Lab. Manual دليل التجارب –

Eng. Mohammed AL-manari

2025-2026

No.	Subject	Page
1	Safety procedures in the laboratory	2
2	The lab's equipment and tools	3
3	Lab (1) - Starting with Environments of python	4
4	Lab (2) - Being familiar with Anaconda	8
5	Lab (3) - Basic data types & Strings	11
6	Lab (4) - Lists Operations, Tuples and Dictionaries	14
7	Lab (5) - Functions & Loops (for and while with if condition)	17
8	Lab (6) - Python Libraries	23
9	Lab (7) - SKLEARN Data Preparation & ML Algorithms	30
10	Lab (8) - Cleaning Data and Data Scaling in SKLEARN	33
11	Lab (٩) - SKLEARN Work in ML	41
12	Lab (١٠) - SKLEARN Work in Neural Network	50
13	References	62

إجراءات السلامة في المعمل

Safety procedures in the laboratory

توفر منظم كهربائي لكل معمل.

توفر فاصل كهربائي أتوماتيكي لفصل الكهرباء عن الأجهزة في حال حدوث مشاكل كهربائية.

توفر طفاية حريق.

عدم تحريك أي جهاز من الموضع الذي وضعه فيه الفني المختص بالمعمل.

إجراء الفحص الدوري بداية كل فصل دراسي لكافة القطع والأجهزة في المعمل والتأكد من سلامتها.

التأكد من تشغيل الأجهزة بطريقة صحيحة.

التأكد من إغلاق الأجهزة عند نهاية العمل.

فصل التيار الكهربائي عن المعمل عند إنتهاء الدوام.

The lab's equipment and tools

No.	Equipment and Tools	Remarks
	Personal Computers	
	python	
	VS Code OR Anaconda	
	Download libraries that we need	

Lab. (1)

Starting with Environments of python

Objectives:

- Describe the characteristics and purposes of the different environments in python.
- Discus difference types of programs that providing python language.

The needed time for this Lab.: 2 hours.

Why Python?

• Python has become the language for many data science applications. It combines the power of

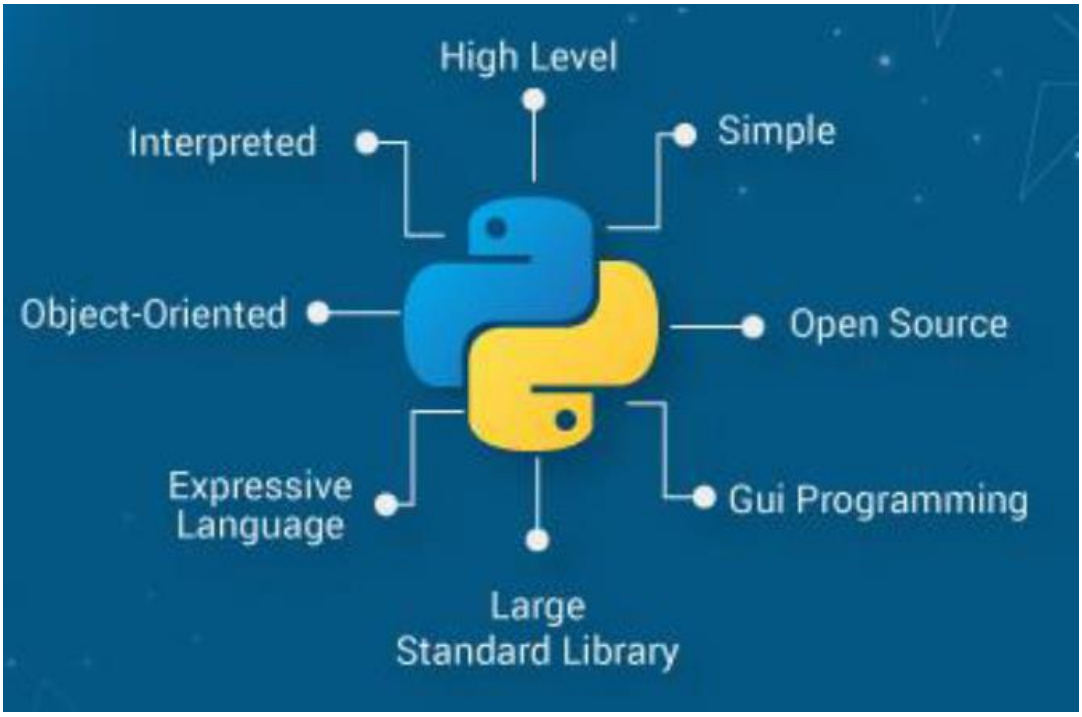
general-purpose programming languages with the ease of use of domain-specific scripting languages

