



INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES
DE MONTERREY
CAMPUS GUADALAJARA

Introducción a la Programación Programación básica I (Python)

Apuntes de clase

Impartido por:

Dr. Eduardo de Jesús Dávila Meza

para el grupo:

PS5005.401

para estudiantes de:

Quinto Semestre

de la Escuela de:

PrepaTec

Zapopan, Jalisco, México. Agosto - Noviembre 2025.

Contents

Contents	iii
List of Acronyms and Special Terms	v
1 Environment Setup and Development Tools	1
1.1 Introduction	1
1.1.1 An Introduction to Python	1
1.1.2 Some Ubuntu Distributions and Their Python Versions	2
1.1.3 Ubuntu Distribution and Python Version for the Course	2
1.2 Installation of Ubuntu 22.04 LTS (Jammy Jellyfish)	3
1.2.1 Installation of Ubuntu 22 Using the Universal USB Installer	3
1.2.2 First Steps After Installing Ubuntu 22	4
1.3 Configuring the Development Environment	4
1.3.1 Installing Visual Studio Code	4
1.3.2 Installing Recommended VS Code Extensions	4
1.4 Extra: Some Useful Terminal Programs in Ubuntu	5
1.5 Practice Assignment 1: Bootable USB Creation	6
1.5.1 Steps to Create the Bootable USB - For Windows Users	6
1.5.2 Steps to Open Disk Utility - For Mac Users	6
1.5.3 Submission Instructions	6
1.6 Practice Assignment 2: VS Code & Ubuntu Terminal	8
1.6.1 VS Code Session	8
1.6.2 Commands in Ubuntu Terminal	8
1.6.3 Submission Instructions	8
2 print() and input() Functions with String Values	11
2.1 Introduction	11
2.1.1 What is a System?	11
2.1.2 What is a Process?	12
2.1.3 What is a Program?	12
2.1.4 What is an Algorithm?	12
2.1.5 What is a Code?	12
2.1.6 What is a Pseudocode?	12
2.2 Download the Course Repository	12
2.2.1 Using Git	12
2.3 Examples and Exercises	13
2.4 Quiz!	13

2.5 Practice Assignment 2: Academic Introduction	13
2.5.1 Requirements	13
2.5.2 VS Code Session	14
2.5.3 Run your Python Code from an Ubuntu Terminal	14
2.5.4 Submission Instructions	15

List of Acronyms and Special Terms

Acronyms

L

LTS Long-Term Support. — Page: [2](#).

V

VS Code Visual Studio Code. — Page: [4](#).

CHAPTER

Environment Setup and Development Tools

Author: Dr. Eduardo de Jesús Dávila Meza.

 [EduardoDavila-AI](#) 

 **Abstract** - This chapter guides you through the setup of their programming environment for the course **Introduction to Programming - Programming Basics I (Python)**.

The objectives are:

- Understand the Python language's role and brief history.
 - Identify suitable operating system and Python versions for development.
 - Install and configure [Ubuntu 22.04 LTS](#) and required development tools.
 - Set up [Visual Studio Code](#) with essential extensions for Python development.
 - Explore fun and educational terminal programs.
 - Complete a practice task to verify the environment setup.
-

1.1 Introduction

1.1.1 An Introduction to Python

Python is a high-level, interpreted, general-purpose programming language created by Guido van Rossum and first released in 1991. Its design philosophy emphasizes code readability and simplicity, making it a popular choice for both beginners and professionals. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python has a large standard library, active community support, and is widely used in fields such as:

- Web development
 - Data science and machine learning
 - Automation and scripting
 - Software testing
 - Education

1.1.2 Some Ubuntu Distributions and Their Python Versions

Table 1.I shows some [Long-Term Support \(LTS\)](#) Ubuntu releases along with their default Python versions.

Ubuntu Version	Codename	Default Python Version
16.04 LTS	Xenial Xerus	Python 2.7 & Python 3.5
18.04 LTS	Bionic Beaver	Python 3.6
20.04 LTS	Focal Fossa	Python 3.8
22.04 LTS	Jammy Jellyfish	Python 3.10
24.04 LTS	Noble Numbat	Python 3.12

Table 1.I: Ubuntu LTS versions and their default Python versions

For more details, see the [Ubuntu Releases list](#). Currently, the Focal Fossa (20.04), Jammy Jellyfish (22.04), and Noble Numbat (24.04) distributions are actively supported.

