CLASS – 2

Functions and Modules

Avnit Bambah

Python for Everybody

FUNCTIONS

- We implement code that we want to use over and over again at different places.
- This can make the over all code very large.
- We can use functions
- Defining a function
  - Def hello():
    - Print(“Hello”)

PARAMETERS

- Def print_sum(number1,number2):
  - Print(number1 + number2)
- Return value
  - The keyword return is used to return the function result back as a variable
  - Def add(num1,num2):
    - Return num1+num2

Default parameter

```
Def say(text="Hello")
        print(text)
```

SCOPE

- Scope is not only important for functions but also important for loops.
- Local and global variable

program show local and global variable

STRING FUNCTIONS

- Name,age = “john”,25
- Print(“My name is {} and I am {} years old”.format(name,age))
- Placeholders

Placeholders	DataType
%c	Character
%s	String
%d or %i	Integer
%f	float

CASE MANIPULATING FUNCTIONS

Function	Description
String.lower()	Converts all letters to lowercase
String.upper()	Converts all letters to uppercase
String.title()	Converts all letters to title case
String.capitalize()	Converts first letter to upper case
String.swapcase()	Swaps the case of all letters

COUNT, FIND , REPLACE AND SPLIT FUNCTION

- Counts – counts how many times a specific string occurs in another string
- Find – the first occurrence of a certain string in another string
- Join – With the join function we can join a sequence to a string and separate each element by this particular string
- Replace – One string within a text by another one.
- Split – we want to split specific parts of a string and put them into a list.

MODULES NUMPY

What is NumPy?

- NumPy is a Python library used for working with arrays.
- It also has functions for working in domain of linear algebra, fourier transform, and matrices.
- NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
- NumPy stands for Numerical Python.

Why Use NumPy?

- In Python we have lists that serve the purpose of arrays, but they are slow to process.
- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
- The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
- Arrays are very frequently used in data science, where speed and resources are very important.

NUMPY INSTALL

- Pip install numpy
- Import numpy

NumPy has some extra data types, and refer to data types with one character, like `i` for integers, `u` for unsigned integers etc.

Below is a list of all data types in NumPy and the characters used to represent them.

- `i` - integer – (Python)
- `b` - Boolean – (Python)
- `u` - unsigned integer
- `f` - float – (Python)
- `c` - complex float
- `m` - timedelta
- `M` - datetime
- `O` - object
- `S` - string – (python)
- `U` - unicode string
- `V` - fixed chunk of memory for other type (void)

MODULE PANDAS

What is Pandas?

- Pandas is a Python library used for working with data sets.
- It has functions for analyzing, cleaning, exploring, and manipulating data.
- The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

Why Use Pandas?

- Pandas allows us to analyze big data and make conclusions based on statistical theories.
- Pandas can clean messy data sets, and make them readable and relevant.
- Relevant data is very important in data science

