February 19, 2025

ITLPA701: PYTHON AND FUNDAMENTALS OF AI

LEARNING UNIT 1– PREPARE PYTHON ENVIRONMENT

Mr. Etienne NTAMBARA

ntambaraienne94@gmail.com

Assistant Lecturer in ICT Department

Rwanda Polytechnic, IPRC-HUYE

Homepage: https://94etienne.github.io/profile/

IPRC HUYE

Learning hours: 9

Content

Learning Outcome 1.1: Identify Python Version

- Definition of terms
- Characteristic of python
- Application of python

Content

Learning Outcome 1.2: Add python directory

- Local Environment set up
- Getting python set up
- Set up path

Content

Learning Outcome 1.3: Identify Library

- Definition of key terms
- Identification of problem
- Getting library

1.1: Identify Python Version

Definition of terms

- **Python**: is a programming language commonly used for developing websites and software, task automation, data analysis, and data visualization. **Key features of Python include**: Readability, Interpretation, Object-Oriented, Dynamic Typing and Extensive Standard Library.

1.1: Identify Python Version

Definition of terms

Key features of Python include:

Readability: Python's syntax is designed to be clear and readable, which makes it easy for developers to write and maintain code.

Interpretation: Python is an interpreted language, which means that the source code is executed line by line by an interpreter. This makes it easier to debug and test code.

Object-Oriented: Python supports object-oriented programming, allowing developers to structure code using classes and objects.

Dynamic Typing: Python is dynamically typed, meaning that variable types are interpreted at runtime, making it more flexible.

Extensive Standard Library: Python comes with a large standard library that includes modules and packages for various purposes, reducing the need for developers to write code from scratch.

1.1: Identify Python Version

Definition of terms

- **Programming language:** a programming language is a formalized set of instructions that a computer can interpret and execute to perform specific tasks or operations.
- Key characteristics of programming languages include: Syntax, Semantics, Abstraction, Variables and Data Types, Control Structures, Functions or Procedures, Libraries and Frameworks.

1.1: Identify Python Version

Definition of terms

- **Use of python**

1. **Server**: Python is used for server-side scripting, allowing developers to embed Python code within web pages or server-side scripts to create dynamic content.
2. **Software development**: Python is often used to develop the back end of a website or application.

Python offers several frameworks for web development. Commonly used ones include Django and Flask.

1.1: Identify Python Version

Definition of terms

- **Use of python**

3. Mathematics

Python is a powerful programming language that is widely used in various fields, and mathematics is no exception.

Python provides extensive support for mathematical operations and computations through built-in functions, libraries, and modules.

4. System scripting: A script, in the context of computer programming, refers to a set of instructions or commands written in a

scripting language that is executed by a computer. This means that the script's instructions are executed line by line by an interpreter or runtime environment.

1.1: Identify Python Version

Definition of terms

➢ Use of python

Examples of scripts include:

- **Shell Scripts:** Used for automating tasks in a Unix or Linux shell environment. Python Scripts: Python is often used for scripting tasks, thanks to its readability and versatility. i.e Automating backup
- Scripting is a very common practice among Python programmers. It's used for automation of daily tasks, reporting, server management, security, social media management, business growth and development, financial trading, automating software and many other intelligent solutions.

1.1: Identify Python Version

Definition of terms

➢ Use of python

Examples of scripts include:

Server-side

- Web Development:

Django: A high-level web framework that follows the "don't repeat yourself" (DRY) principle. It facilitates rapid development of web applications with a clean and pragmatic design.

- **Flask:** A lightweight and flexible web framework that is easy to use and suitable for small to medium-sized web applications. Flask provides the essentials for building web applications without imposing too much structure.
- **API Development:**
- Python is often used to build RESTful APIs (Application Programming Interfaces) that allow communication between different software systems. Libraries like Flask and Django REST framework make it easy to develop robust APIs.
- Backend Services:
- Python is employed in building the backend logic for various services, handling tasks such as data processing, authentication, authorization, and business logic.