1.1.3 Ubuntu Distribution and Python Version for the Course

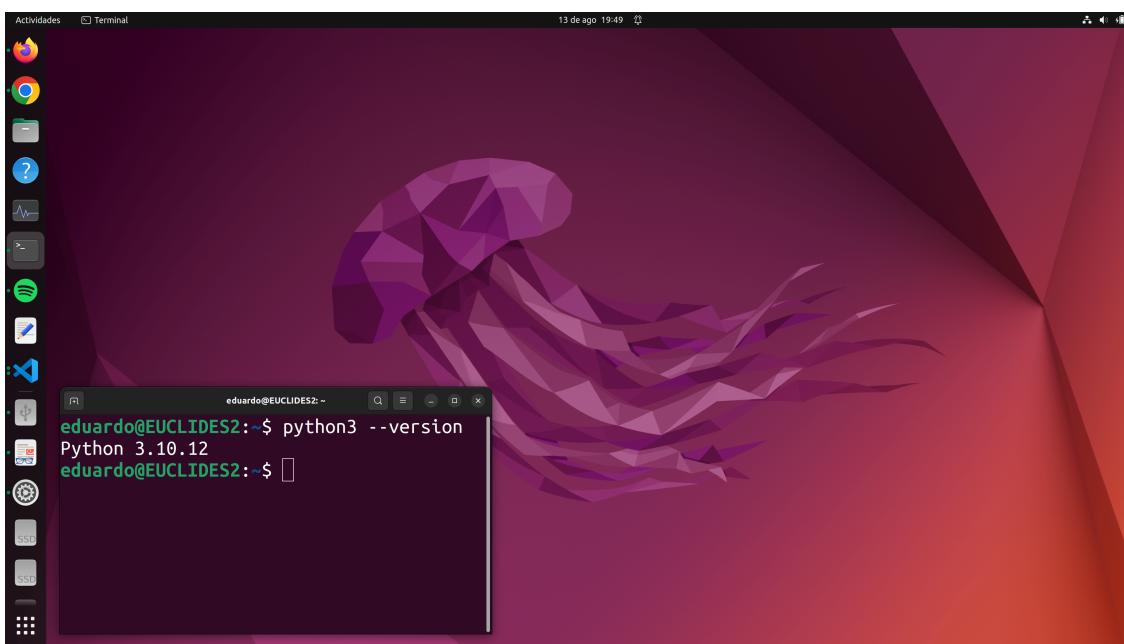


Figure 1.1: Ubuntu 22.04 LTS Jammy Jellyfish with Python 3.10.

For this course, you will use:

- **Operating System:** Ubuntu 22.04 LTS (Jammy Jellyfish),
- **Python Version:** Python 3.10

as shown in Figure 1.1. This combination ensures stability, compatibility with educational resources, and up-to-date features without sacrificing reliability.

1.2 Installation of Ubuntu 22.04 LTS (Jammy Jellyfish)

1.2.1 Installation of Ubuntu 22 Using the Universal USB Installer

1. Download the Universal USB Installer

Visit the [Uptodown website](#) to download the [Universal USB Installer](#), version 2.0.1.4, on [Windows](#).

2. Download the Ubuntu ISO

Download the Ubuntu 22.04.5 LTS (Jammy Jellyfish) ISO - *64-bit PC (AMD64) desktop image*, from the official release page: [Ubuntu releases](#).

3. Create a Bootable USB Drive

Insert a USB drive (minimum 8 GB) and use the Universal USB Installer tool with the downloaded ISO to create a bootable USB stick. Follow the on-screen prompts to properly set up the drive. You may also follow the instructions provided by your instructor, the official guide: [Universal USB Installer](#), or refer to this tutorial: [YouTube Guide](#).

4. Prepare Your Machine (if dual-booting with Windows)

If you are keeping Windows and installing Ubuntu alongside it, you should free some unallocated disk space before installation:

- Defragment the C: drive.
- Additionally, you can shrink the Windows partition to create at least 64 GB of unallocated space (advanced installation).
- If the volume reduction fails, delete old restore points to free up additional space. See, for example, this tutorial: [Windows - Shrinking hard disk volume to create new partition](#).

