Python Modules: Notes in Hinglish

1. What is a Python Module?

A **module** in Python is just a file containing Python code — like functions, variables, or classes. It's used to **organize** and **reuse** code easily in different programs.

Why Use Modules?
- To avoid writing the same code again and again.
- To organize code into different files (to keep it clean and simple).
- To reuse code in multiple projects.

---

2. Types of Python Modules

1. Built-in Modules
- Ye modules Python ke saath pehle se hi aate hain, isliye inhe install karne ki zarurat nahi hoti.
- Examples:
  - `math`: Maths ke operations (like square root, sin, cos, etc.)
  - `os`: Operating system ke functions (like file management).
  - `sys`: System-specific functions (like command-line arguments).

Example:
```python
import math
print(math.sqrt(16))  # Output: 4.0
```

2. User-defined Modules

- Ye aap khud bana sakte ho apne functions aur classes ko ek alag file mein likh kar.
- Example:
  - Suppose, you create a file called `greetings.py`:

python
Copy code
```python
# greetings.py
def say_hello(name):
    return f"Hello, {name}!"
```

- Phir aap use kisi aur file mein import kar sakte ho:

python
Copy code
```
import greetings
print(greetings.say_hello("Alice"))  # Output: Hello, Alice!
```

3. Third-party Modules

- Ye modules dusre log banaate hain aur aap inhe Python ki package manager `pip` se install kar sakte ho.
- Examples:
  - `requests`: Websites aur APIs ke saath kaam karna.
  - `numpy`: Mathematical operations aur arrays ke liye.

Install:
bash
Copy code
```
pip install requests
```

Example:
python
Copy code
```
import requests
response = requests.get("https://api.github.com")
print(response.status_code)  # Output: 200 (agar website sahi se reachable hai)
```

---

4. Compiled Modules (C Extensions)

- Ye modules dusre languages (jaise C) mein likhe jaate hain taaki performance fast ho sake.
- Example: `numpy`, jo C mein likha gaya hai taaki mathematical operations fast ho sake.

5. Namespace Modules

- Ye multiple modules ko ek common name ke under group karte hain.
- Example: `collections` module mein alag-alag data types hote hain jaise `namedtuple`, `Counter`, etc.

---

3. How to Use Python Modules

1. Importing a Module

- Module ko import karte hain agar uske functions ya classes use karni hain.

python
Copy code
```
import math
print(math.sqrt(25))  # Output: 5.0
```

---

2. Importing Specific Functions

- Agar aapko sirf kisi ek function ki zarurat hai, toh aap sirf wo function import kar sakte ho.

python
Copy code
```
from math import sqrt
print(sqrt(25))  # Output: 5.0
```

---

3. Importing with an Alias

- Aap module ko short name (alias) de sakte ho, jo code likhne mein asaan ho.

python
Copy code
```
import numpy as np
print(np.array([1, 2, 3]))  # Creates a NumPy array
```

---

4. Importing Everything (Not Recommended)

- Aap sab kuch import kar sakte ho, lekin ye **avoid karo** kyunki ye confusing ho sakta hai.

python
Copy code
```
from math import *
```

```
print(sin(0))  # Sin function without math prefix
```

---

4. How to Create Your Own Module

1. **Create a Python file** (module) — for example, `mymodule.py`:

python
Copy code
```
# mymodule.py
def greet(name):
    return f"Hello, {name}!"
```

2. **Import the Module** in another Python file:

python
Copy code
```
import mymodule
print(mymodule.greet("Alice"))  # Output: Hello, Alice!
```

---

5. Packages in Python

- **Package** ek collection hota hai related modules ka, jo folders mein organize hote hain.
- Package ka folder `__init__.py` file hona chahiye.

Example folder structure:

markdown
Copy code
```
mypackage/
    __init__.py
    module1.py
    module2.py
```

Aap modules ko package se import kar sakte ho:

python

Copy code

```
from mypackage import module1
```