1.1: Identify Python Version

Definition of terms

➢ Use of python

Examples of scripts include:

- Content Management Systems (CMS):
- Python can be used to create content management systems for websites. Examples include Wagtail, a Django-based CMS, and Plone, a content management system built on top of Zope.
- Data Processing and Analysis:
- Python is used for server-side data processing and analysis tasks. Libraries like NumPy, Pandas, and SciPy are commonly used for handling and analyzing large datasets.
- Task Automation:

1.1: Identify Python Version

Definition of terms

➢ Use of python

Examples of scripts include:

- Python scripts running on servers are commonly used for task automation, whether it's periodic maintenance tasks, file processing, or system monitoring.
- Microservices Architecture:
- Python is often part of microservices architectures, where small, independent services communicate with each other to build scalable and maintainable applications.
- Server-side Scripting:
- Python is used for server-side scripting, allowing developers to embed Python code within web pages or server-side scripts to create dynamic content.
- Networking:
- Python is utilized for server-side networking tasks, such as building network servers, handling network protocols, and implementing network-related functionalities.
- Cloud Computing:
- Python is commonly used for interacting with cloud services and APIs. Platforms like AWS, Azure, and Google Cloud provide Python SDKs for managing resources and deploying applications.

1.1: Identify Python Version

Definition of terms

- **Use of python**

Examples of scripts include:

- Machine Learning Deployment:
- Python is used for deploying machine learning models on the server side. Frameworks like Flask or FastAPI are often employed to create APIs for serving machine learning predictions.

1.1.2 Characteristic of python

- Python is a versatile programming language known for its simplicity, readability, and flexibility.

Here are some key characteristics of Python:

- **Readable and Simple Syntax:** Python emphasizes readability and a clean syntax, making it easy for developers to express concepts with fewer lines of code. This readability is supported by the use of indentation to define code blocks.
- **Interpreted Language:** Python is an interpreted language, meaning that the source code is executed line by line by an interpreter at runtime. This allows for quick testing and debugging.

1.1.2 Characteristic of python

- **High-Level Language**: Python is a high-level programming language, abstracting low-level details and providing a more straightforward and human-readable syntax. This makes it accessible for beginners and efficient for experienced developers.
- **Object-Oriented Programming (OOP):** Python supports object-oriented programming, allowing developers to structure code using classes and objects. Encapsulation, inheritance, and polymorphism are fundamental concepts in Python's OOP paradigm.
- **Dynamic Typing**: Python is dynamically typed, meaning that the data type of a variable is interpreted at runtime. This provides flexibility but requires careful attention to variable types during development.

1.1.2 Characteristic of python

- **Cross-Platform Compatibility:** Python code is generally platform-independent. A Python program written on one operating system (e.g., Windows) can run on another (e.g., Linux) with minimal or no modifications.
- **Extensive Standard Library:** Python comes with a comprehensive standard library that includes modules and packages for various tasks. This eliminates the need for developers to write code from scratch for common functionalities.
- **Integration Capabilities:** Python can easily integrate with other languages and technologies, making it a popular choice for building complex and integrated systems. It supports interfacing with C, C++, and Java, among others.

1.1.3 Application of python

- Based on Flexibility:

1. **Web Development**: Python's frameworks like Django and Flask make it flexible for creating web applications of varying complexity.
2. **Automation and Scripting**: Python's simplicity and adaptability make it highly flexible for scripting tasks and automating workflows.
3. **Scientific Computing**: Python's libraries like SciPy and Matplotlib provide flexibility for performing various scientific tasks, from visualization to numerical analysis.
4. **Embedded Systems**: Python's lightweight frameworks make it suitable for programming embedded systems and IoT devices

1.1.3 Application of python

- Based on Audience:

1. **Data Science and Machine Learning**: Targeted towards data scientists, researchers, and AI engineers who rely on Python's libraries like NumPy, Pandas, Scikit-learn, and TensorFlow.
2. **Game Development**: Primarily for game developers and hobbyists using libraries like Pygame.
3. **Network Programming**: Designed for network engineers and developers creating tools and applications for network systems.
4. **Desktop GUI Applications**: Geared towards developers building user-friendly desktop applications for end-users.