5. Install Ubuntu on Your Machine

Boot the target computer from the USB drive. Follow the Ubuntu installation wizard to install Ubuntu 22.04 LTS and configure your system settings as needed:

- Select language and keyboard layout.
- Connect to a network (optional during installation).
- Choose installation type (*Normal* or *Minimal*).
- Allocate disk space (at least 64 GB) if you want to *Install Ubuntu alongside Windows* or select *Erase disk and install Ubuntu*.
- Create a user account, password, and set the hostname.



1.2.2 First Steps After Installing Ubuntu 22

Open a terminal from the application menu by typing `terminal` or by pressing **Ctrl+Alt+T**, and then:

- Update the system:

```
$ sudo apt update && sudo apt upgrade -y
```

- Check the installed Python version:

```
$ python3 --version
```

1.3 Configuring the Development Environment

With the following tools and extensions, your development environment will be well-equipped for Python projects.

1.3.1 Installing Visual Studio Code

[Visual Studio Code \(VS Code\)](#) is a versatile code editor that supports various programming languages and development environments. To install it on Ubuntu 22.04, follow these steps:

1. Using the Ubuntu Software Center

- Open the *Ubuntu Software* application from the application menu.
- In the search bar, type [Visual Studio Code](#).
- Locate [Visual Studio Code](#) in the search results and click *Install*.

2. Using the Official .deb Package

- Visit the official [VS Code](#) download page: code.visualstudio.com.
- Click on the **.deb** package suitable for Debian/Ubuntu.
- Once downloaded, open a terminal (**Ctrl+Alt+T**) and navigate to the directory containing the downloaded file.
- Install the package using:

```
$ sudo apt install ./<file>.deb
```

Replace **<file>** with the actual filename.

These methods ensure that [VS Code](#) is added to your system repositories and receives updates automatically.

1.3.2 Installing Recommended VS Code Extensions

Enhance your development experience by installing the following [VS Code](#) extensions:

- **Error Lens** by *Alexander* - Highlights error and warning messages inline in the code editor.



- **Indent-Rainbow** by *oderwat* - Highlights indentation levels with different colors.
- **Jupyter** by *Microsoft* - enables notebook support with Jupyter notebooks, allowing any Python environment to be used as a Jupyter kernel.
- **Python (and Pylance)** by *Microsoft* - Provides rich support for Python, including features such as IntelliSense, linting, and debugging.
- **vscode-icons** by *VSCodium Icons Team* - Adds file icons for better visual identification.

To install these extensions:

1. Open VS Code from the application menu by typing `code`.
2. Navigate to the *Extensions* view by clicking on the square icon (⧉) in the sidebar or pressing **Ctrl+Shift+X**.
3. Search for each extension by name and click *Install*.

1.4 Extra: Some Useful Terminal Programs in Ubuntu

Try installing and running:

- **Neofetch** — displays system information in ASCII art.

```
$ sudo apt install neofetch  
$ neofetch
```

- **Htop** — interactive process viewer.

```
$ sudo apt install htop  
$ htop
```

- **Hollywood** (optional) — creates a “Hollywood movie hacking” terminal effect.

```
$ sudo apt install hollywood  
$ hollywood
```



1.5 Practice Assignment 1: Bootable USB Creation

In this assignment, you will create a bootable USB drive with the ISO image of [Ubuntu 22.04 LTS \(Jammy Jellyfish\)](#) using the *Universal USB Installer* (Windows). After completing the process, you must capture a screenshot showing the folders and files created in the USB drive. This evidence will be submitted in the designated assignment area on Canvas as an image file. The filename must follow the format:

FirstnameLastname_s1_e1.png,

where **s1** corresponds to the session number and **e1** is the evidence number. For example:

EduardoDavila_s1_e1.png.

