

---

# Introduction to Artificial Intelligence

## Lab 1(week 1) -Introduction to Python Programming

### 2024 - 2025

---

02 February 2025

### Objectives

- Setting Up Python Environment Install Python and a package manager (e.g., Anaconda).
- Introduction to Jupyter Notebooks.
- Install Spyder.
- Basic Python Programming Variables, data types, and operators. Control structures (if statements, loops).

### Part 1: Environment Preparation

#### 1. Setting Up Python Environment Install Python and a package manager Anaconda:

##### Step 1: Install Anaconda:

- Visit the <https://www.anaconda.com/> and download the latest version of Anaconda suitable for your operating system (Windows, macOS, or Linux).
- Follow the installation instructions provided on the website. During installation, you can choose to add Anaconda to your system's PATH, which makes it easier to use from the command line.

##### Step 2: Check the installation:

Open a command prompt or terminal window. To check if Anaconda is installed, type the following command:

```
conda --version
```

This command should display the version of conda, the package manager that comes with Anaconda. If 'conda' is not recognized as a command, you need to add it to the global PATH on your system by following these steps:

For windows:



Figure 1: (Step 1)

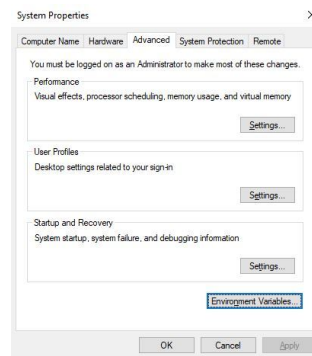
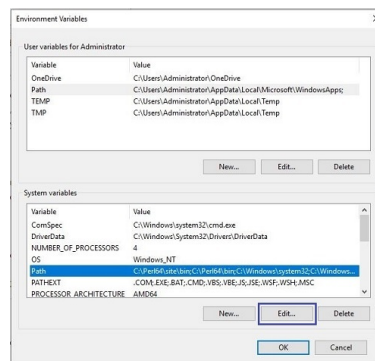
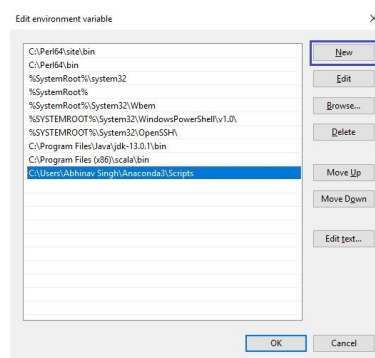


Figure 2: (Step 2)

We need to modify the "Path" variable within the System variables to include the path to the Anaconda environment. Choose the "Path" variable and click on the Edit button, as illustrated below:



Upon doing so, a list of various paths will be displayed. Click on the "New" button, then proceed to add the path where Anaconda is installed.



Click on OK, save the settings, and the process is complete! To verify the installation's accuracy, open

the command prompt and enter "anaconda-navigator." If installed correctly, this will launch the Anaconda Navigator app.

#### **For Linux:**

There are multiple methods for installing Anaconda, but we will focus on the most straightforward approach, which involves installing Anaconda using the terminal.

Follow the steps outlined in the instructions on how to install Anaconda on Linux. Typically, the Path variable is configured automatically during the installation process on Linux. However, if needed, it can also be manually set by following these steps:

#### **Step 1:** Go to Application – > Accessories – > Terminal

To configure the Environment Variable, type the following commands in the Terminal, utilizing the installation path:

```
% export ANACONDA = /home/...path to anaconda/anaconda3
%export PATH = $PATH:/home/...path to anaconda/anaconda3/bin
%export PATH = $PATH:/home/...path to anaconda/anaconda3/Scripts
```

The process is now complete! To verify the installation's success, open the Terminal and type "anaconda-navigator." If installed correctly, this will launch the Anaconda Navigator application.

#### **Step 3: Create and Activate a Virtual Environment**

Different projects may have different dependencies and versions. Isolating these dependencies is crucial, it helps avoid conflicts and ensures that each project has access to the specific packages it needs without interfering with others.

When using Anaconda specifically, the process of creating and activating a virtual environment involves using the conda package manager. The following are common commands:

```
conda create --name myenv
```

where "myenv" reflects the name of your virtual environment. To activate the virtual environment, you can use the following command:

```
conda activate myenv
```

To deactivate it, you simply use:

```
conda deactivate
```

To check for existing Virtual environments:

```
conda info --envs
```

#### **Step 4: Install Additional Packages:**

Once the virtual environment is activated, you can install additional packages using the `conda install` command. For example, to install popular data science libraries, you can use:

```
conda install numpy pandas matplotlib seaborn jupyter
```

This command installs NumPy, Pandas, Matplotlib, Seaborn, and Jupyter Notebook in your virtual environment.

## 2. Introduction to Jupyter Notebook:

Jupyter Notebooks are a popular, open-source tool that allows you to create and share live code, equations, visualizations, and narrative text. They are widely used in data science, machine learning, research, and education. Jupyter Notebooks support multiple programming languages, but they are most commonly used with Python.

### Step 1. Opening a Notebook:

- After installing Jupyter Notebook through Anaconda, you can open it by typing ‘jupyter notebook’ in the command line. This will launch the Jupyter Notebook server, and you can access the interface through your web browser.

### Step 2. Creating a New Notebook:

- In the Jupyter Notebook interface, click on “New” and choose a kernel (e.g., Python 3) to create a new notebook.

### Step 3. Working with Cells:

- Jupyter Notebook consists of cells, which can be of two main types: code cells and markdown cells. You can switch between them using the drop-down menu in the toolbar.
- These cells are used for writing and executing code. Python is the default kernel, but you can choose other languages.

### Step 4. Executing Code:

- To execute a code cell, select it and press **Shift + Enter** or click the “Run” button in the toolbar. The output will be displayed below the cell.

### Step 5. Markdown Cells:

- Use markdown cells for adding formatted text, headers, lists, and equations. You can render the markdown by executing the cell.

### Step 6. Saving and Exporting:

- Save your work using the “Save” button or **Ctrl + S** (Windows/Linux) or **Command + S** (Mac). You can also export the notebook in various formats, such as HTML, PDF, or as a script.

### Note:

Google Colab is essentially a Jupyter Notebook environment hosted on Google’s servers. This means you can write, run, and share Jupyter notebooks directly in your web browser without the need for any local installation.

## Part 2: Start working with python

Now that you have successfully launched your Jupyter Notebook, either locally or via Google Colab, you can begin learning the basics of Python, such as variables, conditions, loops, lists, dictionaries, and functions, using the websites below.

- Imperial College Python Tutorial: <https://python.pages.doc.ic.ac.uk/cpp/lessons/cpp/01-intro/index.html>
- W3schools Python Tutorial: [https://www.w3schools.com/python/python\\_intro.asp](https://www.w3schools.com/python/python_intro.asp)