1.1.3 Application of python

- Based on Prerequisite:

1. **Web Development**: Requires knowledge of web technologies like HTML, CSS, JavaScript, and server-side scripting.
2. **Data Science and Machine Learning**: Requires a basic understanding of mathematics, statistics, and machine learning concepts.
3. **Scientific Computing**: Assumes knowledge of numerical methods and data visualization techniques.
4. **Game Development**: Requires understanding of game mechanics and design principles.
5. **Network Programming**: Requires knowledge of networking protocols and systems.
6. **Desktop GUI Applications**: Assumes familiarity with event-driven programming and GUI design.
7. **Embedded Systems**: Requires knowledge of hardware interfaces and low-level programming.

QUIZ

- Duration Five minutes
- https://forms.gle/3Rb8KcQtU79HsbPg6

1.2 Add python Directory

1.2.1 Local Environment set up

Setting up a local development environment for Python involves installing the necessary tools and libraries on your computer.

Here are the general steps for setting up a Python environment on your local machine:

1.2.1.1 Windows

Steps to Install Python on Windows:

1. Download Python Installer:

Go to the https://www.python.org/downloads/

Click on the "**Download Python**" button for the latest version suitable for Windows.

1.2 Add python Directory

2. Run the Installer:

Locate the downloaded installer (usually in the **Downloads** folder) and double-click to run it.

3. Select Installation Options:

Important: Check the box labeled "**Add Python to PATH**" to ensure Python is accessible from the command line.

Click on "**Customize Installation**" if you want to modify default options, or simply choose "**Install Now**" for the standard setup.

4. Customize Installation (Optional):

In the customization window, select optional features like **pip** (Python's package manager), **IDLE** (Python's development environment), and others.

Choose the destination folder if you don't want the default location.

1.2 Add python Directory

5. Wait for Installation to Complete:

The installer will copy files and set up Python on your system. This may take a few minutes.

6. Verify Installation:

Open a **Command Prompt** or **PowerShell** and type:

python --version or

python –V

7. Choose an editor you want to use like: vscode, pycharm, sublime, jupyter notebook, jupyter lab, atom.

8. Write your python codes with .py or .ipnb extension file.

1.2 Add python Directory

1.2.1.2 Linux/Unix (CentOS 7/Redhat/SUSE/UBUNTU)

1.2.1.3. Installation on Mac OS

Read lecture from page 16-17

1.2 Add python Directory

1.2.2 Set up path: https://youtu.be/91SGaK7_eeY?si=s0zpCN60m6jSgNbh

1.2.3.1 Find Python Installation Path

If you need to manually find the path where Python is installed, you can usually find it in the Scripts folder within the Python installation directory. The typical path is C:\Users\YourUsername\AppData\Local\Programs\Python\PythonXX\Scripts, where YourUsername is your Windows username, and XX represents the Python version.

1.2.3.1.1 Update PATH Environment Variable

For Windows 10:

Right-click on the Start menu and select "System."

Click on "Advanced system settings" on the left.

Click the "Environment Variables..." button.

In the "User variables" or "System variables" section, find and select the "Path" variable, then click "Edit..."

Click "New" and add the path to the "Scripts" folder within your Python installation directory. For example, C:\Users\YourUsername\AppData\Local\Programs\Python\PythonXX\Scripts.

Click "OK" to close each window.

