CO1/L3
6	The package numpy Creating a matrix of given order *m x n* containing *random numbers* in the range 1 to 99999	Manual			Demonstration on Projector	3		CO2/L3
7	Write a program that adds, subtracts and multiplies two matrices. Provide an interfacesuch that, based on the prompt, the function (addition, subtraction, multiplication) should be performed	Manual			Demonstration on Projector	3		CO2/L3
8	Write a program to solve a system of *n* linear equations in *n* variables using matrix	Manual			Black Board	3		CO1/L3
9	The package scipy and pyplot Finding if two sets of data have the same *mean* value	Manual			Black Board	3		CO3/L3
10	Plotting data read from a file	Manual			Black Board &Demonstration onProjector	3		CO3/L3
11	Fitting a function through a set a data points using *polyfit* function	Manual			Black Board	3		CO3/L4

12	Plotting a histogram of a given data set	Manual			Black Board &Demonstration on Projector	3		CO3/L3
13	The strings package Read text from a file and print the number of lines, words and characters	Manual			Black Board &Demonstration on Projector	3		CO3/L3
14	Read text from a file and return a list of all *n* letter words beginning with a vowel	Manual			Black Board &Demonstration on Projector	3		CO3/L3
15	Finding a secret message hidden in a paragraph of text	Manual			Black Board &Demonstrati on on Projector	3		CO3/L3
16	Plot a histogram of words according to their length from text read from a file	Manual			Black Board &Demonstration on Projector	3		CO1/L4
17	Istalling OS on Raspberry Pi Installation using PiImager Installation using image file Downloading an image Writing the image to SD card Using Linux Using Windows Booting up using the following URL https://www.raspberrypi.com/documentation/computers/getting-started.html	Manual			Demonstration on Projector	3		
18	Accessing GPIO pins using Python Installing GPIO Zero library. First, update your repositories list: *sudo apt update* Then install the package for Python 3: *sudo apt install python3-gpiozero* Blinking an LED connected to one of the GPIO pin Adjusting the brightness of an LED Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness) usingthe in-built PWM wavelength	Manual			Demonstration on Projector	3		
19	Collecting Sensor Data DHT Sensor interface Connect the terminals of DHT GPIO pins of Raspberry Pi. Import the DHT library using *import Adafruit_DHT* Read sensor data and display it on screen	Manual			Demonstration on Projector	3		

SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

B.TECH – DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Academic Year: 2022-2023 B.Tech I Year II Sem

(R22CSE1224) APPLIED PYTHON PROGRAMMING LABORATORY

I B.TECH-II SEMESTER 2022-2023 TIME TABLE **Room No: 105(First Year Block)**

TIME DAY	09:40am To 10:30am	10:30am To 11:20pm	11:20am To 12:10pm		12:40pm To 1:45pm	1:45pm To 2:50pm	2:50pm To 4:00pm
	1	2	3		4	5	6
MON	ECE-A			L U N C H			
TUE							
WED							
THU							
FRI							
SAT					ECE-B		

LAB INCHARGE HOD PRINCIPAL

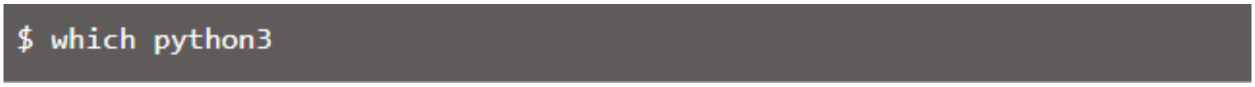
1. Downloading and Installing Python and Modules

   a) Python 3 on Linux

Follow the instructions given in the URL https://docs.python-guide.org/starting/install3/linux/

Install Python on Linux

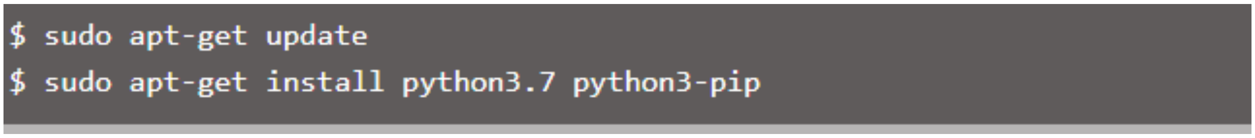
Most of Linux distributions come with Python already installed. However, the Python 2.x version is incorporated in many of them. To check if Python 3.x is available, run the following command in the Linux terminal:



```
$ which python3
```

If available, it will return the path to the Python3 executable as `/usr/local/bin/python3`.

To install Python on Ubuntu 18.04, Ubuntu 20.04 and above, execute the following commands:



```
$ sudo apt-get update
$ sudo apt-get install python3.7 python3-pip
```

After the installation, you can run Python 3.8 and pip3 commands.

For other Linux distributions use the corresponding package managers, such as YUM for Red Hat, aptitude for debian, DNF for Fedora, etc.

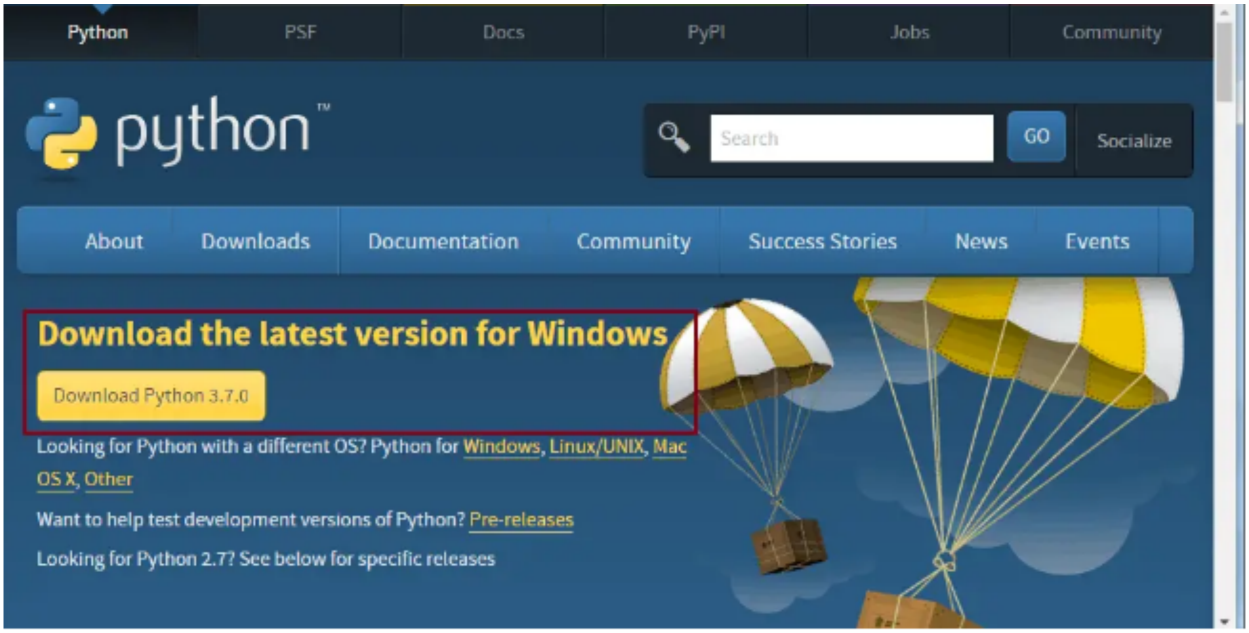
For installation on other platforms as well as installation from the source code, please refer to the official documentation on Python Source Releases page.

2. Python 3 on Windows Follow the instructions given in the URL https://docs.python.org/3/using/windows.html

Install Python on Windows

To install Python on a Windows platform, you need to download the installer. A web-based installer, executable installer and embeddable zip files are available to install Python on Windows. Visit https://www.python.org/downloads/windows and download the installer based on your local machine's hardware architecture.

The web-based installer needs an active internet connection. So, you can also download the standalone executable installer. Visit https://www.python.org/downloads and click on the **Download Python 3.7.0** button as shown below. (3.7.0 is the latest version as of this writing.)



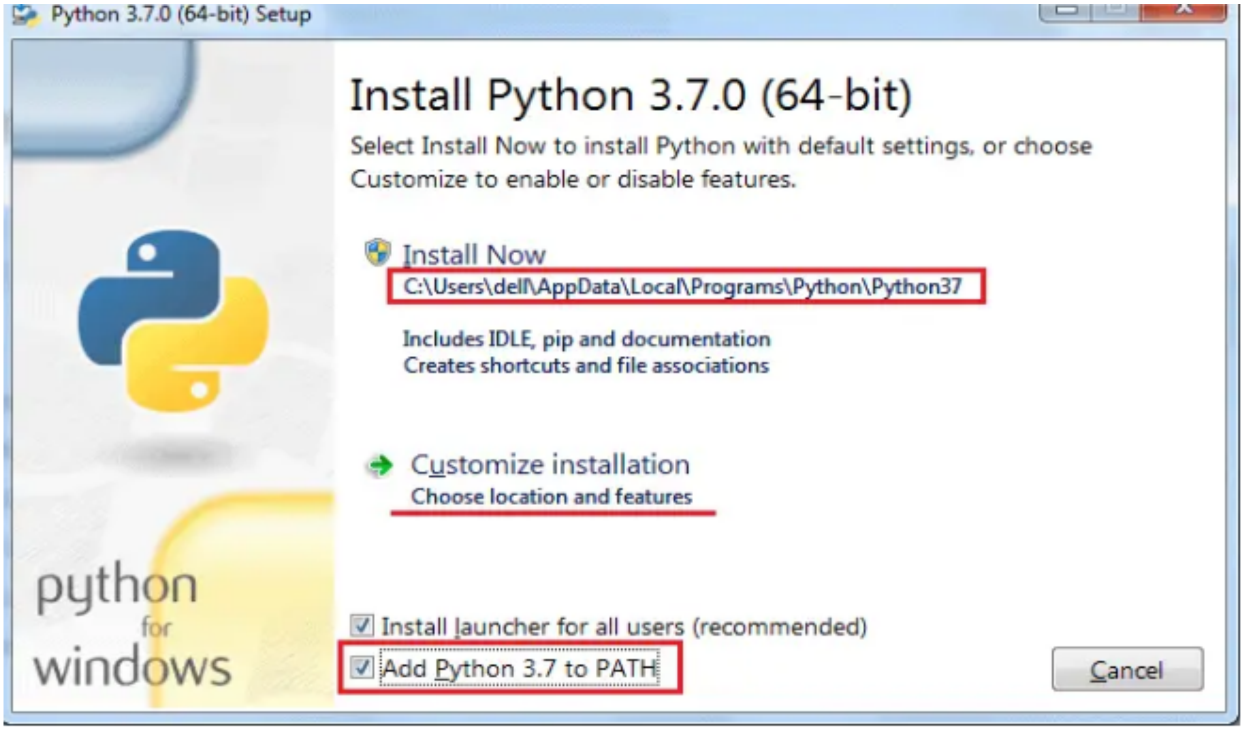
Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases

Looking for Python 2.7? See below for specific releases

This will download python-3.7.0.exe for 32 bit. For the 64 bit installer, go to https://www.python.org/downloads/windows and select the appropriate 64 bit installer, as shown below.