1.5.1 Steps to Create the Bootable USB - For Windows Users

1. Download the Universal USB Installer, version [2.0.1.4](#), from: [uptodown.com](#).
2. Download the Ubuntu 22.04.5 LTS (Jammy Jellyfish) ISO - *64-bit PC (AMD64) desktop image* - from: [Ubuntu releases](#).
3. Insert a USB drive (minimum 8 GB) and format it with the default settings ([FAT32/exFAT](#) properties).
4. Run the Universal USB Installer to create the bootable USB stick, as described in Section [1.2.1](#). You may also follow the official guide: [Universal USB Installer](#), or refer to this tutorial: [YouTube Guide](#).
5. After completion, open the USB drive in [File Explorer](#) and verify that the Ubuntu boot files and folders (such as **boot**, **EFI**, **casper**, etc.) were created.

1.5.2 Steps to Open Disk Utility - For Mac Users

1. Open the built-in [Disk Utility](#) app to visualize the disk properties of your machine. You may watch this tutorial: [YouTube Guide](#).
2. Remember to bring your [USB Type-C to USB Type-A adapter](#) to the next class so you can complete the installation on your machine.

1.5.3 Submission Instructions

1. Complete the steps above according to your case: either Windows or Mac user.
2. Capture a **screenshot of the USB drive contents** (see Figure [1.2](#)) for Windows users, or a **screenshot of the Disk Utility** app for Mac users.
3. Save the screenshot in PNG or JPG/JPEG format.
4. Name the file according to the required format: [FirstnameLastname_s1_e1.png](#).
5. Submit the file through the assignment submission area on Canvas.

NOTE: Your screenshot must clearly display either the USB drive contents (Windows) or the Disk Utility app (Mac). Submissions that appear copied, unclear, or altered will be considered invalid and may result in a score of 0 for this assignment.



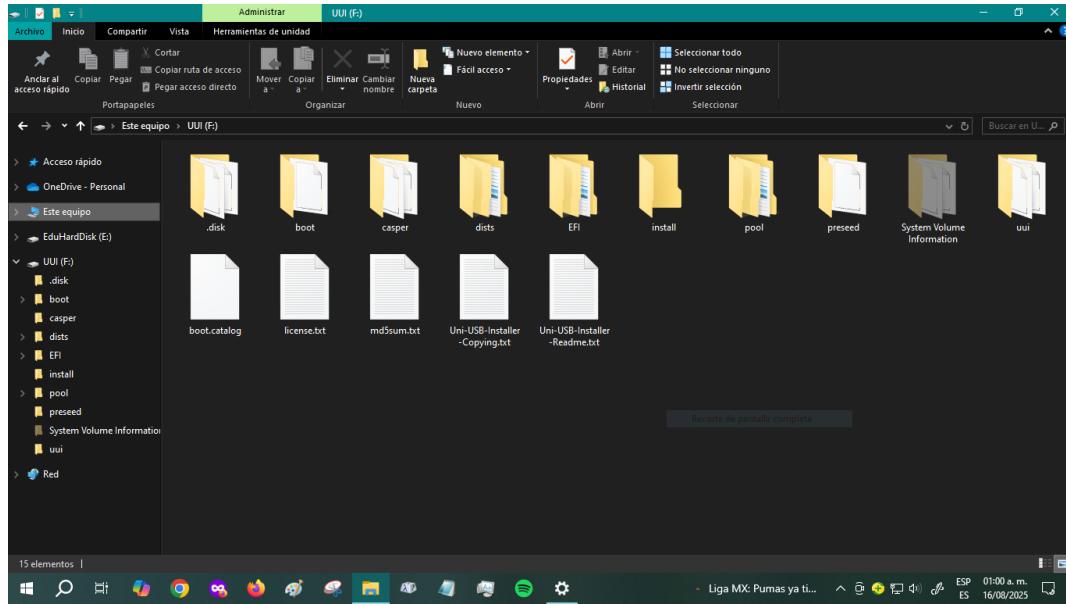


Figure 1.2: Example screenshot of bootable USB contents (Windows users).