1.2 Add python Directory

User Variables: Application is accessed only by that user

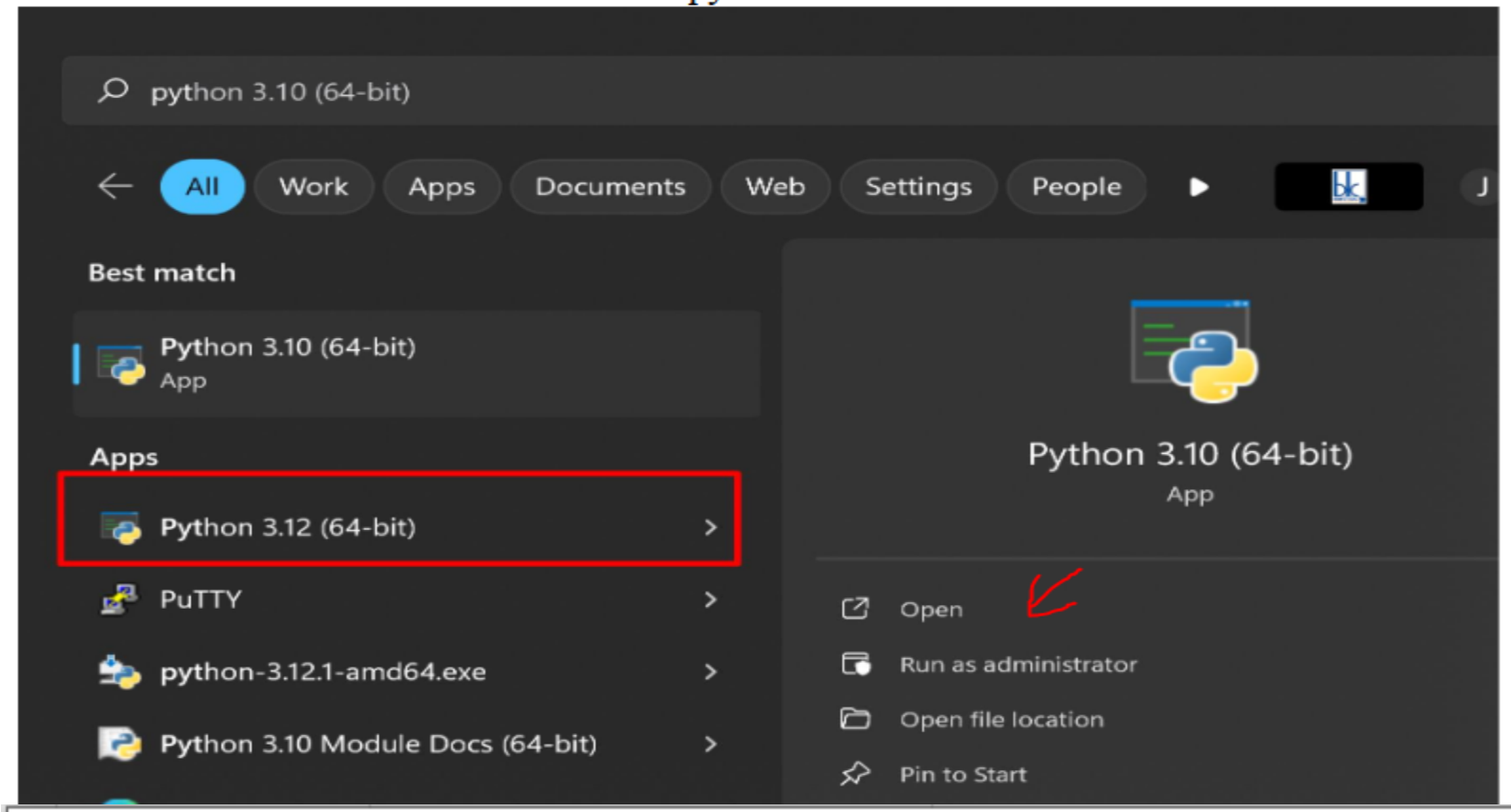
System Variables: Application is accessible that all users of that system

