

Class #08: Modules and Packages

1. Understanding the difference between modules and packages.
2. Creating and using modules.
3. Working with built-in modules (e.g., datetime).
4. Creating and using packages.
5. Using pip and PyPI. Assignment #8:

1. Understanding the difference between modules and packages.

পাইথনে মডিউল এবং প্যাকেজ এর মধ্যে মূল পার্থক্য বোঝানোর জন্য আমরা বাস্তব জীবনের কিছু উদাহরণ ব্যবহার করতে পারি।

১. মডিউল (Module)

ধরা যাক, আপনি একটি বইয়ের দোকানে গিয়েছেন। সেই দোকানে একটি বই আছে, যেটি কেবল একটি নির্দিষ্ট বিষয়ের উপর লেখা। যেমন, যদি সেই বইটি "গণিতের গাণিতিক সমীকরণ" সম্পর্কিত হয়, তাহলে এই বইটি এককভাবে একটি নির্দিষ্ট বিষয়ের তথ্য দেয়।

এখনে, বইটি মডিউল এর মতো কাজ করছে, কারণ এটি একটি নির্দিষ্ট বিষয় বা কাজকে কভার করে এবং আপনি এই বইটির মধ্যে থাকা তথ্য ব্যবহার করতে পারেন।

২. প্যাকেজ (Package)

এখন ধরুন, সেই বইয়ের দোকানে শুধু গণিতের গাণিতিক সমীকরণ সম্পর্কিত একটি বইই নেই, বরং গণিতের বিভিন্ন বিষয় (যেমন - আলজেব্রা, ক্যালকুলাস, জ্যামিতি) নিয়ে অনেক বই রয়েছে। দোকানে প্রতিটি বিষয় সম্পর্কিত আলাদা আলাদা বইয়ের বিভাগ রয়েছে, এবং প্রতিটি বিভাগের মধ্যে অনেক বই রয়েছে।

এখনে, বইয়ের দোকানটি একটি প্যাকেজ এর মতো, যেখানে বিভিন্ন বিষয়ের বইয়ের বিভাগ রয়েছে, আর প্রতিটি বিভাগের বই হচ্ছে মডিউল। এই প্যাকেজের মধ্যে একাধিক মডিউল থাকতে পারে, যেগুলি একত্রে একটি পূর্ণ ধারণা দেয়।

পার্থক্য:

1. মডিউল হলো একটি একক বই, যা একটি নির্দিষ্ট বিষয় বা কাজের উপর আলোকপাত করে।
2. প্যাকেজ হলো একটি বইয়ের দোকান, যেখানে একাধিক বইয়ের (মডিউল) বিভাগ রয়েছে এবং সেই বিভাগগুলির মধ্যে বিভিন্ন বই রয়েছে।

অর্থাৎ, মডিউল হলো একক ফাইল, আর প্যাকেজ হলো একাধিক মডিউল (ফাইল) নিয়ে গঠিত একটি গোষ্ঠী।

নিচে মডিউল এবং প্যাকেজ এর মধ্যে পার্থক্য টেবিল আকারে দেওয়া হল:

| বিষয় | মডিউল | প্যাকেজ |
|--------------|---|---|
| সংজ্ঞা | একটি একক পাইথন ফাইল যা কোড ধারণ করে। | একাধিক মডিউল নিয়ে গঠিত একটি ডিরেক্টরি (ফোল্ডার)। |
| ফাইল format | .py ফাইল | একটি ডিরেক্টরি/ফোল্ডার যা __init__.py ফাইল ধারণ করে। |
| ব্যবহার | কোডের পুনঃব্যবহারযোগ্য অংশ (ফাংশন, ক্লাস, ভ্যারিয়েবল ইত্যাদি) সংরক্ষণ। | সম্পর্কিত মডিউলগুলিকে একত্রে সংগঠিত করা। |
| ইম্পোর্ট ধরন | import module_name | from package_name import module_name |
| সংগঠন | একক .py ফাইল। | একাধিক .py ফাইলের সমন্বয়ে গঠিত ডিরেক্টরি/ফোল্ডার। |
| ফাংশন | একক ফাংশন বা ক্লাসের কার্যকারিতা সরবরাহ করে। | একাধিক মডিউল নিয়ে বড় ফিচার বা কার্যকারিতা প্রদান করে। |

module হলো, কোনো python ফাইল। আর অনেকগুলো python ফাইল যদি কোনো ফোল্ডারের মধ্যে রাখা হয় তাহলে সেটা package.