GETTING STARTED PANDAS

- Pip install pandas
- Import pandas

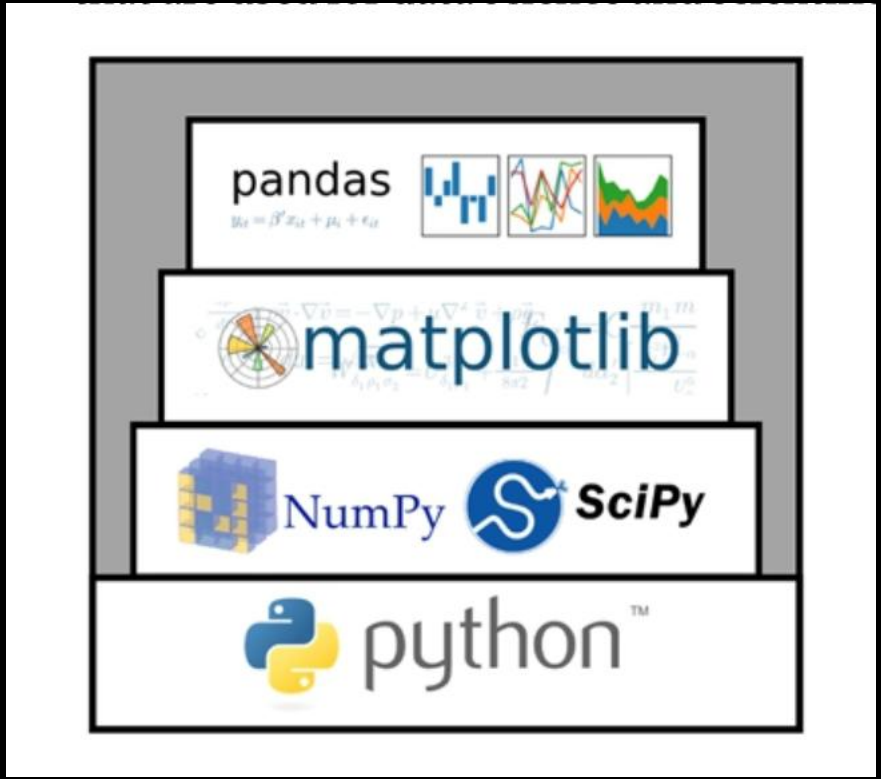
reading file with pandas

- `import pandas as pd`

```
df = pd.read_csv('data.csv')

print(df.to_string())
```

INSTALLING MODULES



TYPES OF PYTHON ENVIRONMENTS

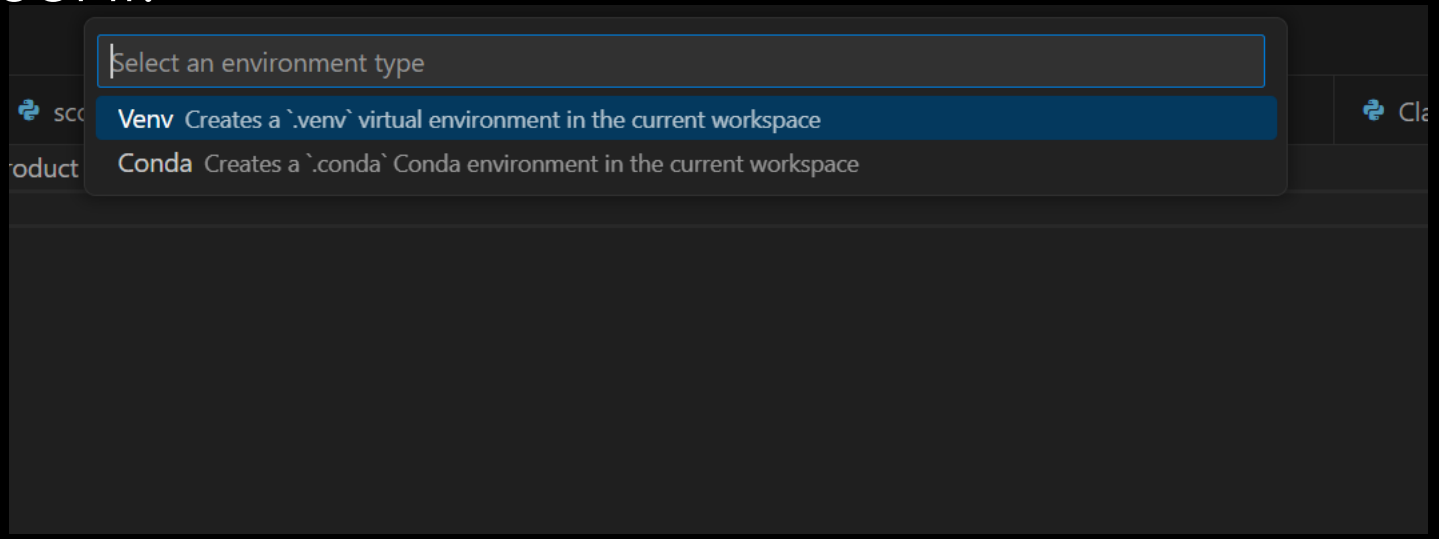
- Global Environment
- Local Environment
  - Virtual Environment (venv)
  - Conda Environment (https://conda.io/projects/conda/en/latest/user-guide/getting-started.html)