Python 3.7.0 (64-bit) Setup

Install Python 3.7.0 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\dell\AppData\Local\Programs\Python\Python37

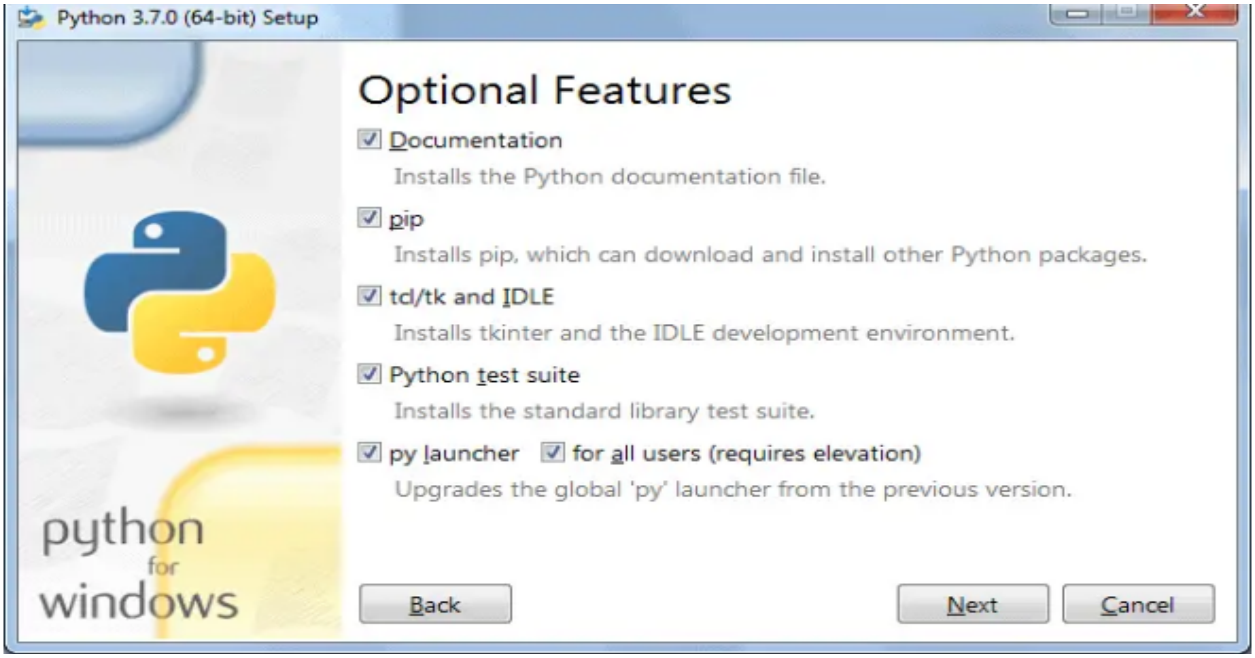
Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.7 to PATH

Installation is a simple wizard-based process. As you can see in the above figure, the default installation folder will be C:\ Users\ {UserName}\ AppData\ Local\Programs\ Python\ Python37 for Python 3.7.0 64 bit. Check the **Add Python 3.7 to PATH** checkbox, so that you can execute python scripts from any path. You may choose the installation folder or feature by clicking on **Customize installation**. This will go to the next step of optional features, as shown below.



Optional Features

☑ Documentation

Installs the Python documentation file.

☑ pip

Installs pip, which can download and install other Python packages.

☑ tcl/tk and IDLE

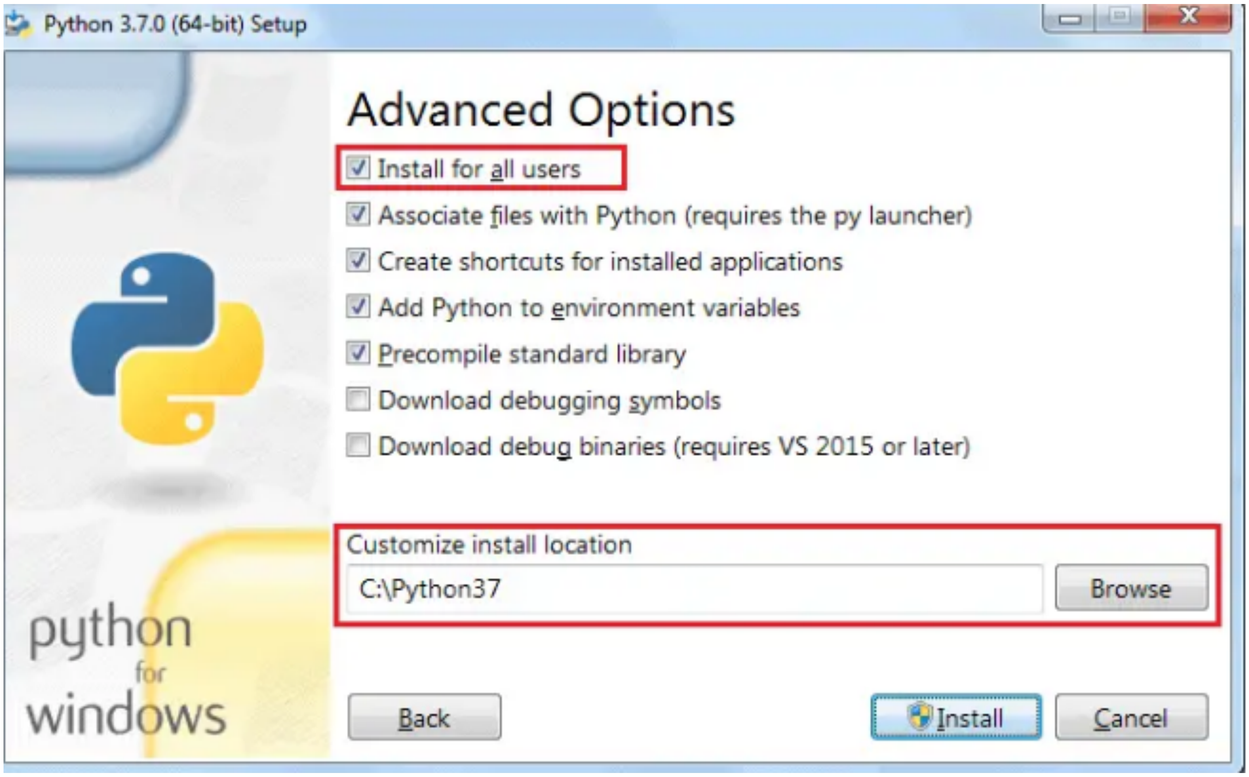
Installs tkinter and the IDLE development environment.

☑ Python test suite

Installs the standard library test suite.

☑ py launcher ☑ for all users (requires elevation)

Upgrades the global 'py' launcher from the previous version.



Advanced Options

☑ Install for all users

☑ Associate files with Python (requires the py launcher)