2. Creating and using modules.

পাইথন প্রজেক্ট এর মধ্যে ziaur_rahman.py এবং mujibur_rahman.py নামক দুইটা python file অর্থাৎ দুইটা module রয়েছে। কারণ, আমরা জানি python file এর অপরানাম module.

```
ziaur_rahman.py

# Function with return
def add_num(x, y):
    return x + y

# Function with no return
def greeting_wish(name):
    print(f"Hello {name}")

# Dictionary containing country details
country = {"name": "Bangladesh", "language": "Bangla", "age": 25}
```

```

# a Variable
indipendente_date = 26

mujibur_rahman.py

# Import the ziaur_rahman module
import ziaur_rahman

# Calling the add_num function but nothing is printed because it is not wrapped with print()
ziaur_rahman.add_num(10, 30)
# Calling the add_num function and printing the result
print(ziaur_rahman.add_num(10, 20))

# Calling the greeting_wish function to print a greeting message
ziaur_rahman.greeting_wish("Bangladesh")

# Accessing the 'age' from the 'country' dictionary and printing it
print(ziaur_rahman.country['age'])

# Adding 53 to the 'indipendente_date' variable and printing the result
print(ziaur_rahman.indipendente_date + 53)

```

ziaur_rahman.py

```

# Function with return
def add_num(x, y):
    return x + y

# Function with no return
def greeting_wish(name):
    print(f"Hello {name}")

# Dictionary containing country details
country = {"name": "Bangladeshi", "language": "Bangla", "age": 25}

# a Variable
indipendente_date = 26

```

এই লিংক module শিখার এর জন্য ভালোঃ

https://www.w3schools.com/python/python_modules.asp

mujibur_rahman.py

```

# Import the ziaur_rahman module
import ziaur_rahman

# Calling the add_num function but nothing is printed because it is not wrapped with print()
ziaur_rahman.add_num(x: 10, y: 30)
# Calling the add_num function and printing the result
print(ziaur_rahman.add_num(x: 10, y: 20))

# Calling the greeting_wish function to print a greeting message
ziaur_rahman.greeting_wish("Bangladesh")

# Accessing the 'age' from the 'country' dictionary and printing it
print(ziaur_rahman.country['age'])

# Adding 53 to the 'indipendente_date' variable and printing the result
print(ziaur_rahman.indipendente_date + 53)

```

Run mujibur_rahman

"C:\Users\Minhazul Kabir\AppData\Local\Programs\Python\Python313\python.exe" D:\PythonProject\MinhazulKabir\mujibur_rahman.py

30
Hello Bangladesh
25
79

যখন আমরা একটি function কে ব্যবহার করি, তখন তাকে function call বলা হয়। function call এর মাধ্যমে function এর কার্যকরিতা ব্যবহার করা হয়। একইভাবে, একটি function যদি অন্য কোনো page/file এ ব্যবহার করা হয়, তখন তাকে module use বলা হয়।

Alias ব্যবহার:

Alias ব্যবহার বলতে, একটি মডিউল বা প্যাকেজকে একটি ছোট নাম দিয়ে সংক্ষিপ্ত করা, যাতে কোড লেখার সময় সহজতা থাকে এবং বার বার বড় নামটি লিখতে না হয়।