like MATLAB.

Simple syntax.

Less coding.

Inbuilt libraries for AI projects.



1.Installing Python

Step 1: Download Python

Open a web browser and go to Python's official website.

https://www.python.org/downloads/release/python-3123/

Click the "Download Python" button (the website suggests the correct version for your OS).

Click on "Windows installer (64-bit)" or "macOS installer" depending on your system.

Save the installer file.

Step 2: Run the Python Installer

Locate the downloaded installer and double-click to run it.

Add path.

Step 3: Run the installer.

For Windows:

1. Check the box “Add Python 3.12 to PATH”.
2. Click “Customize Installation”.
3. Select all options (pip, documentation, etc.) → Next.
4. Check “Install for all users” and “Add to environment variables”.
5. Click “Install”. Wait until it finishes, then click “Close”.

For Mac:

1. Double-click the downloaded .pkg file.
2. Follow the installation wizard and accept the license.
3. Wait until it finishes and click “Close”.

Verification: Open Command Prompt or Terminal and type:

python --version
pip --version

2. **Installing Visual Studio Code (VS Code)**

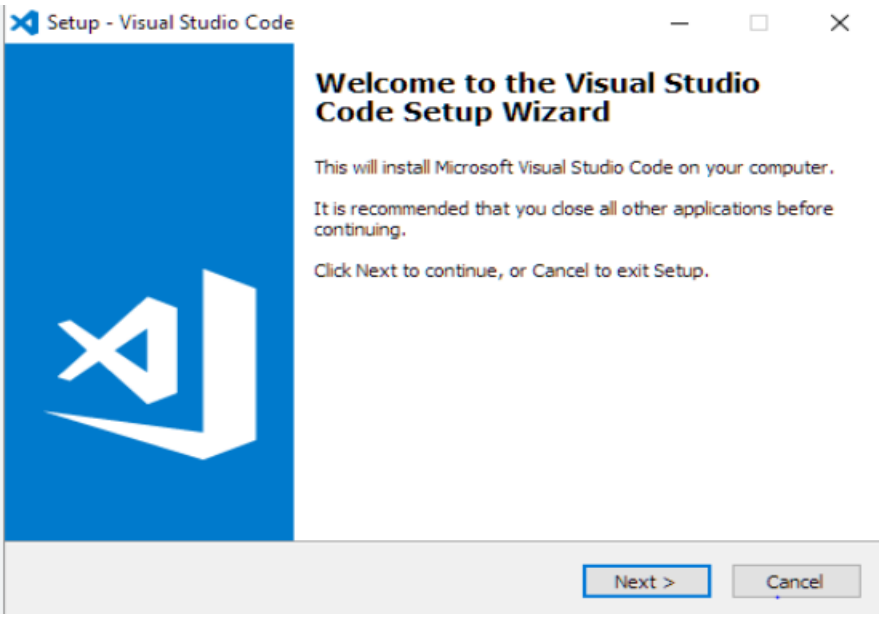
Step 1: Download VS Code

Go to VS Code's official website.

Click Download **for your OS (Windows, macOS, or Linux).**

Go to https://code.visualstudio.com/ and download VS Code.
Step 2: Run the installer and accept the agreement.
Step 3: Check these options:

✓ Add "Open with Code" to Windows Explorer
✓ Add to PATH



^

✓ Register Code as editor for supported files

Step 4: Click “Install”, then “Finish”.