☑ Create shortcuts for installed applications

☑ Add Python to environment variables

☑ Precompile standard library

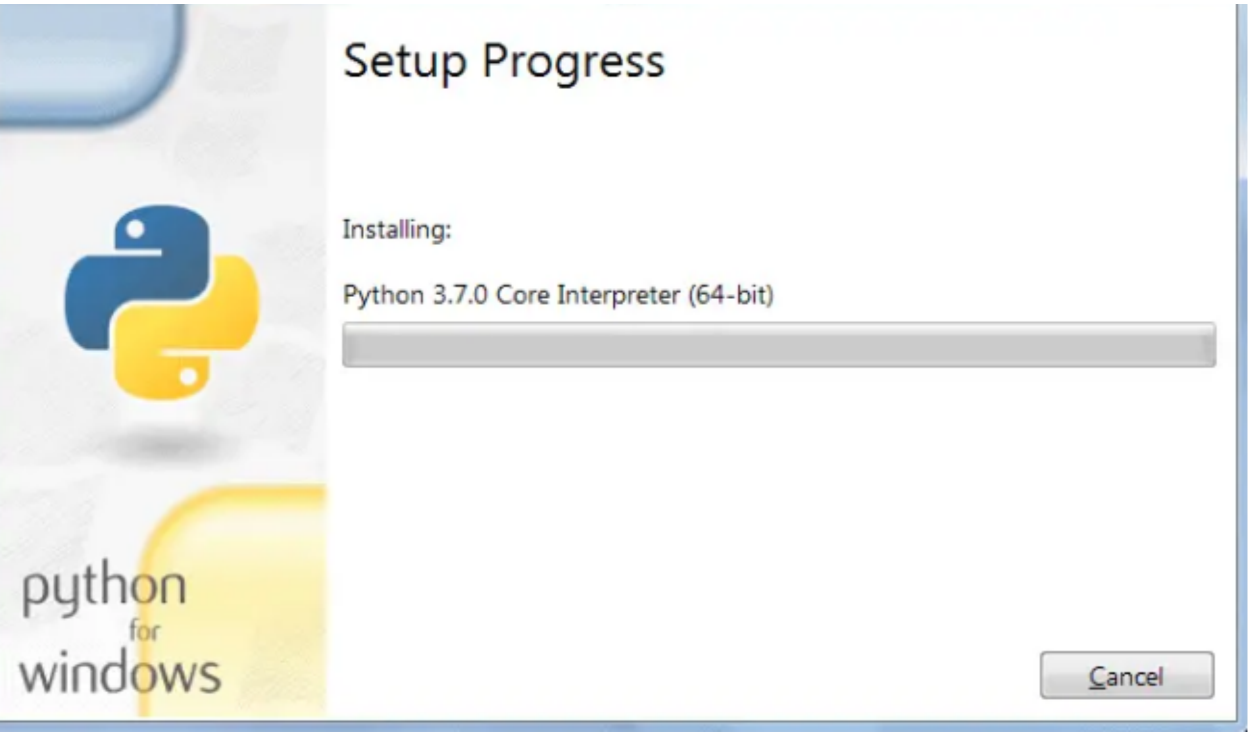
□ Download debugging symbols

□ Download debug binaries (requires VS 2015 or later)

Customize install location

C:\Python37

In Advanced Options, select the **Install for all users** option so that any user of your local machine can execute Python scripts. Also, choose the installation folder to make a shorter path for Python executable (something like C:\python37), keeping the rest of the choices to default and finally click on the Install button.

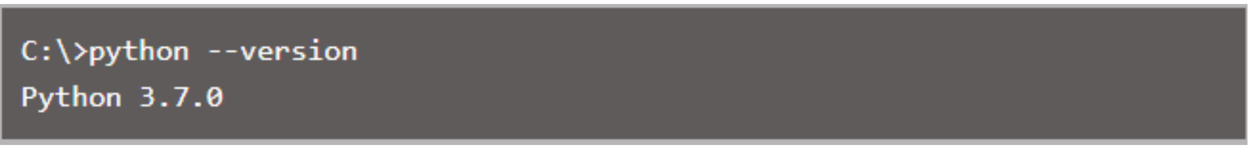


Setup Progress

Installing:

Python 3.7.0 Core Interpreter (64-bit)

After successful installation, you can check the Python installation by opening a command prompt and type `python --version` or `python -V` and press Enter. If Python installed successfully then it will display the installed version.



```
C:\>python --version
Python 3.7.0
```

3. pip3 on Windows and Linux

Install the Python package installer by following the instructions given in the URL https://www.activestate.com/resources/quick-reads/how-to-install-and-use-pip3/

Installing PIP On Windows

Follow the steps outlined below to install PIP on Windows.

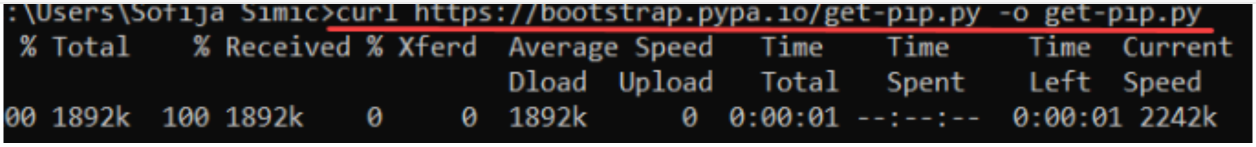
Step 1: Download PIP get-pip.py

Before installing PIP, download the get-pip.py file.

1. Launch a command prompt if it isn't already open. To do so, open the Windows search bar, type **cmd** and click on the icon.

2. Then, run the following command to download the **get-pip.py** file:

```
curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
```

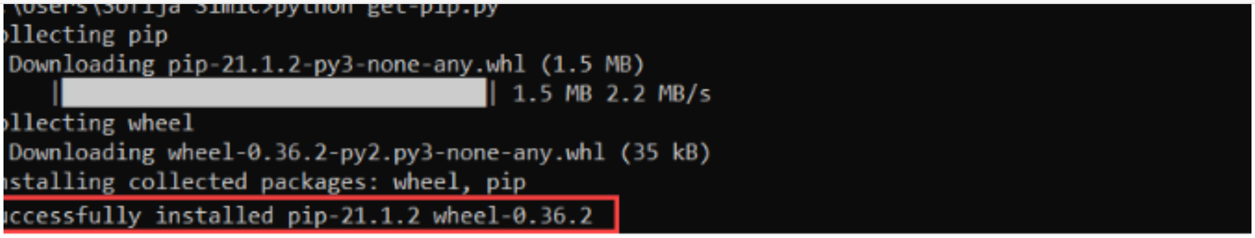


```
:\Users\Sofija Simic>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
00 1892k  100 1892k    0     0  1892k      0  0:00:01 --:--:--  0:00:01 2242k
```

Step 2: Installing PIP on Windows

To install PIP type in the following:

```
python get-pip.py
```



```
\users\Sofija Simic>python get-pip.py
ollecting pip
 Downloading pip-21.1.2-py3-none-any.whl (1.5 MB)
     |████████████████████████████████| 1.5 MB 2.2 MB/s
ollecting wheel
 Downloading wheel-0.36.2-py2.py3-none-any.whl (35 kB)
nstalling collected packages: wheel, pip
uccessfully installed pip-21.1.2 wheel-0.36.2
```

If the file isn't found, double-check the path to the folder where you saved the file. You can view the contents of your current directory using the following command:

```
dir
```

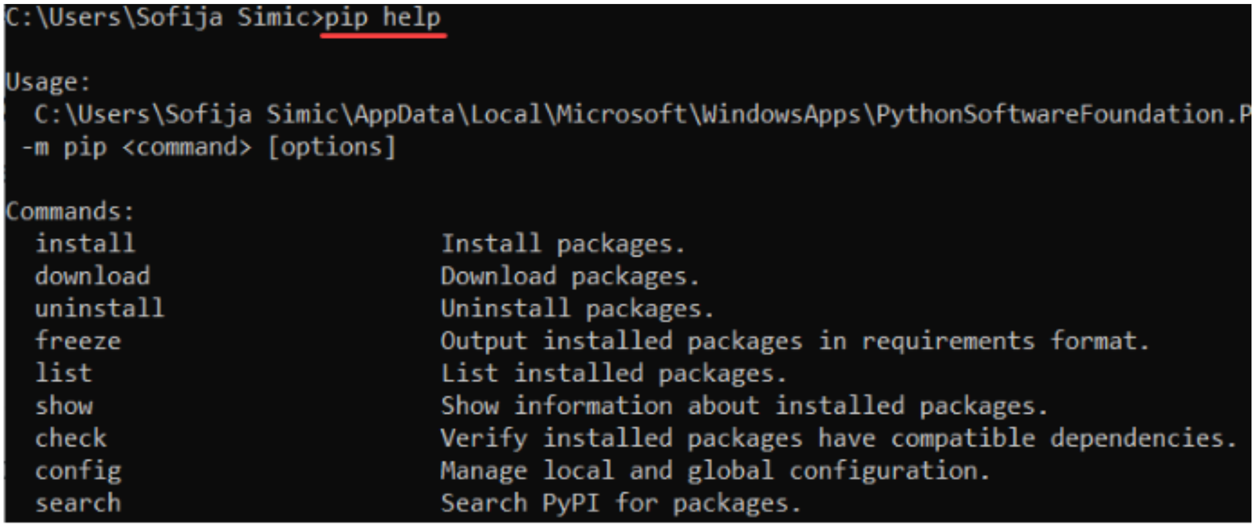
The `dir` command returns a full listing of the contents of a directory.

Step 3: Verify Installation

Once you’ve installed PIP, you can test whether the installation has been successful by typing the following:

```
pip help
```

If PIP has been installed, the program runs, and you should see the location of the software package and a list of commands you can use with `pip`.

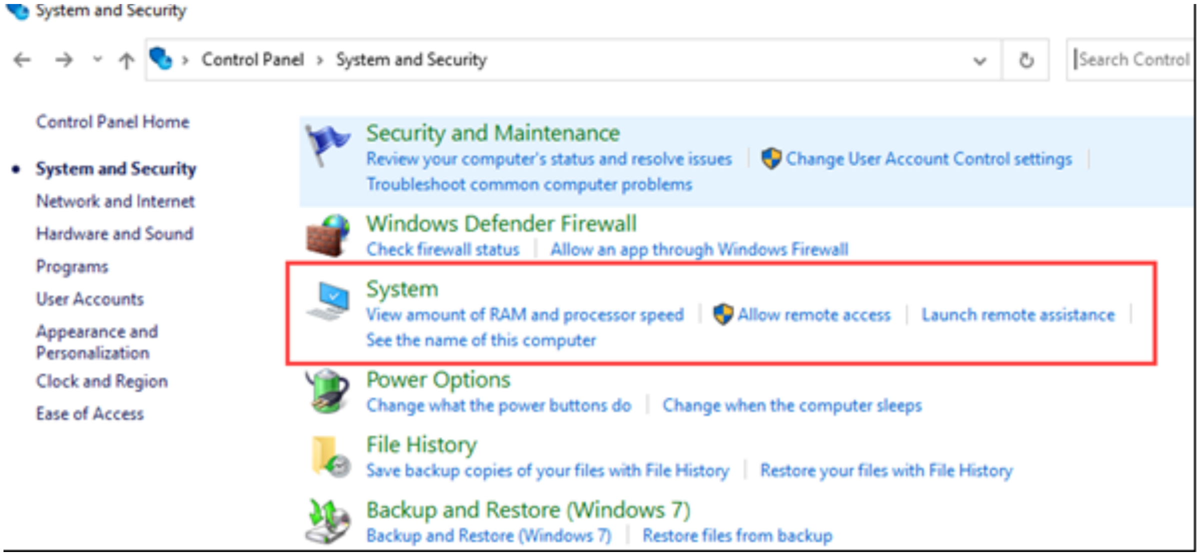


```
C:\Users\Sofija Simic>pip help

Usage:
  C:\Users\Sofija Simic\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.P
 -m pip <command> [options]

Commands:
  install                     Install packages.
  download                    Download packages.
  uninstall                   Uninstall packages.
  freeze                      Output installed packages in requirements format.
  list                        List installed packages.
  show                        Show information about installed packages.
  check                       Verify installed packages have compatible dependencies.
  config                      Manage local and global configuration.
  search                      Search PyPI for packages.
```