পাইথনে যখন আপনি একটি মডিউল ইম্পোর্ট করেন, তখন আপনি চাইলে তার পূর্ণ নাম ব্যবহার করতে পারেন অথবা একে একটি ছোট নাম (alias) দিয়ে সংক্ষেপে ব্যবহার করতে পারেন।

উদাহরণ:

ধরা যাক, আপনি ziaur_rahman নামক মডিউলটি ইম্পোর্ট করেছেন। আপনি যদি একে সংক্ষিপ্ত নাম দিয়ে ব্যবহার করতে চান, তবে আপনি as কিওয়ার্ড ব্যবহার করে মডিউলটির জন্য একটি ছোট নাম দিতে পারেন। সেটিকে alias বলা হয়।

Alias ব্যবহার:

```
import ziaur_rahman as z
```

এখানে:

- ziaur_rahman মডিউলটির জন্য z নামে একটি alias তৈরি করা হয়েছে।
- এর ফলে, পরবর্তীতে ziaur_rahman লিখতে হবে না, বরং z লিখে কাজ করা যাবে।

Alias ছাড়া:

```
import ziaur_rahman
```

এখানে, পুরো মডিউল নাম (ziaur_rahman) ব্যবহার করতে হবে কোডে প্রতিবার।

উপকারিতা:

- **কোড সংক্ষিপ্ত করা:** বড় মডিউল নাম বার বার লিখতে হয় না।
- **পাঠ্যোগ্যতা বৃদ্ধি:** ছোট নাম ব্যবহার করলে কোড আরও পরিষ্কার ও সহজ হয়ে ওঠে।

```
mujibur_rahman.py
# Import the ziaur_rahman module and give it an alias 'z'
import ziaur_rahman as z

# Calling the add_num function but nothing is printed because it
# is not wrapped with print()
z.add_num(10, 30)
# Calling the add_num function and printing the result
print(z.add_num(10, 20))

# Calling the greeting_wish function to print a greeting message
z.greeting_wish("Bangladesh")

# Accessing the 'age' from the 'country' dictionary and printing
# it
print(z.country['age'])

# Adding 53 to the 'independente_date' variable and printing the
# result
print(z.independente_date + 53)
```

পার্থক্য:

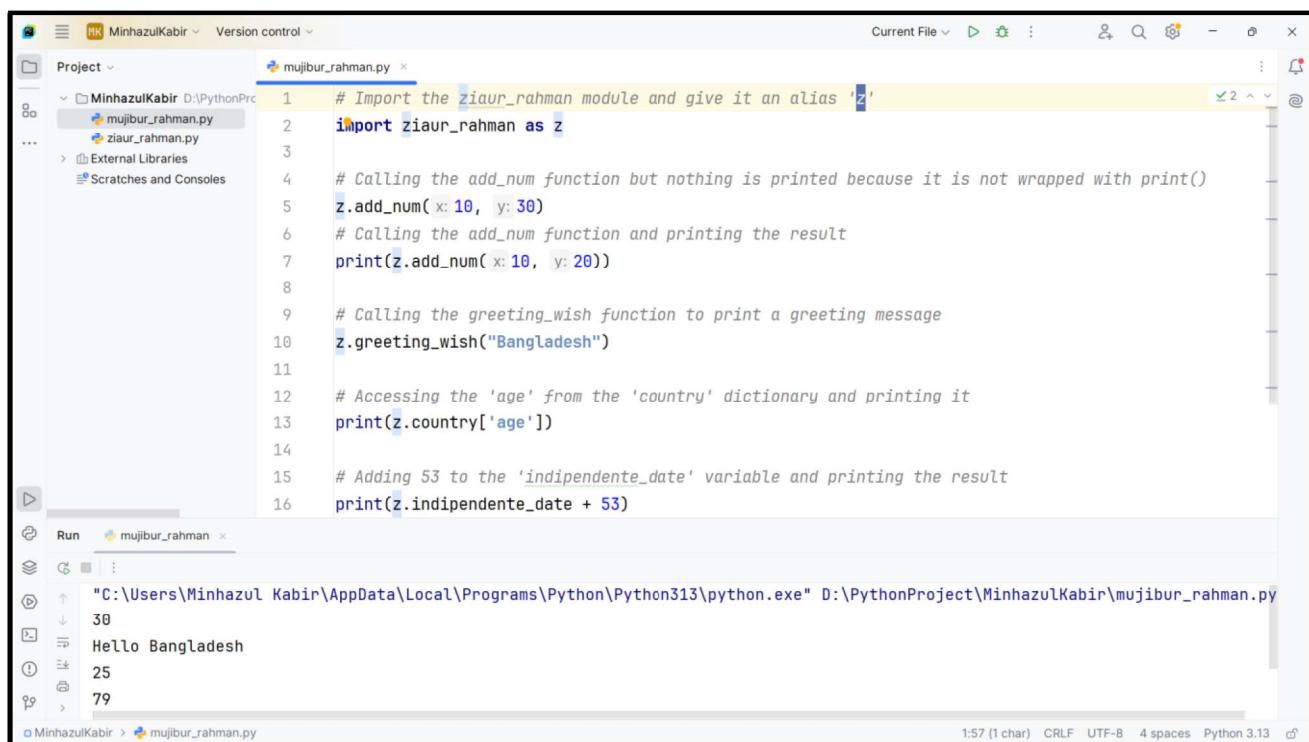
| বিষয় | প্রথম কোড (Alias ব্যবহার করা) | দ্বিতীয় কোড (Alias ছাড়া) |
|----------------------|---|---|
| Module Import | import ziaur_rahman as z - ziaur_rahman মডিউলকে z নামে ছোট নামে ইম্পোর্ট করা হয়। | import ziaur_rahman - মডিউলটি সম্পূর্ণ নাম দিয়ে ইম্পোর্ট করা হয়। |

| | | |
|-----------------------------|--|---|
| Module Usage | মডিউল ফাংশন এবং ভ্যারিয়েবল ব্যবহার করতে <code>z</code> ব্যবহার করতে হয়। যেমন: <code>z.add_num(10, 20)</code> | মডিউল ফাংশন এবং ভ্যারিয়েবল ব্যবহার করতে সম্পূর্ণ মডিউল নাম ব্যবহার করতে হয়। যেমন: <code>ziaur_rahman.add_num(10, 20)</code> |
| সংক্ষেপে কোড ব্যবহার | ছোট নাম (alias) ব্যবহার করে কোডটি সংক্ষিপ্ত ও সহজ হয়। | মডিউল নাম পুরোপুরি ব্যবহার করতে হয়, যা কোডটি কিছুটা দীর্ঘ হতে পারে। |

সারণী থেকে যে পার্থক্যগুলো দেখা যাচ্ছে:

- Alias ব্যবহার করলে মডিউল নাম ছোট হয়ে যায়, ফলে কোড লেখার সময় সহজতা থাকে।
- Alias ছাড়া পুরো মডিউল নাম ব্যবহার করতে হয়, যা কখনও কখনও কোড দীর্ঘ করতে পারে, তবে এতে কিছুটা স্পষ্টতা পাওয়া যায়।

`mujibur_rahman.py`



```

# Import the ziaur_rahman module and give it an alias 'z'
import ziaur_rahman as z

# Calling the add_num function but nothing is printed because it is not wrapped with print()
z.add_num(x: 10, y: 30)
# Calling the add_num function and printing the result
print(z.add_num(x: 10, y: 20))

# Calling the greeting_wish function to print a greeting message
z.greeting_wish("Bangladesh")

# Accessing the 'age' from the 'country' dictionary and printing it
print(z.country['age'])

# Adding 53 to the 'independente_date' variable and printing the result
print(z.independente_date + 53)

```

The run output shows:

```

C:\Users\Minhazul Kabir\AppData\Local\Programs\Python\Python313\python.exe D:\PythonProject\MinhazulKabir\mujibur_rahman.py
30
Hello Bangladesh
25
79