4. Setting Up Python in VS Code

1. Open VS Code.
2. Go to Extensions (icon on left).
3. Search for "Python" and install the extension by Microsoft.
4. Also install "Pylance" and "Code Runner".
5. Create a folder named AI_Course.
6. Create a file named test.py and write:

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

7. Run the file using the play button or press Ctrl + F5.

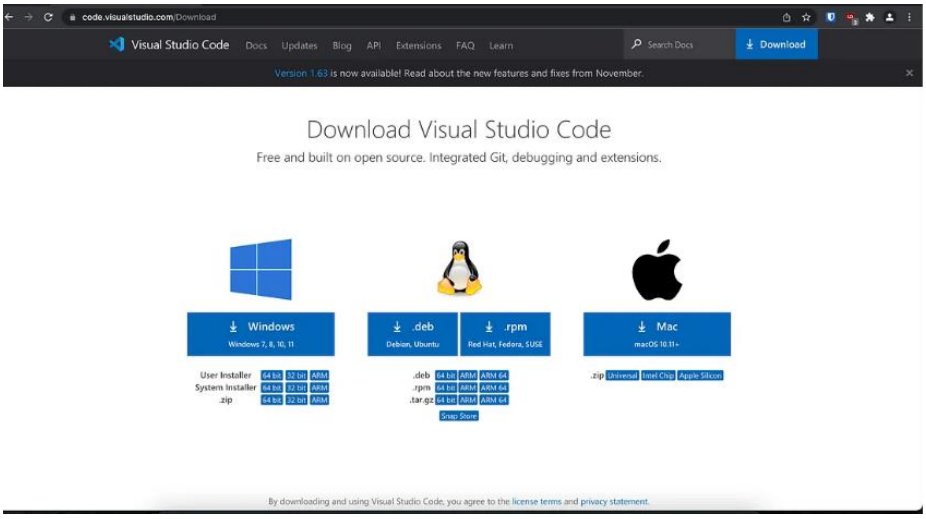
5. Installing Jupyter Notebook in VS Code

Open VS Code terminal (View → Terminal) and type:

```
pip install jupyter
```

After installation, press Ctrl + Shift + P → "Create New Jupyter Notebook".
Run this test code:

```
print("Jupyter is working perfectly!")
```



Step 3: Configure VS Code for Python

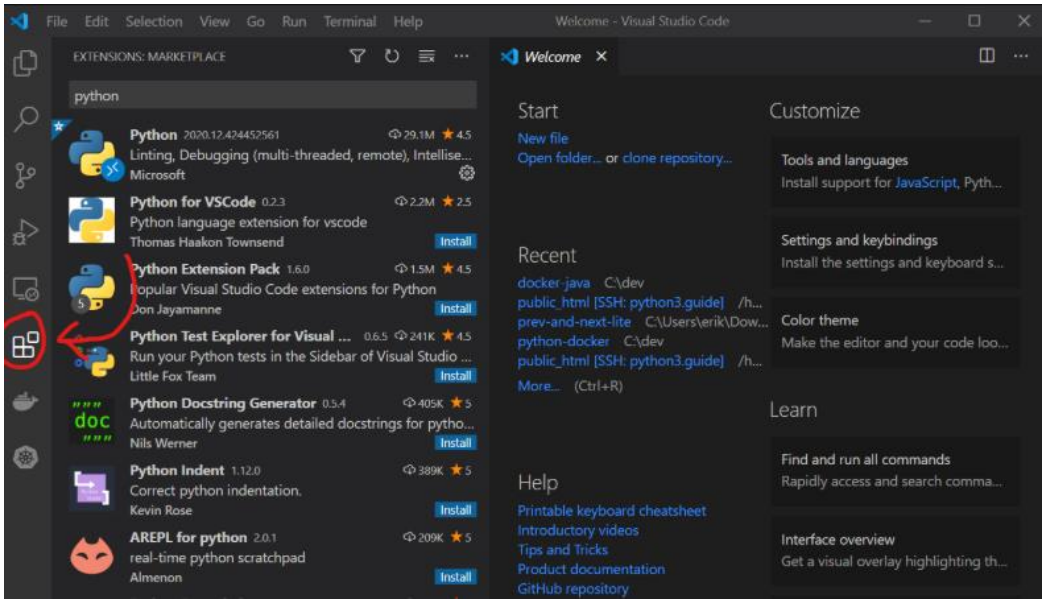
Open VS Code.