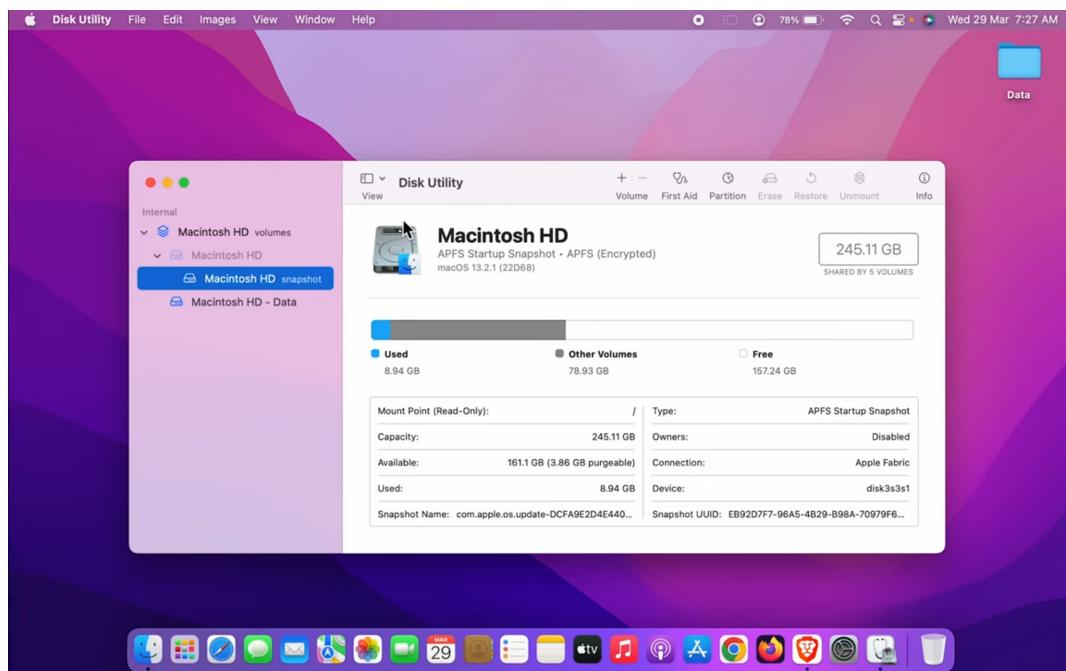


Figure 1.3: Example screenshot of Disk Utility (Mac users).

1.6 Practice Assignment 2: VS Code & Ubuntu Terminal

In this assignment, you will open a [Visual Studio Code \(VS Code\)](#) session and run specific terminal commands in an [Ubuntu Terminal](#), capturing evidence of successful execution via a screenshot. This evidence will be submitted in the designated assignment area on Canvas as an image file. The filename must follow the format:

FirstnameLastname_s1_e2.png,

where **s1** corresponds to the session number and **e2** is the evidence number. For example:

EduardoDavila_s1_e2.png.

1.6.1 VS Code Session

1. Open [VS Code](#) from the application menu by searching for `code`, or launch it from a terminal with: `code`.
2. Navigate to the *Extensions* view by clicking on the square icon (⧉) in the sidebar or pressing **Ctrl+Shift+X**. Ensure that the extensions listed in Section 1.3.2 are installed.
3. Position the [VS Code](#) window on either the left or right half of the screen to prepare for a side-by-side view with the terminal (see Figure 1.4).

1.6.2 Commands in Ubuntu Terminal

1. Open a terminal (**Ctrl+Alt+T**) and display your machine's hostname and username:

```
$ hostname  
$ whoami
```

2. In the same terminal, verify that [Python 3.10](#) is installed by running:

```
$ python3 --version
```

3. Move the terminal window to the opposite side of the screen from [VS Code](#), so both are visible simultaneously (see Figure 1.4).

1.6.3 Submission Instructions

1. Complete all steps as described above.
2. Capture a **screenshot showing both the VS Code and terminal windows**, ensuring that the terminal output is readable (see Figure 1.4).
3. Save the screenshot in PNG or JPG/JPEG format.
4. Name the file according to the required format: **FirstnameLastname_s0.png**.
5. Submit the file through the assignment submission area on Canvas.

NOTE: Your screenshot must clearly display both your machine's username and hostname to verify authenticity. Submissions that appear copied, unclear, or altered will be considered invalid and may result in a score of 0 for this assignment.