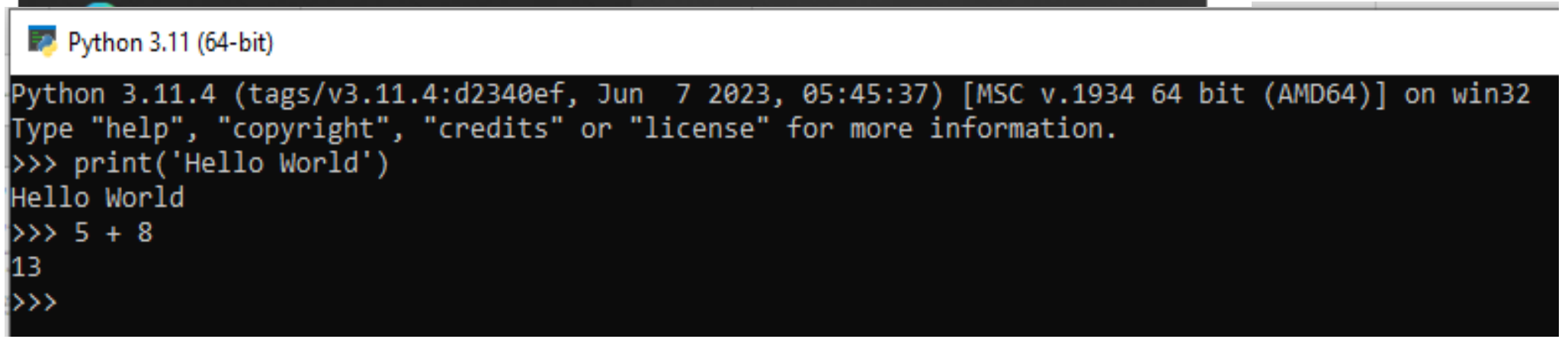
1.2.3.2 Running python

Interactive Interpreter You can start Python from Unix, DOS, or any other system that provides you a commandline interpreter or shell window. Enter python the command line

