WORKING WITH EXTERNAL LIBRARIES – COMPLETE CONCEPTS

Understanding Python Packages and pip

Packages are collections of modules that provide additional functionality. pip is used to install, upgrade, and remove packages.

```
# Installing a package
pip install requests

# Checking installed packages
pip list

pip install numpy
pip uninstall numpy
pip install --upgrade numpy
```

Working with Dates and Times (datetime Module)

The datetime module helps manage dates, times, and timestamps.

```
from datetime import datetime

now = datetime.now()
print(now)
print(now.strftime("%Y-%m-%d"))
```

Making HTTP Requests (requests Library)

The requests library is used to fetch data from websites and APIs.

```
import requests

response = requests.get("https://api.github.com")
print(response.status_code)
print(response.json())
```

Working with JSON Data

JSON is a widely used data format for sending and receiving structured data.

```
import json

data = '{"name": "Ava", "age": 20}'
obj = json.loads(data)
print(obj["name"])

python_data = {"course": "Python", "level": "Beginner"}
json_text = json.dumps(python_data, indent=4)
print(json_text)
```

Virtual Environments & Dependency Management

Virtual environments help isolate project-specific dependencies.

```
# Creating a virtual environment
python3 -m venv myenv

# Activating (Mac/Linux)
source myenv/bin/activate

# Activating (Windows)
myenv\Scripts\activate

# Deactivate
deactivate
```

Reading Documentation & Using External APIs

Official documentation provides usage details and examples for libraries and APIs.

```
import requests

url = "https://api.example.com/data"
params = {"user": "Ava", "limit": 5}

response = requests.get(url, params=params)
print(response.json())
```