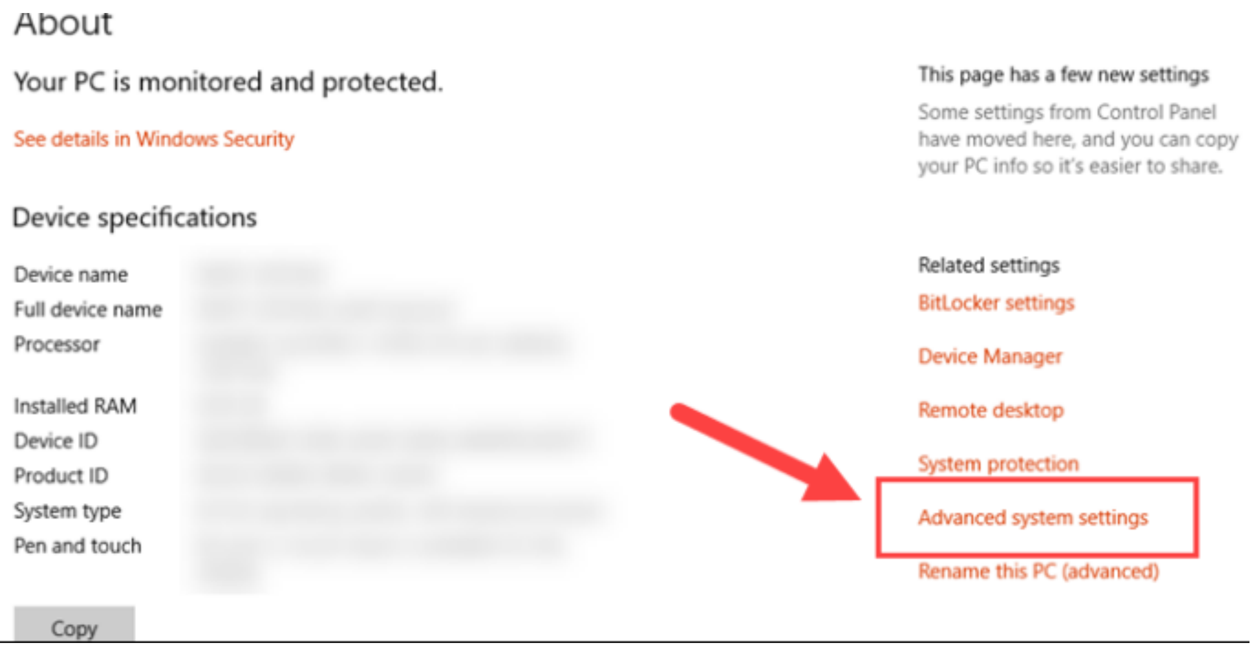
Step 4: Add Pip to Windows Environment Variables

To run PIP from any location, you need to add it to Windows environment variables to avoid getting the "not on PATH" error. To do so, follow the steps outlined below:

1. Open the **System and Security** window by searching for it in the **Control Plane**.
2. Navigate to **System** settings


System and Security
Control Panel > System and Security
Search Control
Control Panel Home
System and Security
Network and Internet
Hardware and Sound
Programs
User Accounts
Appearance and Personalization
Clock and Region
Ease of Access
Security and Maintenance
Review your computer's status and resolve issues
Change User Account Control settings
Troubleshoot common computer problems
Windows Defender Firewall
Check firewall status
Allow an app through Windows Firewall
System
View amount of RAM and processor speed
Allow remote access
Launch remote assistance
See the name of this computer
Power Options
Change what the power buttons do
Change when the computer sleeps
File History
Save backup copies of your files with File History
Restore your files with File History
Backup and Restore (Windows 7)
Backup and Restore (Windows 7)
Restore files from backup

3. Then, select **Advanced system settings**.



About

Your PC is monitored and protected.

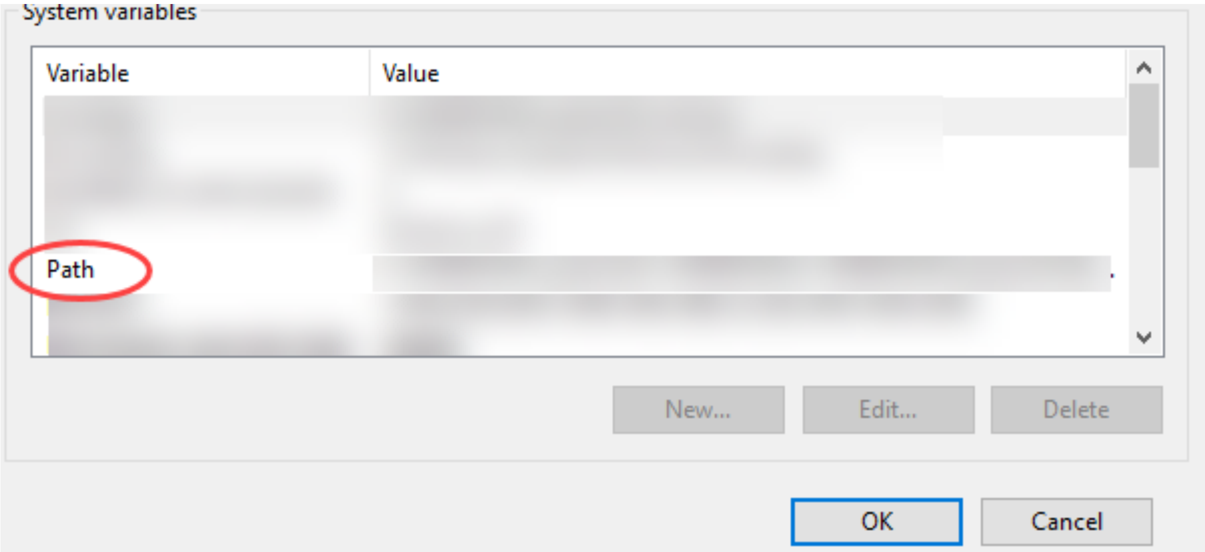
See details in Windows Security

This page has a few new settings

Some settings from Control Panel have moved here, and you can copy your PC info so it's easier to share.

Device specifications

4. Open the **Environment Variables** and double-click on the **Path** variable in the **System Variables**.



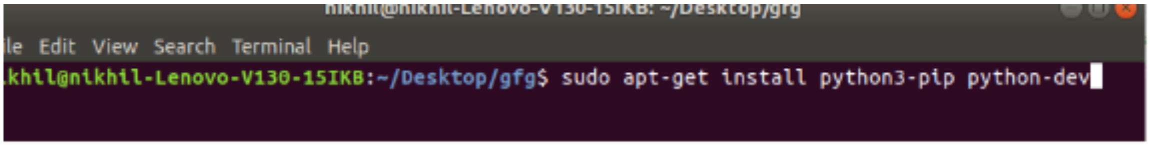
5. Next, select **New** and add the directory where you installed PIP.
6. Click **OK** to save the changes.

Step 5: Configuration

In Windows, the PIP configuration file is `%HOME%\pip\pip.ini.`

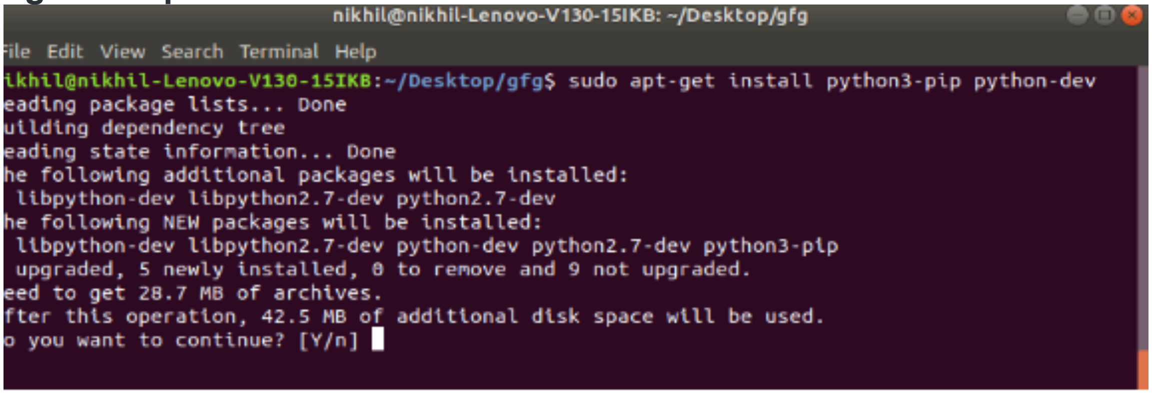
There is also a legacy per-user configuration file. The file is located at `%APPDATA%\pip\pip.ini`.