Windows

click on start menu and search for installed python APP



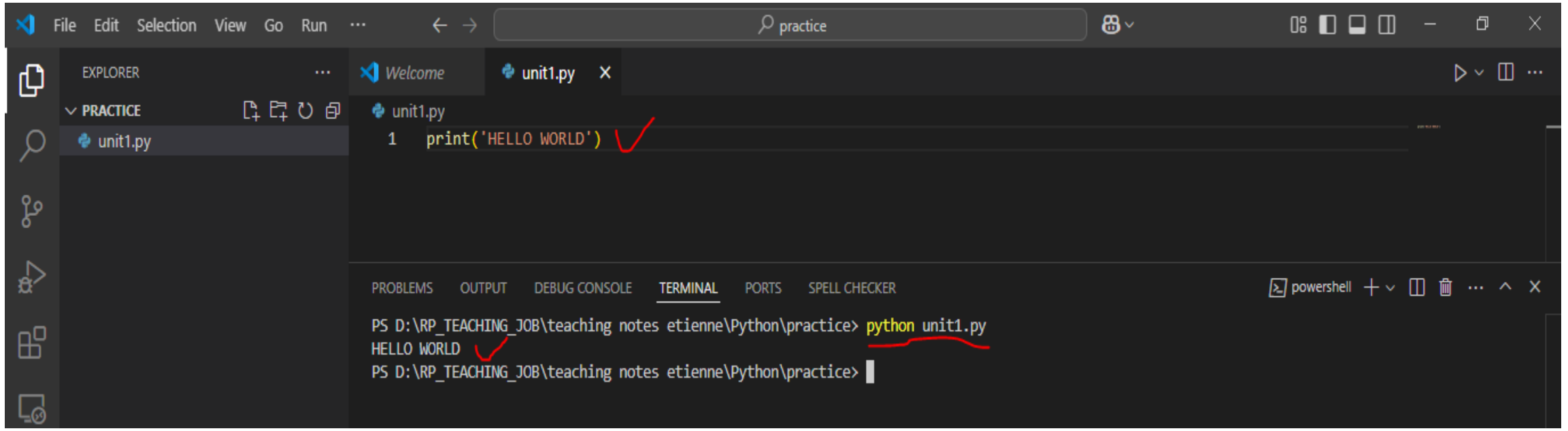


1.2 Add python Directory

1.2.3.3 Integrated Development Environment(IDE) or GUI

Integrated Development Environment You can run Python from a Graphical User Interface (GUI) environment as well, if you have a GUI application on your system that supports Python.

To run python file: python filename.py



QUIZ (PRACTICE) / 5 marks

https://forms.gle/J4nHat7wU3Cs528x7

15 minutes

1.3 Identify library

1.3.1 Definition of key terms

1.3.1.1 Library

A Python library is a collection of modules and functions that allow you to perform specific tasks or implement certain functionalities in your Python programs

Libraries are pre-written code that you can reuse in your own projects, saving you time and effort by not having to write everything from scratch.

Python has a rich ecosystem of libraries that cover a wide range of domains, such as data science, machine learning, web development, networking, graphics, and more. These libraries are typically distributed as packages and can be easily installed using tools like **pip**, which is the default package installer for Python.

1.3.1.1 Library

Ex: Compute area of a circle

```python
from math import pi
# Prompt the user to input the radius of the circle
r = float(input("Input the radius of the circle : "))
# Calculate the area of the circle using the formula: area = π * r^2
area = pi * r ** 2
# Display the result, including the radius and calculated area
print("The area of the circle with radius " + str(r) + " is: " + str(area))
```

✓ 4.1s

```
The area of the circle with radius 6.0 is: 113.09733552923255
```

```python
import math

def area_of_circle(radius):
    return math.pi * radius ** 2

# Example usage
radius = float(input("Enter the radius of the circle: "))
area = area_of_circle(radius)
print(f"The area of the circle is: {area}")
```

✓ 3.3s

```
The area of the circle is: 113.09733552923255
```

1.3.1.1 Library

Some popular Python libraries include:

NumPy: For numerical computing, providing support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays.

Pandas: Used for data manipulation and analysis. It provides data structures like DataFrame for efficiently handling and analyzing structured data.

Matplotlib and Seaborn: These libraries are used for data visualization, helping you create various types of plots and charts.

Requests: Simplifies the process of making HTTP requests, allowing you to interact with web services and APIs.

Django and Flask: These are web frameworks that simplify the process of building web applications in Python.

TensorFlow and PyTorch: Popular libraries for machine learning and deep learning, providing tools for building and training neural networks.

Beautiful Soup: A library for web scraping, allowing you to extract information from web pages.

1.3.1.2 Package

When developing a large application, you may end up with many different modules that are difficult to manage. In such a case, you'll benefit from grouping and organizing your modules. That's when packages come into play.

To be considered a package (or subpackage), a directory must contain a file named __init__.py. This file

usually includes the initialization code for the corresponding package.

1.3.1.3 Import

In Python, the "import" statement is used to bring in external modules or libraries into your program. This allows you to use the functions, classes, and variables defined in those modules within your own Python script or program.

1.3.1.3 Import

There are several ways to use the import statement:

1. Basic Import

```
# import module_name
import pandas
```

2. Import with alias

```
# import module_name as alias_name
import pandas as pd
```

3. Import Specific Items:

```
# from module_name import item_name
from math import pi
```

1.3.1.3 Import

There are several ways to use the import statement:

4. Import Everything:

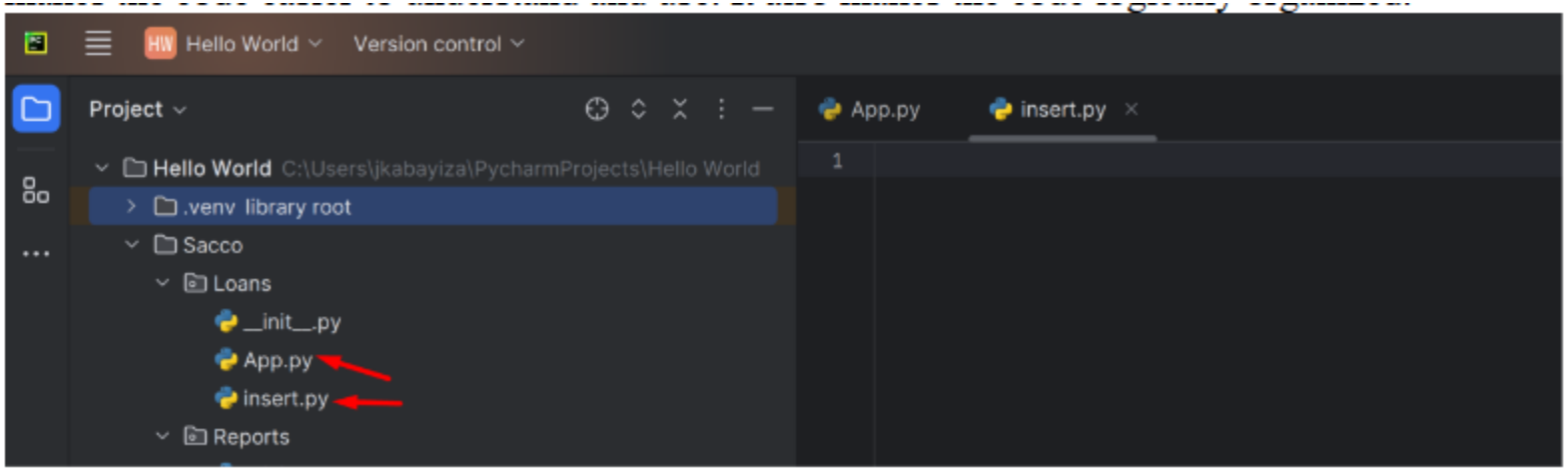
```
# from module_name import *

from math import *
```

1.3.1.4 Module

A Python module is a file containing Python definitions and statements. A module can define functions, classes, and variables. A module can also include runnable code. Grouping related code into a module

makes the code easier to understand and use. It also makes the code logically organized.

1.3.1.4 Module


Hello World
Version control
Project
App.py
insert.py
1
Hello World C:\Users\jkabayiza\PycharmProjects\Hello World
.venv library root
Sacco
Loans
__init__.py
App.py
insert.py
Reports

1.3.1.5 PIP

Pip is a package manager that is used to install and manage software packages and modules found in the Python Package index or any other indexes.

1.3.2 Identification of problem

HR added unwanted feature called **heart rate** during employee registration.

Please prepare your own employee dataset with those feature (Name, Age, Salary, **heart rate** payment status) add 20 random records from after dataset preparation keep it in data frame drop heart rate feature then store final dataset into csv file.

SOLUTION: In the statement import pandas as pd, the pd is an alias or shorthand name for the pandas library. It is commonly used in Python programming to save time when writing code and to improve readability.

```python
import pandas as pd
import random
# Prepare a dataset with the features: Name, Age, Salary, Heart Rate, Payment Status
data = {
    "Name": [f"Employee_{i}" for i in range(1, 21)],
    "Age": [random.randint(20, 60) for _ in range(20)],
    "Salary": [random.randint(30000, 100000) for _ in range(20)],
    "Heart Rate": [random.randint(60, 100) for _ in range(20)],
    "Payment Status": [random.choice(["Paid", "Unpaid"]) for _ in range(20)]
}

# Create DataFrame
df = pd.DataFrame(data)
df.head()
```