```

3. Working with built-in modules (e.g., datetime).

পাইথনে built-in modules হলো এমন মডিউলগুলো যেগুলি পাইথনের সাথে পূর্বনির্ধারিতভাবে আসে এবং আপনাকে আলাদাভাবে ইনস্টল করতে হয় না। এগুলি Python এর স্ট্যান্ডার্ড লাইব্রেরির অংশ এবং বিভিন্ন ধরনের কাজের জন্য ব্যবহৃত হয়।

এখানে কিছু সাধারণ built-in modules এবং তাদের কাজের বর্ণনা দেওয়া হলো:

1. math

কাজ: গণনা সম্পর্কিত কার্যাবলি যেমন গাণিতিক ফাংশন (যোগ, বিয়োগ, গুণ, ভাগ, বর্গমূল, কসমস, সাইন, কসমাইন ইত্যাদি) প্রদান করে।

উদাহরণ:

```
import math

print(math.sqrt(16)) # Output: 4.0

print(math.pi)      # Output: 3.141592653589793
```

2. os

কাজ: অপারেটিং সিস্টেমের সাথে সম্পর্কিত কার্যাবলী (ফাইল, ডিরেক্টরি ম্যানিপুলেশন, পরিবেশ ভেরিয়েবল ইত্যাদি) পরিচালনা করতে ব্যবহৃত হয়।

উদাহরণ:

```
import os

print(os.getcwd()) # Output: current working directory

os.mkdir("new_folder") # Creates a new folder
```

3. sys

কাজ: পাইথন ইন্টারপ্রেটার এবং এর পরিবেশের সাথে সম্পর্কিত তথ্য দেয় এবং কমান্ড লাইন আর্গুমেন্টগুলি (যেমন স্ক্রিপ্টে ইনপুট) ম্যানেজ করতে সহায়তা করে।

উদাহরণ:

```
import sys

print(sys.version) # Output: Python version

print(sys.argv)   # Command line arguments passed to script
```

4. datetime

কাজ: তারিখ এবং সময় সম্পর্কিত কাজ (তারিখ এবং সময়ের মান নির্ধারণ, সময়ের গাণিতিক হিসাব ইত্যাদি) করার জন্য ব্যবহৃত হয়।

উদাহরণ:

```
import datetime
```

```
print(datetime.datetime.now()) # Output: Current date and time
```

5. random

কাজ: এলোমেলো সংখ্যা এবং অন্যান্য এলোমেলো তথ্য তৈরি করার জন্য ব্যবহৃত হয়, যেমন এলোমেলো সংখ্যার তালিকা, এলোমেলোভাবে একটি আইটেম নির্বাচন ইত্যাদি।

উদাহরণ:

```
import random
print(random.randint(1, 100)) # Output: Random integer between 1 and 100
```

6. time

কাজ: সময় সম্পর্কিত কার্যাবলি, যেমন টাইমিং, বিলম্ব (sleep) ইত্যাদি পরিচালনা করার জন্য ব্যবহৃত হয়।

উদাহরণ:

```
import time
time.sleep(2) # Pauses the program for 2 seconds
print("Hello after 2 seconds")
```

7. json

কাজ: JSON (JavaScript Object Notation) ডেটা পদ্ধতিতে ডেটা পার্স এবং সিরিয়ালাইজ করার জন্য ব্যবহৃত হয়।

উদাহরণ:

```
import json
data = {'name': 'John', 'age': 30}
json_str = json.dumps(data) # Converts Python dictionary to JSON string
print(json_str)
```

8. re

কাজ: রেগুলার এক্সপ্রেশন (regex) সম্পর্কিত কার্যাবলি যেমন প্যাটার্ন ম্যাটিং, টেক্সট অনুসন্ধান ইত্যাদি।