You can set a custom path location for this config file using the environment variable `PIP_CONFIG_FILE`.


nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
ile Edit View Search Terminal Help

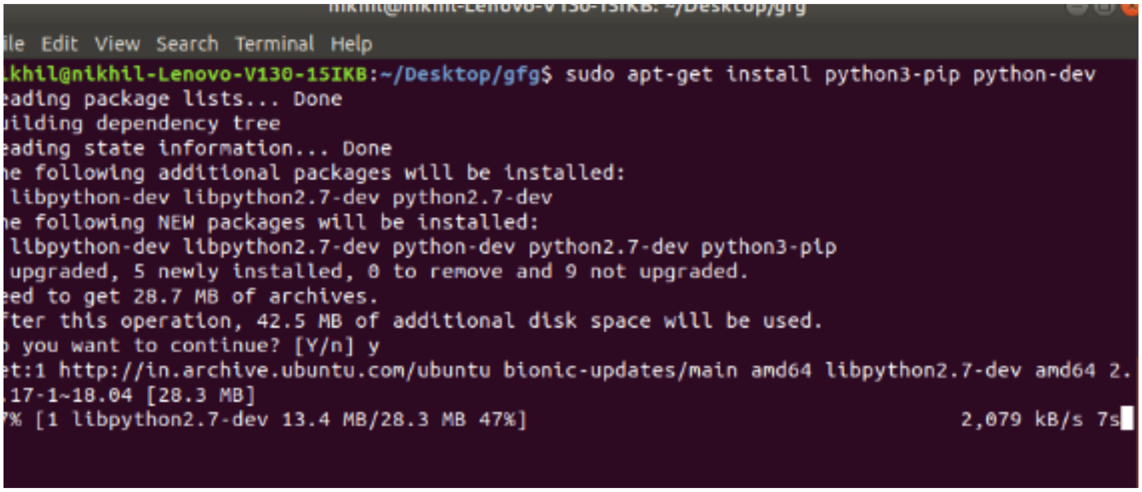
```
khil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$ sudo apt-get install python3-pip python-dev
```

Providing Disk Space:


nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
File Edit View Search Terminal Help

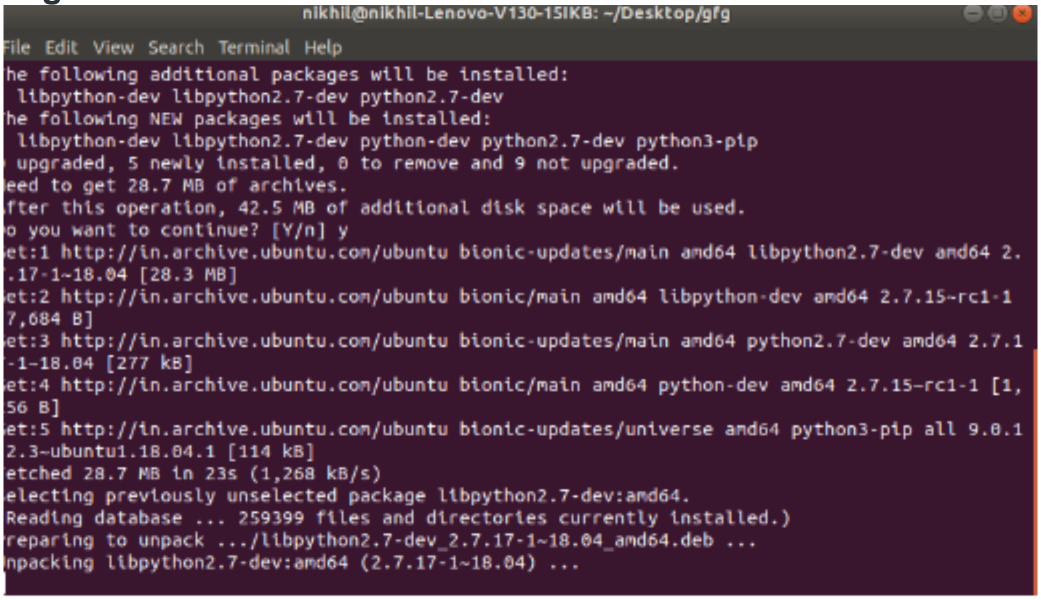
```
ikhil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$ sudo apt-get install python3-pip python-dev
eading package lists... Done
uilding dependency tree
eading state information... Done
he following additional packages will be installed:
 libpython-dev libpython2.7-dev python2.7-dev
he following NEW packages will be installed:
 libpython-dev libpython2.7-dev python-dev python2.7-dev python3-pip
 upgraded, 5 newly installed, 0 to remove and 9 not upgraded.
eed to get 28.7 MB of archives.
fter this operation, 42.5 MB of additional disk space will be used.
o you want to continue? [Y/n]
```

Downloading Libraries:


nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
ile Edit View Search Terminal Help

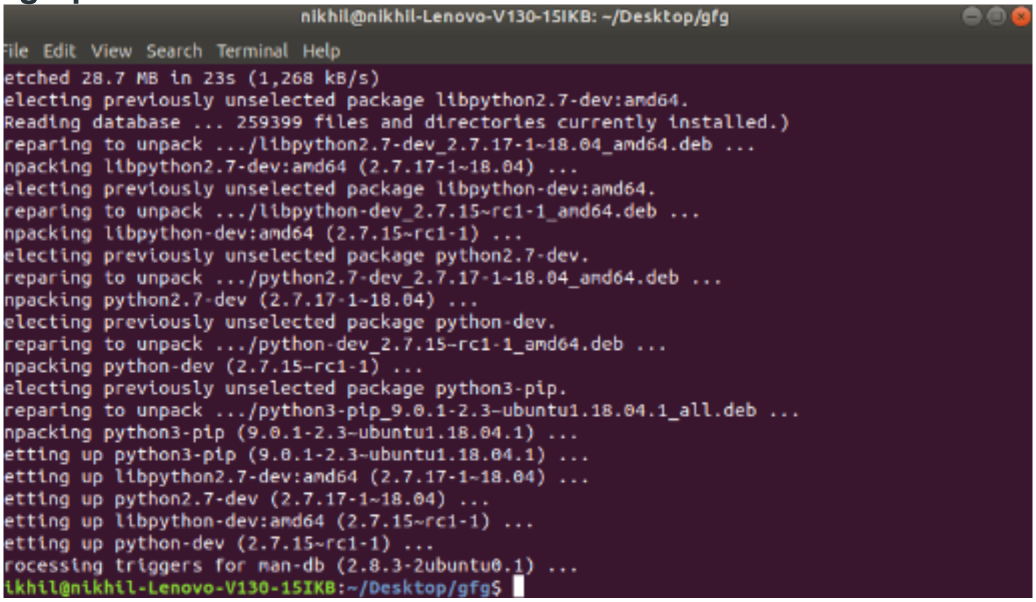
```
khil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$ sudo apt-get install python3-pip python-dev
eading package lists... Done
uilding dependency tree
eading state information... Done
he following additional packages will be installed:
 libpython-dev libpython2.7-dev python2.7-dev
he following NEW packages will be installed:
 libpython-dev libpython2.7-dev python-dev python2.7-dev python3-pip
 upgraded, 5 newly installed, 0 to remove and 9 not upgraded.
eed to get 28.7 MB of archives.
fter this operation, 42.5 MB of additional disk space will be used.
 you want to continue? [Y/n] y
et:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython2.7-dev amd64 2.
.17-1~18.04 [28.3 MB]
7% [1 libpython2.7-dev 13.4 MB/28.3 MB 47%]                          2,079 kB/s 7s
```

Unpacking File bundles:


nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
File Edit View Search Terminal Help

```
he following additional packages will be installed:
 libpython-dev libpython2.7-dev python2.7-dev
he following NEW packages will be installed:
 libpython-dev libpython2.7-dev python-dev python2.7-dev python3-pip
 upgraded, 5 newly installed, 0 to remove and 9 not upgraded.
eed to get 28.7 MB of archives.
fter this operation, 42.5 MB of additional disk space will be used.
o you want to continue? [Y/n] y
et:1 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libpython2.7-dev amd64 2.
.17-1~18.04 [28.3 MB]
et:2 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 libpython-dev amd64 2.7.15~rc1-1
7,684 B]
et:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 python2.7-dev amd64 2.7.1
-1~18.04 [277 kB]
et:4 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 python-dev amd64 2.7.15~rc1-1 [1,
56 B]
et:5 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 python3-pip all 9.0.1
2.3~ubuntu1.18.04.1 [114 kB]
etched 28.7 MB in 23s (1,268 kB/s)
electing previously unselected package libpython2.7-dev:amd64.
Reading database ... 259399 files and directories currently installed.)
reparing to unpack .../libpython2.7-dev_2.7.17-1~18.04_amd64.deb ...
npacking libpython2.7-dev:amd64 (2.7.17-1~18.04) ...
```