PYTHON ENVIRONMENT TOOLS

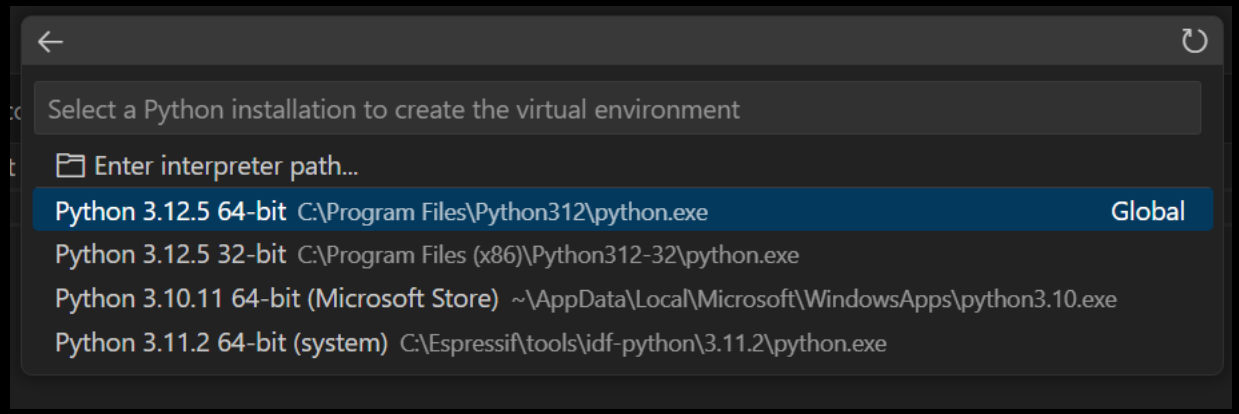
- Pip (install python3-pip)
- Venv (install python3-venv)
- Conda (installed with miniconda)

CREATING ENVIRONMENTS

- Ctrl+shift+P search for the python : Create Environment command and select it.



COMMANDS TO RUN



FUNCTIONS BASICS

- Call expressions - myfunc('hack', tool=python, *versions)
- def - def printer(message): print('Hello', message)
- Return - def adder(a, b=1, *c): **return** a + b + c[0]
- Lambda - funcs = [lambda x: x**2, lambda x: x**3]
- Global - x = 'old' def changer(): global x; x = 'new'
- Nonlocal - def outer(): x = 'old' def changer(): **nonlocal** x; x= ‘new’
- Yield - def square(x): for I in range(x): **yield** i**2

PYTHON: A GREAT CHOICE FOR WEB DEVELOPMENT

- **Readability:** Python's clear syntax makes it easier to learn and understand.
- **Large Community:** Extensive libraries, frameworks, and resources available.
- **Versatility:** Python is used for web development, data science, machine learning, scripting, and more.

PYTHON WEB FRAMEWORKS: THE BUILDING BLOCKS

The model likely chose one of these popular Python web frameworks:

- **Flask:** (our focus) A lightweight, microframework. Good for smaller, simpler web applications. Easy to get started with.
- **Django:** A full-featured framework. Suitable for larger, more complex projects with features like databases, user authentication, etc.
- **FastAPI:** A modern framework focused on high performance and building APIs (which can be used to serve web pages).
- **Streamlit/Dash** Used to easily display data, charts, and interactive elements.

CODE EXPLANATION

- **from flask import Flask:** Imports the Flask library.
- **app = Flask(__name__):** Creates a Flask application instance.
- **@app.route("/"):** This is a *decorator*. It tells Flask that the hello_world function should be executed when someone visits the root URL (/) of the website.
- **def hello_world():**: Defines a function that returns the HTML content to be displayed.
- **return " Hello, World! ":** Returns a simple HTML paragraph.
- **if __name__ == "__main__":**: This ensures the code runs only when
- the script is executed directly (not when imported as a module).
- **app.run(debug=True):** Starts the Flask development server. debug=True
- enables helpful debugging features.