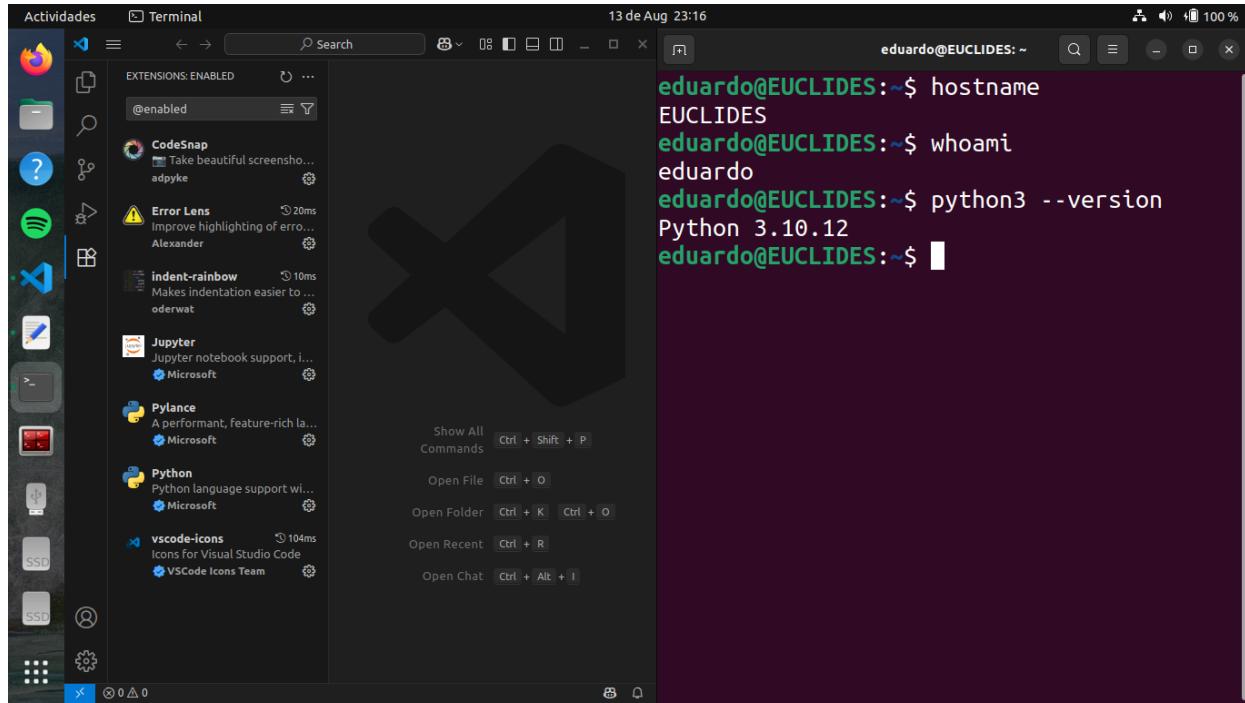


Figure 1.4: Example screenshot showing the [VS Code](#) session and the terminal output.

CHAPTER

`print()` and `input()` Functions with String Values

Author: Dr. Eduardo de Jesús Dávila Meza.

 [EduardoDavila-AI](#)

 **Abstract** - In this chapter, you will learn the basics of interacting with the computer through `print()` and `input()` functions of Python.

The objectives are:

- Distinguish between process, system, program, algorithm, code, and pseudocode.
 - Understand the role of `print()` in displaying output.
 - Work with string values in Python.
 - Use `input()` to capture user data.
 - Practice basic programming through guided exercises.
-

2.1 Introduction

Before diving deeper into programming with Python, it is important to clarify some fundamental concepts that form the basis of computer science and programming.

2.1.1 What is a System?

A **system** is an organized collection of components (hardware, software, data, and users) that interact to apply a process. Examples include a banking system, an operating system, or an educational learning platform.

2.1.2 What is a Process?

A **process** is a sequence of actions or operations carried out to transform inputs into outputs. In computing, a process often refers to a running instance of a program that performs tasks.

2.1.3 What is a Program?

A **program** is a set of instructions written in a programming language that tells a computer what to do. When executed, a program performs specific tasks such as calculating, storing data, or controlling devices.

2.1.4 What is an Algorithm?

An **algorithm** is a step-by-step procedure for solving a problem or completing a task. Algorithms can be described using natural language, diagrams, or pseudocode.