Finishing up the Installation

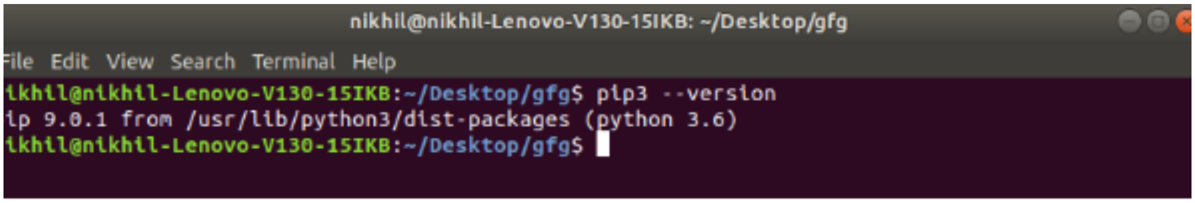

nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
File Edit View Search Terminal Help

```
etched 28.7 MB in 23s (1,268 kB/s)
electing previously unselected package libpython2.7-dev:amd64.
Reading database ... 259399 files and directories currently installed.)
reparing to unpack .../libpython2.7-dev_2.7.17-1~18.04_amd64.deb ...
npacking libpython2.7-dev:amd64 (2.7.17-1~18.04) ...
electing previously unselected package libpython-dev:amd64.
reparing to unpack .../libpython-dev_2.7.15~rc1-1_amd64.deb ...
npacking libpython-dev:amd64 (2.7.15~rc1-1) ...
electing previously unselected package python2.7-dev.
reparing to unpack .../python2.7-dev_2.7.17-1~18.04_amd64.deb ...
npacking python2.7-dev (2.7.17-1~18.04) ...
electing previously unselected package python-dev.
reparing to unpack .../python-dev_2.7.15~rc1-1_amd64.deb ...
npacking python-dev (2.7.15~rc1-1) ...
electing previously unselected package python3-pip.
reparing to unpack .../python3-pip_9.0.1-2.3~ubuntu1.18.04.1_all.deb ...
npacking python3-pip (9.0.1-2.3~ubuntu1.18.04.1) ...
etting up python3-pip (9.0.1-2.3~ubuntu1.18.04.1) ...
etting up libpython2.7-dev:amd64 (2.7.17-1~18.04) ...
etting up python2.7-dev (2.7.17-1~18.04) ...
etting up libpython-dev:amd64 (2.7.15~rc1-1) ...
etting up python-dev (2.7.15~rc1-1) ...
rocessing triggers for man-db (2.8.3-2ubuntu0.1) ...
ikhil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$
```

Verification of the Installation process:

One can easily verify if the pip has been installed correctly by performing a version check on the same. Just go to the command-line and execute the following command:

```
pip3 --version
```


nikhil@nikhil-Lenovo-V130-15IKB: ~/Desktop/gfg
ile Edit View Search Terminal Help

```
ikhil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$ pip3 --version
ip 9.0.1 from /usr/lib/python3/dist-packages (python 3.6)
ikhil@nikhil-Lenovo-V130-15IKB:~/Desktop/gfg$ 
```

JupyterLab

Install JupyterLab with `pip`:

```
pip install jupyterlab
```

Printing your biodata on the screen

```
print("name: abc")
print("roll number: 22D41A0 444")
print("branch : ECE")
print("section: A")
print("emailid: abc@gmail.com")
print("phone number: 34567890")
```

1. Printing all the primes less than a given number

```
lower = int(input("enter lower value"))
upper = int(input("enter upper value"))
print("Prime numbers between", lower, "and", upper, "are:")
  for num in range(lower, upper + 1):
      if num > 1:
       for i in range(2, num):
           if (num % i) == 0:
               break
       else:
                print(num)
```

2. Finding all the factors of a number and show whether it is a *perfect* number, i.e., the sum of all its factors (excluding the number itself) is equal to the number itself

```
N = int(input("enter n"))

i = 1
sum=0
while(i<N):
  if(N%i==0):
    sum=sum+i
  i=i+1

if(sum == N):
  print(N, "is a perfect number")
else:
  print(N, "is not a perfect number")
```

3. Defining and Using Functions

Write a function to read data from a file and display it on the screen

```
file = open("myfile.txt","w")
L = ["This is Lagos \n","This is Python \n","This is Fcc \n"]

file.write("Hello There \n")
file.writelines(L)
file.close()
f = open("myfile.txt", "r")

print(f.read())
```

4. Define a boolean function *is palindrome*(<input>)

```
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Enter string:"))
if(is_palindrome(a)==True):
    print("String is a palindrome!")
else:
    print("String isn't a palindrome!")
```

5. Write a function *collatz(x)* which does the following: if *x* is odd, *x = 3x + 1*; if *x* is even, then *x = x/2*. Return the number of steps it takes for *x* = 1

```
     def collatz_sequence(x):
    num_seq = [x]
    if x < 1:
       return []
    while x > 1:
       if x % 2 == 0:
         x = x / 2
       else:
         x = 3 * x + 1
    # Added line
       num_seq.append(x)
    return num_seq

print(collatz_sequence(12))

print(collatz_sequence(19))
```

6. Creating a matrix of given order *m x n* containing *random numbers* in the range 1 to 99999

```
import numpy as np

# random is a function, doing random sampling in numpy.
array = np.random.randint(10, size=(999))

# the array will be having 20 elements.
print(array)
```

7. Write a program that adds, subtracts and multiplies two matrices. Provide an interface such that, based on the prompt, the function (addition, subtraction, multiplication)should be performed

```
import numpy as np

print("1.addition 2 subtraction 3 multiplication")
choice=int(input("enter your choice"))
      # creating first matrixA =
np.array([[1, 2], [3, 4]])

# creating second matrix
B = np.array([[4, 5], [6, 7]])
if choice==1:
  print("Printing elements of first matrix")print(A)
  print("Printing elements of second matrix")print(B)
  print("Addition of two matrix")
  print(np.add(A, B))
elif choice==2:

# subtracting two matrix
  print("Printing elements of first matrix")print(A)
  print("Printing elements of second matrix")print(B)
  print("Subtraction of two matrix")
  print(np.subtract(A, B))
else:
   print("Printing elements of first matrix")print(A)
   print("Printing elements of second matrix")print(B)
   print(" multiplication of two matrices is")res =
   np.dot(A,B)
   print(res)
```

8.Write a program to solve a system of *n* linear equations in *n* variables using matrix Inverse

```
    import numpy as np
    A = np.array([[4, 3, 2], [-2, 2, 3], [3, -5, 2]])
    B = np.array([25, -10, -4]) X =
    np.linalg.inv(A).dot(B)
print(X)
```

9.Finding if two sets of data have the same *mean* value

```
from scipy import mean

arr1 = [[1, 3, 27],
    [3, 4, 6],
    [7, 6, 3],
    [3, 6, 8]]
```

```
print("Arithmetic Mean is :", mean(arr1))arr2 = [[1,
3, 27],
      [3, 0, 6],
      [7, 6, 4],
      [8, 6, 8]]
print("Arithmetic Mean is :", mean(arr2))if
mean(arr1)==mean(arr2):
   print("menas are equal")else:
print("means are not equal")
```

10. Plotting data read from a file

```
import matplotlib.pyplot as plt

names = []marks = []

f = open('sample.txt','r')for row in f:
   row = row.split(' ')
   names.append(row[0])
   marks.append(int(row[1]))

plt.bar(names, marks, color = 'g', label = 'File Data')

plt.xlabel('Student Names', fontsize = 12)
plt.ylabel('Marks', fontsize = 12)

plt.title('Students Marks', fontsize = 20)
plt.legend()
plt.show()
```

11.Fitting a function through a set a data points using *polyfit* function

```
   import numpy as np
   import matplotlib.pyplot as mp
   np.random.seed(12)
   x = np.linspace( 0, 1, 25 )
   y = np.cos(x) + 0.3*np.random.rand(25)p =
   np.poly1d( np.polyfit(x, y, 4) )
     t = np.linspace(0, 1, 250) mp.plot(x, y, 'o', t,
p(t), '-')mp.show()
```

12. Plotting a histogram of a given data set

```
from matplotlib import pyplot as pltimport
numpy as np


# Creating dataset
a = np.array([22, 87, 5, 43, 56,
          73, 55, 54, 11,
          20, 51, 5, 79, 31,
          27])

# Creating histogram
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])

# Show plotplt.show()
```

13. Read text from a file and print the number of lines, words and characters

```
file = open('text.txt')lines = 0

words = 0

symbols = 0 for line in file:

   lines += 1

   words += len(line.split()) symbols +=

   len(line.strip('\n'))

  print("Lines:", lines) print("Words:", words)
  print("Symbols:", symbols)
```

14. Read text from a file and return a list of all n letter words beginning with a vowel

```
test_list = ["all", "love", "and", "get", "educated", "by", "gfg"]#

printing original list

print("The original list is : " + str(test_list))res = []

vow = "aeiou"
```

```
  for sub in test_list:flag = False

     # checking for begin charfor ele in

     vow:

        if sub.startswith(ele):flag = True

           breakif flag:

        res.append(sub)# printing

  result

  print("The extracted words : " + str(res))
```

15. Finding a secret message hidden in a paragraph of text

```
  # txt_hide.py

  #txt_show.py
from PIL import Image
import numpy as np
import tkinter.filedialog as tkf
# Get the image file name
root = tkf.Tk()
root.withdraw()
img_file = tkf.askopenfilename(
                      parent=root, title="Select a PNG file", filetypes=(("PNG files", "*.png"),)
)
# Load and reshape the image file
img = np.array(Image.open(img_file))
shp = img.shape
the_bytes = img.reshape(np.prod(shp))
# Extract the message
char_codes, n, c, m = [], 0, 0, 1
for byt in the_bytes:
                      if byt & 1:
                         c += m
                      n += 1
                      m <<= 1
                      if n % 8 == 0:
                         if c == 4:
                            break
```

```
                    char_codes.append(c)
                    c, m = 0, 1
s = "".join(map(chr, char_codes))
# Output the results
print(s)
```