উদাহরণ:

```
import re
```

```
pattern = r"\d+" # Matches one or more digits
result = re.findall(pattern, "There are 123 apples and 456 oranges")
print(result) # Output: ['123', '456']
```

9. collections

কাজ: আরও উন্নত ডেটা স্ট্রাকচার যেমন Counter, deque, defaultdict, namedtuple ইত্যাদি প্রদান করে।

উদাহরণ:

```
import collections

counter = collections.Counter(["apple", "banana", "apple"])
print(counter) # Output: Counter({'apple': 2, 'banana': 1})
```

10. itertools

কাজ: ইটারেটর এবং কমিনেশন সম্পর্কিত ফাংশন প্রদান করে, যেমন permutations, কমিনেশন, ক্যাটেনেশন ইত্যাদি।

উদাহরণ:

```
import itertools

result = itertools.permutations([1, 2, 3])
for perm in result:
    print(perm) # Output: (1, 2, 3), (1, 3, 2), (2, 1, 3), etc.
```

11. fileinput

কাজ: এক বা একাধিক ফাইল থেকে ইনপুট নেওয়ার জন্য ব্যবহৃত হয়।

উদাহরণ:

```
import fileinput

for line in fileinput.input(files=('file1.txt', 'file2.txt')):
    print(line)
```

12. socket

কাজ: নেটওয়ার্কিং সম্পর্কিত কার্যাবলী যেমন সার্ভার এবং ক্লায়েন্ট অ্যাপ্লিকেশন তৈরি, যোগাযোগ স্থাপন ইত্যাদি।

উদাহরণ:

```
import socket
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("example.com", 80))
```

13. hashlib

কাজ: হ্যাশ ফাংশন যেমন MD5, SHA-1, SHA-256 ইত্যাদি ব্যবহার করে ডেটা হ্যাশ করা।

উদাহরণ:

```
import hashlib
hash_object = hashlib.sha256(b'hello')
print(hash_object.hexdigest()) # Output:
a591a6d40bf420404a011733cfb7b190d62c65bf0bcda1d2d90bc36b64a62e1d
```

Python এর **built-in modules** আপনার কোডকে আরও কার্যকরী এবং দ্রুত করতে সাহায্য করে, কারণ আপনাকে বাইরের লাইব্রেরি ইনস্টল করার দরকার পড়ে না।

Date module: https://www.w3schools.com/python/python_datetime.asp

Math module: https://www.w3schools.com/python/python_math.asp

JSON module: https://www.w3schools.com/python/python_json.asp

Regular Expression module: https://www.w3schools.com/python/python_regex.asp

যেকোনো Library বা machine learning model (algorithm) ব্যবহার করতে হলে, আপনাকে import দিয়ে তাদের module ব্যবহার করতে হয়। এর মাধ্যমে আপনি সেই Library বা model এর ফিচার এবং ফাংশনগুলো আপনার কোডে ব্যবহার করতে পারেন।

Import করার মাধ্যমে আপনি বিভিন্ন algorithm প্রয়োগ করতে, ডেটা প্রসেসিং করতে বা pre-built ফাংশন ব্যবহার করতে সক্ষম হন।

4. Creating and using packages.

package হচ্ছে folder যার মধ্যে python ফাইল থাকবে। এখন আমরা package অর্থাৎ python ফাইল folder এর মধ্যে রাখবো। dhaka নামে একটা নতুন ফোল্ডার/directory বানালাম। তার মধ্যে ‘mirpur’ ও ‘cantonment’ নামে দুইটা python file বানালাম। সে python ফাইল এ কোড লিখলাম।

```
dhaka/mirpur.py
```

```
def your_name(name):
    print(f"Hi, {name}! Come on to Mirpur.")
```

```
dhaka/cantonment.py
```

```
def loop(n):
    #Print numbers from 0 to n-1.
    for i in range(n):
        print(i)
```

এখন আমরা যদি এই দুইটা python file ব্যবহার করতে চাই, অন্য python file এ তখন আমাদের

```
from package_name import module_name
```