2.1.5 What is a Code?

Code is the implementation of an algorithm written in a specific programming language such as Python. It is the human-readable set of instructions that the computer translates into machine operations.

2.1.6 What is a Pseudocode?

Pseudocode is a simplified way of writing algorithms using structured but informal language. It is not executed by the computer but serves as a bridge between an algorithm and real code.

2.2 Download the Course Repository

To follow along with the examples and exercises in this module, you need to download the course repository.

2.2.1 Using Git

Before you begin, make sure that [Git](#) is installed on your system. If it is not, you can install it with:

```
$ sudo apt install git
```

Once installed, open a terminal and run the following command:

```
$ git clone https://github.com/eDavila-DrRaccoon/python_course_repo.git
```

This will create a folder named `python_course_repo` containing all course materials.

NOTE: It is recommended to clone the repository to the [Documents](#) folder, as well as copy the `python_ws` subfolder from the repository to an independent folder to avoid overwriting your work. Additionally, it is recommended to use [Git](#) to keep the repository updated with the latest changes.



2.3 Examples and Exercises

The repository includes examples and exercises to help you practice. Consult the files provided in the **module** folder of the workspace.

- Jupyter Notebook files (**.ipynb**) for interactive coding.
- Python script files (**.py**) for standard execution.

These examples focus on:

1. Using **print()** to display text and results.
2. Manipulating and combining string values.
3. Capturing input with **input()**.

2.4 Quiz!

After completing the examples, test your knowledge by completing the quiz in **Module 2** of the course on Canvas.

2.5 Practice Assignment 2: Academic Introduction

In this assignment, you will:

“Write a Python program that asks the user a series of questions about themselves to display a short academic introduction, and then run it from an Ubuntu Terminal”.

You will then capture evidence of successful execution via a screenshot. This evidence will be submitted in the designated assignment area on Canvas as an image file. The filename must follow the format:

FirstnameLastname_m2_e1.png,

where **m2** corresponds to the session number and **e1** is the evidence number. For example:

EduardoDavila_m2_e1.png.

2.5.1 Requirements

1. Use the **input()** function to ask the user for:
 - Name
 - Age
 - Degree or field of study
 - Student registration number
 - Favorite subject or course
 - Hobby
 - Favorite food
 - Possible experience in Python



- Desired career to study
2. Do not store the answers in variables.
 3. Use `strings` (concatenation or f-strings) to create a formatted introduction that includes all the collected information.
 4. Display the introduction using `print()`.

Example Interaction

```

1 | Enter your name: Alice
2 | Enter your age: 20
3 | What degree/field are you studying? High School
4 | What is your student registration number (SRN)? AOXXXXXX
5 | What is your favorite subject or course? Python course
6 | What is your favorite hobby? Reading
7 | What is your favorite food? Pizza
8 | What is your experience in Python? Just starting!
9 | What career would you like to study? Software Engineering

```

Output:

Hello! My name is Alice and I am 20 years old.
I am currently studying High School and my SRN is AOXXXXXX.
My favorite subject/course is Python course. In my free time, I enjoy Reading,
and my favorite food is Pizza. Regarding Python, my experience is: Just starting!
In the future, I would like to pursue a career in Software Engineering.

Hint

- Remember: `input()` always returns a string.
- You can use `f-strings` for cleaner formatting, e.g.:

```
f"My name is {input('Enter your name: ')} and I am {input('Enter your age: ')}
↪ years old."
```

2.5.2 VS Code Session

1. Open **VS Code** from the application menu by searching for `code`, or launch it from a terminal with: `code`.
2. Navigate to the `module2` subfolder from the workspace directory (`.../python_ws/module2`), and edit your Python script to complete the practice assignment.
3. Position the **VS Code** window on either the left or right half of the screen to prepare for a side-by-side view with the terminal (see Figure 2.1).