16. Plot a histogram of words according to their length from text read from a file

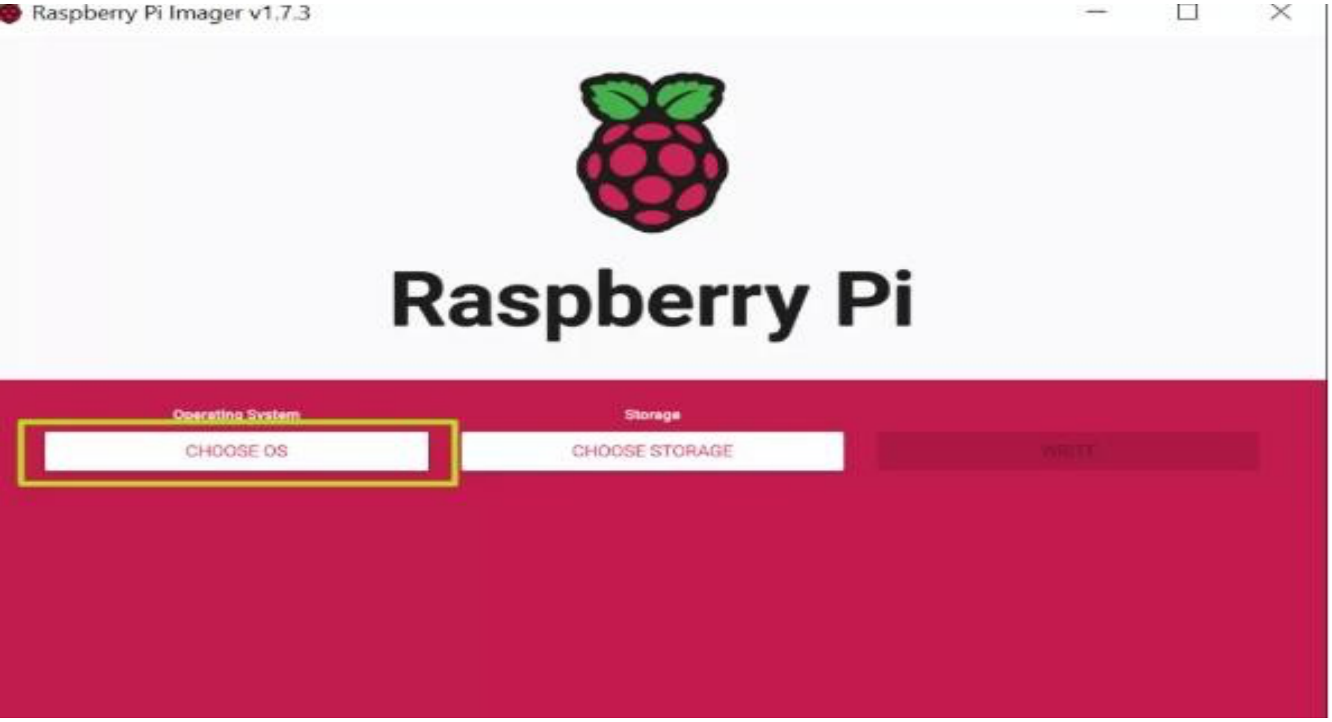
```
import sys import time
from string import punctuation as punc
from string import maketrans as transimport
operator
lengthDict = {}

def insert_Dict(word):
  lengthDict[word] = lengthDict.get(word, 0) + 1

def timing(f):
  def wrap(*args): time1 =
    time.time()ret = f(*args)
    time2 = time.time()
    print '%s function took %0.3f ms' % (f.func_name, (time2-time1)*1000.0)return
    ret
  return wrap

def roundup(x):
  return x if x % 100 == 0 else x + 100 - x % 100


@timing def main():
  fileName = sys.argv[1]

  openFile = open(str(sys.argv[1]))readString

  = openFile.read()

  out = readString.translate(trans("",""), punc)

  for word in readString.split():
    if  word  in  punc: #print(word)
      insert_Dict(0)

  for word in out.split(): #print(word)
    insert_Dict(len(word))

  maxDomain = max(lengthDict.keys()) maxRange =
    max(lengthDict.values()) roundedMaxRange =
    roundup(maxRange)

    for x in range(roundedMaxRange,0, -20):newStr =
        ""
        if x % 100 == 0:
            newStr = newStr + str(x) + " -|"elif x % 20
        == 0:
            newStr = newStr + " |"else:
            newStr = newStr + " "


        for domain in range(maxDomain + 1):
            if lengthDict.get(domain) >= x and lengthDict.has_key(domain) :newStr =
                    newStr + ('***')
            else:
                newStr = newStr + ' 'print(newStr)

    print('   -|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-')print '
               0',
    for x in xrange(1, 16):print "",
        print x,print

if__name_== '_main__':
    print('This program is being run by itself')main()
else:
    print('I am being imported from another module')
```

Cycle 2

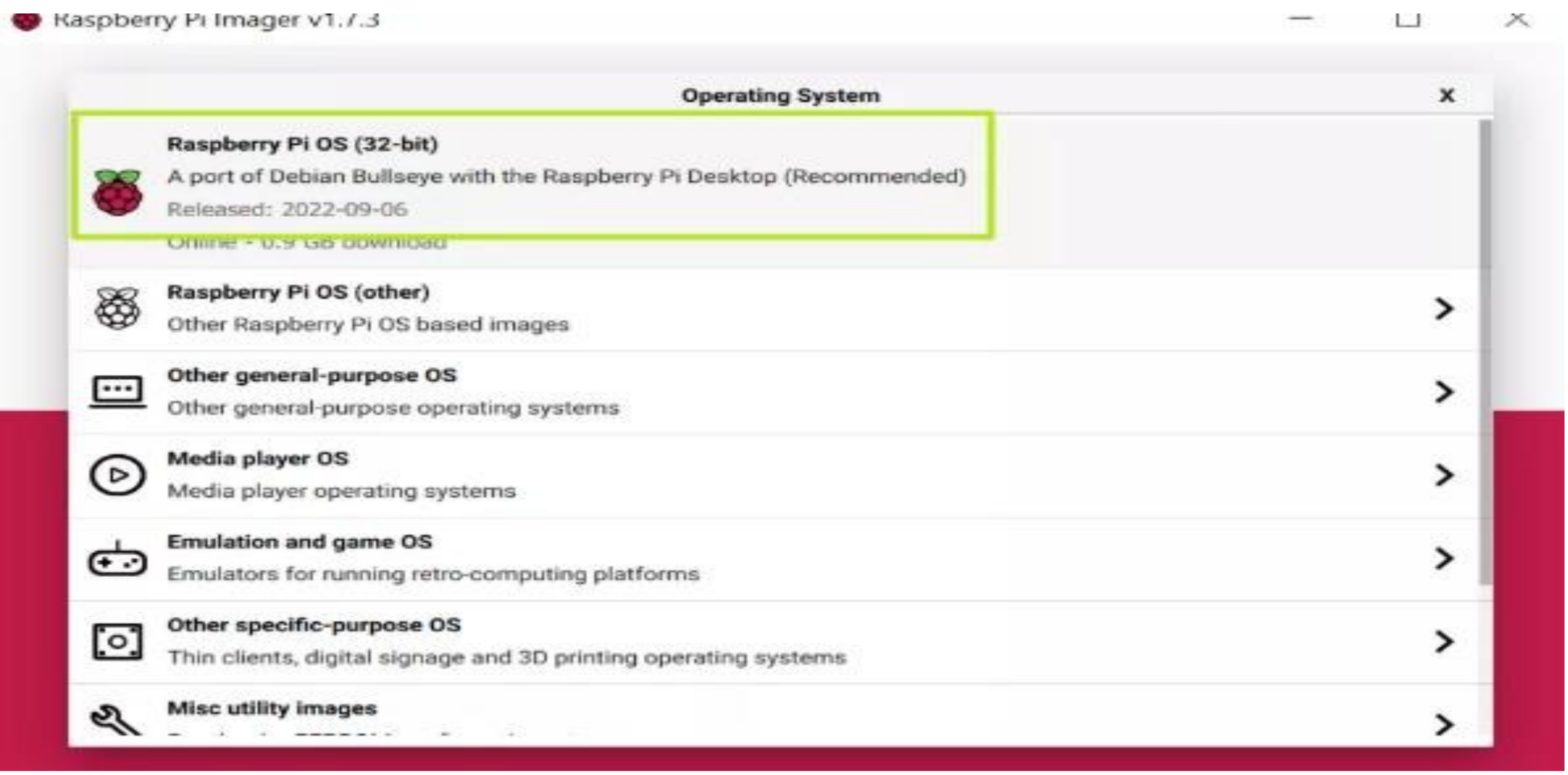
Downloading and Installing Raspberry Pi OS

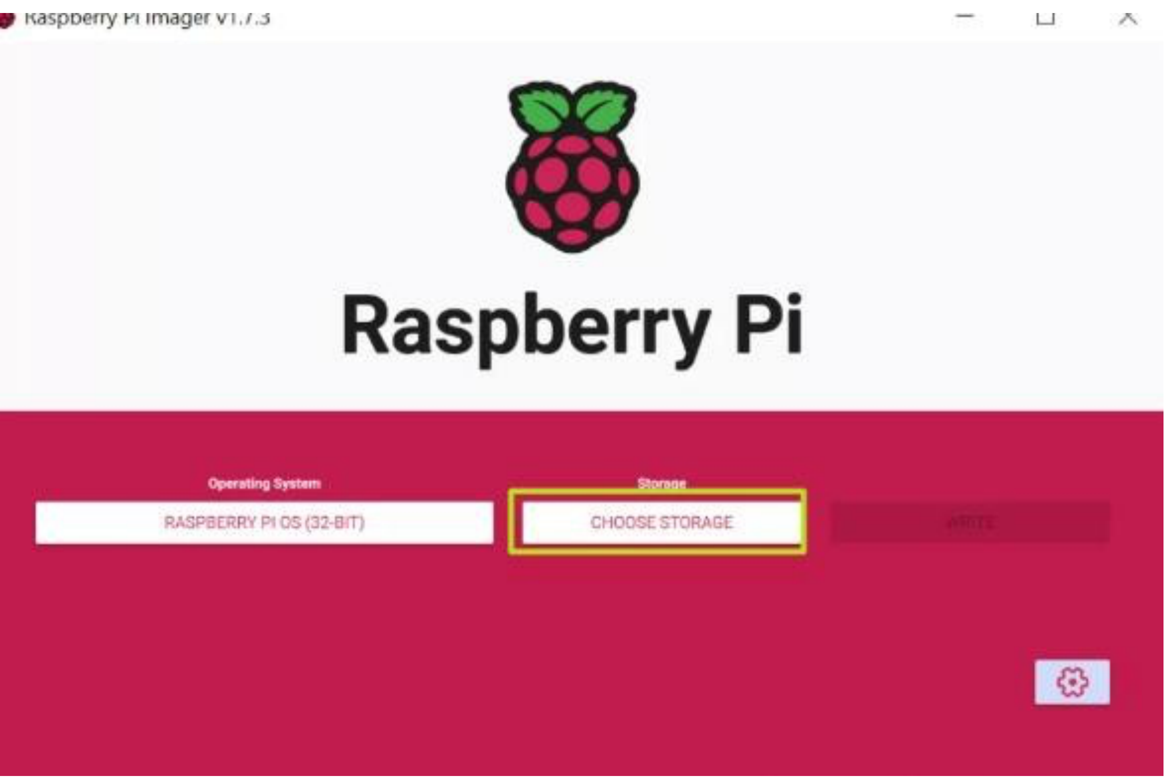
Once you have all the components you need, use the following steps to create the boot disk you will need to set up your Raspberry Pi. These steps should work on a using a Windows, Mac or Linux-based PC (we tried this on Windows, but it should be the same on all three).