Install the Python extension**:**

Click the Extensions icon (Ctrl+Shift+X)**.**

Search for "Python" **(by Microsoft).**

Click Install**.**



This guide covers the fundamental programming concepts you'll need to analyze decantation experiments. We'll explore variables for storing measurements, functions for reusable calculations, and loops for processing multiple data points efficiently. Each concept includes multiple practical examples relevant to laboratory work.

A basic Python curriculum can be broken down into 4 essential topics that include:

Data types (int, float, strings).

Compound data structures (lists, tuples, and dictionaries).

Conditionals, loops, and functions.

Object-oriented programming.

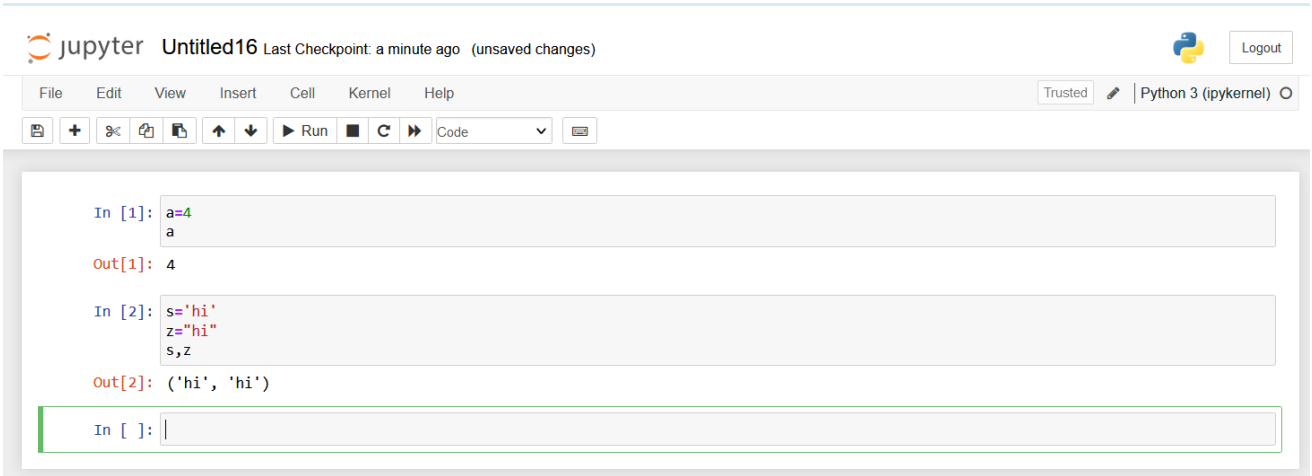
Basic data types:

Like most languages, Python has a number of basic types including **integers**, **floats**, **Booleans**, and **strings**. These data types behave in ways that are familiar from other programming languages.

Numbers: Integers and floats work as you would expect from other languages:

The very first step is to understand how Python interprets data.

Starting with widely used data types, you should be familiar with integers (int), floats (float), strings (str), and Booleans (bool).



function. Common data types include:

□□ **int (for integer)**

□□ **float**

□□ **str (for string)**

□□ **list**

□□ **tuple**

□□ **dict (for dictionary)**

□□ **set**

- **bool (for Boolean True/False)**

String:

Knowing how to deal with textual data and their operators comes in handy when dealing with the string data type. Practice these concepts:

Concatenating strings using + .

Splitting and joining the string using the split() and join() methods.

Changing the case of the string using lower () and upper() method.

Working with substrings of a string

To access the first character on the string

□□□S[0]

You'll get this output:

'H□