2.5.3 Run your Python Code from an Ubuntu Terminal

1. Open a terminal (**Ctrl+Alt+T**), navigate to the `module2` subfolder from the workspace directory (`.../python_ws/module2`), and run your Python file with the following command:

```
$ python3 m2_p4_practice.py
```

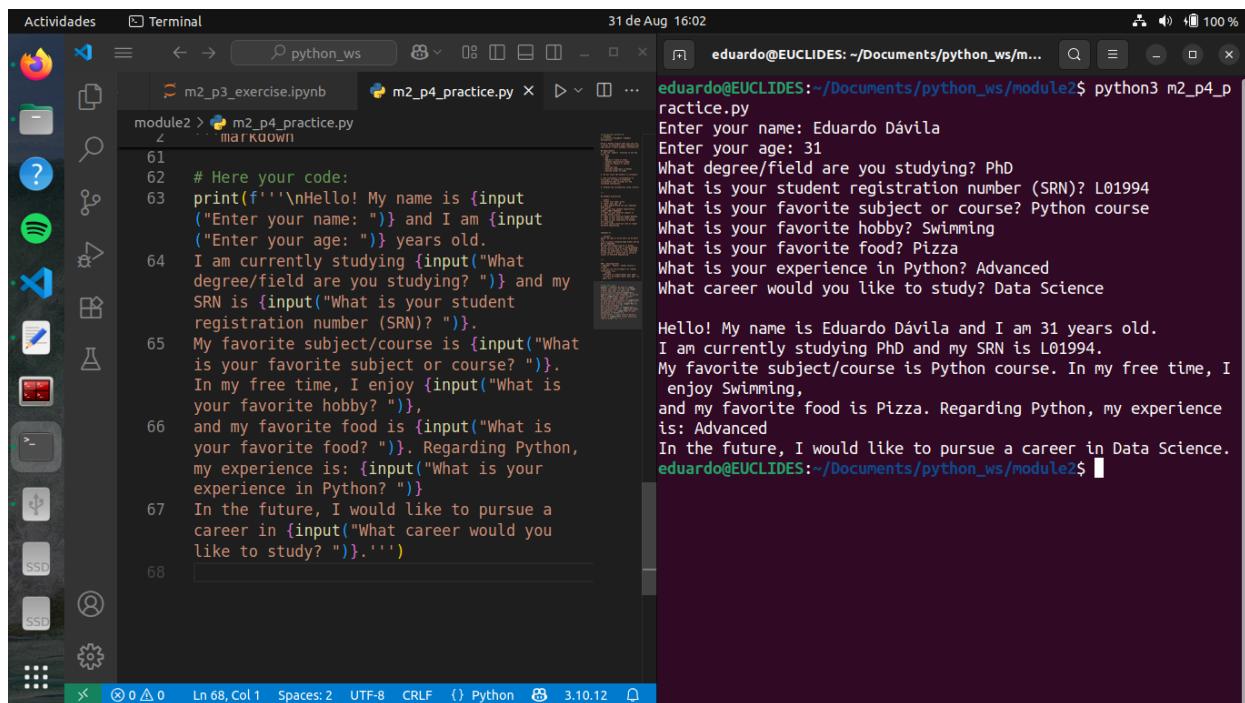


2. Move the terminal window to the opposite side of the screen from **VS Code**, so both are visible simultaneously (see Figure 2.1).

2.5.4 Submission Instructions

1. Complete all steps as described above.
2. Capture a **screenshot showing both the VS Code and terminal windows**, ensuring that the terminal output is readable (see Figure 2.1).
3. Save the screenshot in PNG or JPG/JPEG format.
4. Name the file according to the required format: **FirstnameLastname_s0.png**.
5. Submit the file through the assignment submission area on Canvas.

NOTE: Your screenshot must clearly display both your machine's username and hostname to verify authenticity. Submissions that appear copied, unclear, or altered will be considered invalid and may result in a score of 0 for this assignment.



The screenshot shows a Mac desktop with two windows side-by-side. On the left is the VS Code interface, displaying a Python script named 'm2_p4_practice.py' with code related to user input and string manipulation. On the right is a terminal window titled 'Terminal' with the command 'python3 m2_p4_p'. The terminal output shows the user's responses to various prompts, including their name ('Eduardo Dávila'), age ('31'), degree ('PhD'), student ID ('L01994'), favorite subject ('Python course'), hobby ('Swimming'), food ('Pizza'), Python experience ('Advanced'), and future career ('Data Science').

Figure 2.1: Example screenshot showing the **VS Code** session and the terminal output.