1. **Insert a microSD card / reader** into your computer.
2. **Download and install the official Raspberry Pi Imager.** Available for Windows, macOS orLinux, this app will both download and install the latest Raspberry Pi OS. There are other ways to do this, namely by downloading a Raspberry Pi OS image file and then using a third-party appto "burn it," but the Imager makes it easier.
3. **Click Choose OS.**

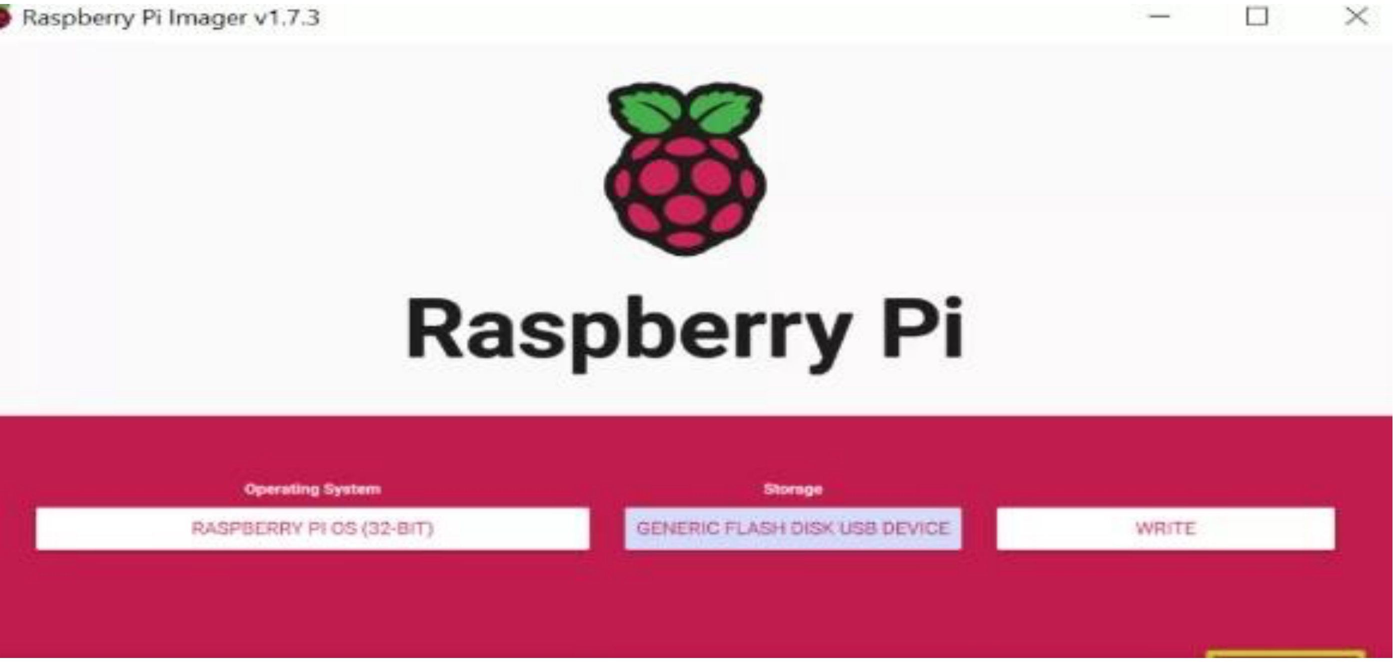


Select Raspberry Pi OS (32-bit) from the OS menu (there are other choices, but for most uses,32-bit is the best).


Raspberry Pi Imager v1.7.3
Operating System
Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-09-06
Online - 0.9 GB download
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
Other general-purpose OS
Other general-purpose operating systems
Media player OS
Media player operating systems
Emulation and game OS
Emulators for running retro-computing platforms
Other specific-purpose OS
Thin clients, digital signage and 3D printing operating systems
Misc utility images



1. **Click the settings button** or hit CTRL + SHIFT + X to enter settings.



Fill in settings fields as follows and then **hit Save**. All of these fields are technically optional,but highly recommended so that can get your Raspberry Pi set up and online as soon as you bootit. If you don't set a username and password here, you'll have to go through a setup wizard that asks you to create them on first boot.

- **Set hostname**: the name of your Pi. It could be "raspberrypi" or anything you like.
- **Enable SSH**: Allow SSH connections to the Pi. Recommended.
- **Use password authentication / public key:** method of logging in via SSH
- **Set username and password:** Pick the username and password you'll use for the Pi
- **Configure wireless LAN: s**et the SSID and password of Wi-FI network
- **Wireless LAN country:** If you're setting up Wi-Fi, you must choose this.
- **Set locale settings:** Configure keyboard layout and timezone (probably chosen correctly bydefault)

1. Accessing GPIO pins using Python

Installing GPIO Zero library

First, update your repositories list:

sudo apt update

Then install the package for Python 3:

sudo apt install python3-gpiozero

Installing GPIO zero

GPIO Zero is installed by default in the Raspberry Pi OS desktop image, and the Raspberry Pi Desktop image for PC/Mac, both available from raspberrypi.org. Follow these guides to installing on Raspberry Pi OS Lite and other operating systems, including for PCs using the remote GPIO feature.

1.1. Raspberry Pi

GPIO Zero is packaged in the apt repositories of Raspberry Pi OS, Debian and Ubuntu. It is also available on PyPI.

1.1.1. apt

First, update your repositories list:

```
pi@raspberrypi:~$ sudo apt update
```

Then install the package for Python 3:

```
pi@raspberrypi:~$ sudo apt install python3-gpiozero
```

or Python 2:

```
pi@raspberrypi:~$ sudo apt install python-gpiozero
```

2.Blinking an LED connected to one of the GPIO pin

Step 1 - Connections

Here we will be using GPIO 21 or pin no. 40 to make the LED blink. Ground (0V) at pin 39 will be used to provide ground with a resistance (270,330,1k ohm) in series to LED. Make connection by referring the Diagram below -

Step 2 - Writing Python code on Python 3(IDLE)

- Open Python 3 (IDLE) in your Raspberry Pi
- Now Create a new file. And add the code import RPi.GPIO as GPIO # Importing GPIO library

```
import RPi.GPIO as GPIO

import time  //Imported time library

GPIO.setmode(GPIO.Board)
GPIO.setup(11, GPIO.OUT)

while:

GPIO.output(11, True)

time.sleep(1)

GPIO.output(11, False)

time.sleep(1)
```

3. Adjusting the brightness of an LED

- Initialize and Start the PWM
- pwm = GPIO.PWM(Pin Number, Frequency), this function will configure the Raspberry Pi pin as PWM pin with the entered frequency (Frequency value should be greater than Persistence Of Vision value of Human Eyes, otherwise you will see flickering).
- pwm.Start(Duty Cycle), this function will start the PWM pulse on selected with the entered duty cycle value, valid values are from 0 to 100 where 0 means completely off and 100 means completely on.

Change Duty Cycle

- pwm.ChangeDutyCycle(Duty Cycle), this function will change the duty cycle after the PWM is started.

Stop PWM

- pwm.Stop(), this function will stop the PWM on the selected pin.

```
import RPI.GPIO as GPIO

import time

GPIO.setmode(GPIO.Board)

GPIO.setup(11, GPIO.OUT)

p = GPIO.PWM(11, 100) //11 is pin number and 100 is max range of PWM.
p.start(0) //Starting point of PWM signal you can select any value between 0 to 100.

while True:

for x in range (0, 100, 1): //Increasing brightness of LED from 0 to 100

p.ChangeDutyCycle(x)

time.sleep(0.1)

for x in range (100, 0, -1): //fading brightness of LED from 100 to 0

p.ChangeDutyCycle(x)
```

```
time.sleep(0.1)
```

4.Adjust the brightness of an LED (0 to 100, where 100 means maximum brightness)usingthe in-built PWM wavelength.

```
From machine import pin,PWM
From time inport sleep
Pwm=PWM(Pin(15))
Pwm.freq(1000)
While True:
For duty in range(62025):
Pwm.duty_u16)duty)
Sleep(0.0001)
For duty in range(65025,0,-1):
Pwm.duty_u16(duty)
Sleep(0.0001)
```

5.Collecting Sensor Data

DHT Sensor interface

Connect the terminals of DHT GPIO pins of Raspberry Pi.

Import the DHT library using *import Adafruit_DHT*

Read sensor data and display on screen

```
import adafruit_dht
import board
dht = adafruit_dht.DHT22(board.D2)
while True:
    try:
        temperature = dht.temperature
        humidity = dht.humidity
        # Print what we got to the REPL
        print("Temp: {:.1f} *C \t Humidity: {}%".format(temperature, humidity))
    except RuntimeError as e:
        # Reading doesn't always work! Just print error and we'll try again
        print("Reading from DHT failure: ", e.args)
    time.sleep(1)
```