অর্থাৎ from ফোল্ডারের_নাম import পাইথন_ফাইলের_নাম

আমাদের সেখানে, folder এর নাম dhaka এবং file এর নাম ‘mirpur’ ও ‘cantonment’ তাই ।

```
from dhaka import cantonment, mirpur
```

এখন, নিচের প্রোগ্রাম চালু করি ।

```
from dhaka import cantonment, mirpur
mirpur.your_name("minhaz")
cantonment.loop(10)
```

```
mujibur_rahman.py
```

```
1 from dhaka import cantonment, mirpur
2 mirpur.your_name("minhaz")
3 cantonment.loop(10)

Run mujibur_rahman
C:\Users\Minhazul Kabir\AppData\Local\Programs\Python\Python313\python.exe D:\PythonProject\MinhazulKabir\mujibur_rahman.py
Hi, minhaz! Come on to Mirpur.
0
1
2
3
4
5
6
7
8
9
```

5. Using pip and PyPI.

pip এবং PyPI হল Python-এর প্যাকেজ ব্যবস্থাপনা সিস্টেমের দুটি গুরুত্বপূর্ণ উপাদান।

- pip (pip install package):** pip একটি টুল (যন্ত্র), যা Python ব্যবহারকারীদের বাইরের প্রোগ্রাম বা লাইব্রেরি ইনস্টল করতে সাহায্য করে। ধরুন, আপনি কোন বিশেষ কাজ করার জন্য একটি কোড লাইব্রেরি (যেমন ডেটা বিশ্লেষণ বা ওয়েব ডেভেলপমেন্টের জন্য) ব্যবহার করতে চান, তাহলে pip ব্যবহার করে আপনি সেই লাইব্রেরি আপনার কম্পিউটারে ইনস্টল করতে পারেন।
- PyPI (Python Package Index):** PyPI হল একটি অনলাইন রেজিস্ট্রি বা সংগ্রহস্থল, যেখানে Python ভাষায় লেখা হাজার হাজার লাইব্রেরি বা প্যাকেজ রাখা থাকে। এটি অনেকটা একটি অনলাইন দোকানের মতো, যেখানে আপনি আপনার প্রয়োজনীয় লাইব্রেরি খুঁজে পাবেন।

সংক্ষেপে, pip হল একটি টুল যা আপনাকে PyPI থেকে লাইব্রেরি ইনস্টল করতে সাহায্য করে।

এটা দেখোঃ https://www.w3schools.com/python/python_pip.asp

PyPI (Python Package Index) হল Python-এর জন্য একটি বিশাল লাইব্রেরি, যেখানে হাজার হাজার প্যাকেজ বা লাইব্রেরি রয়েছে, যা আমরা আমাদের Python প্রজেক্টে ব্যবহার করতে পারি। যেমন, Play Store-এ Android-এর বিভিন্ন অ্যাপ্লিকেশন পাওয়া যায়, তেমনি PyPI-এ Python-এর বিভিন্ন প্যাকেজ পাওয়া যায়, যা আমাদের প্রয়োজন অনুসারে ইনস্টল করে ব্যবহার করা যায়।

Android-এ app ইনস্টল করার জন্য আমরা play store libraryটা app ডাউনলোড ও ইনস্টল করার কাজ করে। Python-এ প্যাকেজ ইনস্টল করার জন্য আমরা pip নামক একটি টুল ব্যবহার করি, যা PyPI library থেকে package ডাউনলোড ও ইনস্টল করার কাজ করে।

গঠন:

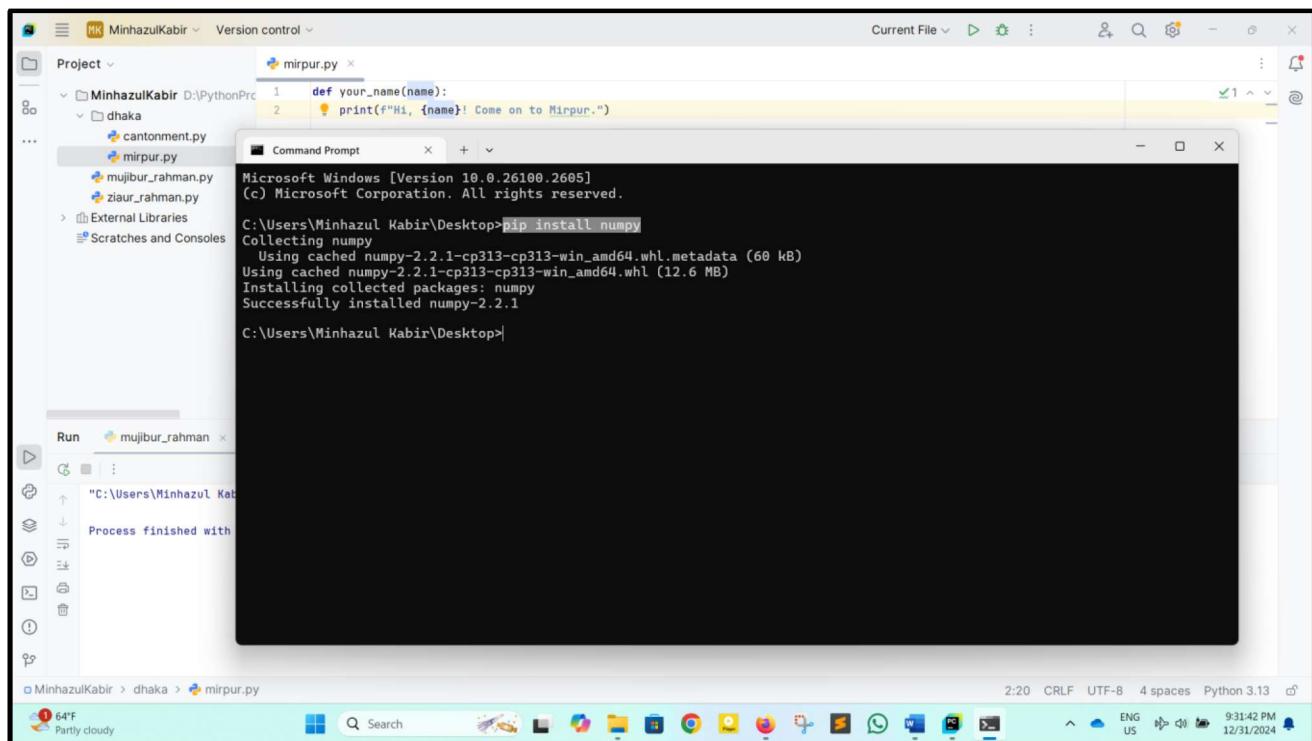
যখন আপনি PyPI থেকে একটি প্যাকেজ ইনস্টল করতে চান, তখন কমান্ড হবে: `pip install package_name`

উদাহরণস্বরূপ, যদি আপনি numpy প্যাকেজটি ইনস্টল করতে চান, যা গণনা ও ডেটা ম্যানিপুলেশনের জন্য ব্যবহৃত হয়, তাহলে Windows কম্পিউটার এর Command Line-এ গিয়ে আপনাকে নিচের কমান্ডটি লিখতে হবে:

```
pip install numpy
```

এটি numpy প্যাকেজটি আপনার কম্পিউটারে ডাউনলোড এবং ইনস্টল করবে। এরপর আপনার Python প্রোগ্রামে numpy ব্যবহার করতে পারবেন।

এখানে, দেখানো হলো কিভাবে pip install numpy ব্যবহার করতে হবে।



নিচের প্রোগামের output কি ?

```
import numpy as np
n = [1,2,3,4,5,6]
z=n
print(z*2)
z = np.array(n)
print(z*2)
```

?????