✓ 0.0s

	Name	Age	Salary	Heart Rate	Payment Status
0	Employee_1	44	80864	80	Unpaid
1	Employee_2	36	71214	60	Unpaid
2	Employee_3	56	30623	82	Paid
3	Employee_4	30	33858	63	Unpaid
4	Employee_5	35	40166	80	Paid

```python
# Drop the "Heart Rate" feature
df_final = df.drop(columns=["Heart Rate"])

# Store the final dataset into a CSV file
file_path = "employee_dataset.csv"
df_final.to_csv(file_path, index=False)

print(f"Dataset saved to {file_path}")

file = "employee_dataset.csv"

df = pd.read_csv(file)
df.head()
```

✓ 0.0s

```
Dataset saved to employee_dataset.csv
```

	Name	Age	Salary	Payment Status
0	Employee_1	44	80864	Unpaid
1	Employee_2	36	71214	Unpaid
2	Employee_3	56	30623	Paid
3	Employee_4	30	33858	Unpaid
4	Employee_5	35	40166	Paid

1.3.3 Getting a library

1.3.2.1 Install PIP

Check if it is installed: in cmd type=> **pip –version**

If PIP is already installed, you'll see its version information.

```
C:\Users\Etienne>pip --version
pip 23.3.2 from C:\Users\Etienne\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)
```

Step 2: Download the get-pip.py script (if PIP is not installed)

https://bootstrap.pypa.io/get-pip.py

Download the get-pip.py file and save it to a convenient location, such as your **Desktop**.

1.3.3 Getting a library

Step 3: Install PIP using the get-pip.py script

1. Open the Command Prompt.

2. Navigate to the directory where you downloaded the **get-pip.py** file. For example: **cd Desktop**

3. Run the following command to install PIP: **python get-pip.py**

Step 4: Verify the installation

Once the script finishes running, check if PIP was installed successfully:

pip –version

Step 5: Add PIP to the system PATH (if needed)

1.3.3 Getting a library

Step 5: Add PIP to the system PATH (if needed)

If the **pip** command is not recognized, you may need to add it to your system's **PATH**:

Locate the **Scripts** folder inside your Python installation directory (e.g., **C:\PythonXX\Scripts**).

Add the folder to your **PATH** environment variable:

Right-click This PC or My Computer → Properties → Advanced System Settings → Environment Variables.

Under System Variables, select Path → Edit → New, and **add the path to the Scripts folder.**

1.3.3 Getting a library

Step 6: Update PIP (optional but recommended)

Run the following command to update PIP to the latest version:

pip install --upgrade pip

1.3.2.2 Download and Install Library

To download a Python library, you typically use a package manager, and pip is the most common one for Python. To download and install a Python library using pip, you can use the following command in your terminal or command prompt:

pip install library_name

Example: Install Django Library**: pip install Django**

1.3.2.3 Import library

To import the **Django** library in Python, you use the import statement. Django is a web framework for building web applications using Python. If you have Django installed, you can import it into your Python

script or project like this:

Importing Django

import django

Now you can use Django functionality in your script

print(django.get_version())

Pandas

To import the pandas library in Python, you use the import statement. Pandas is a powerful library for data manipulation and analysis in Python. If you have Pandas installed, you can import it into your Python script or project like this:

```python
# Importing pandas
import pandas as pd

# Now you can use pandas functionality in your script
data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
df = pd.DataFrame(data)

# Display the DataFrame
print(df)
```

[7] ✓ 0.0s

```
    Name  Age
0  Alice   25
1    Bob   30
```

This script creates a simple DataFrame using pandas and prints it to the console. Ensure that pandas is installed in your Python environment; otherwise, you'll need to install it using:

bash Copy Edit

```
pip install pandas
```

Additional

Do your own research on how to set python **virtual environment** and try to do some practices into your created environment.

Reference: https://www.youtube.com/watch?v=XKS1mulk7AQ

Quiz 5 marks

https://forms.gle/odiiqMwTTTebTnQ57

5 minutes

Q &A

Thanks

Unit ends