NexusHive

KEEP LEARNING, KEEP BUZZING

NexusHive
KEEP LEARNING, KEEP BUZZING

Essential Data Concepts in Python

Course Name: : Python Basics for Data Enthusiasts- Python and Data Science
Python Package and Data Structure – Packages in Python

www.nexushive.ai

What Are Packages?

Understanding Packages

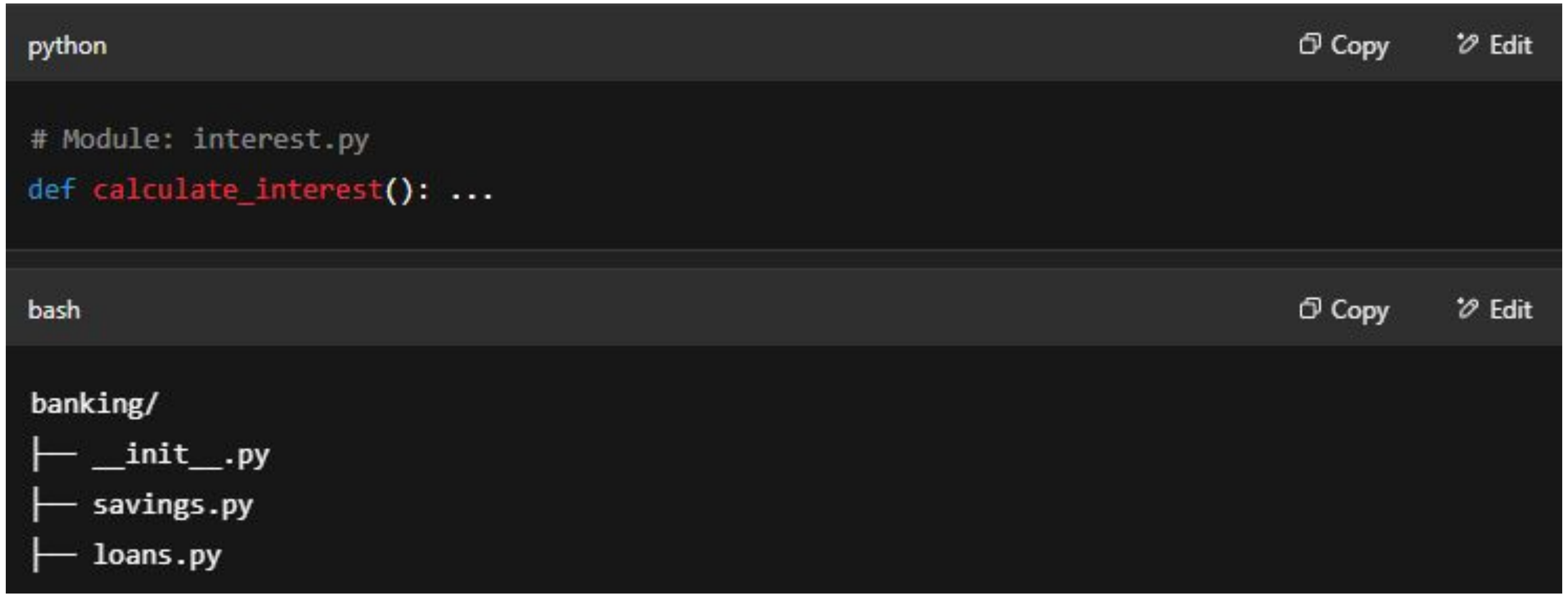
- **Packages are a collection of related Python modules**
- **Think of them as folders that help group code**
- **Each module can contain:**
  - Variables
  - Functions
  - Classes
- **Analogy: Like organizing files into folders on your computer**

Modules vs. Packages

Heading: Modules vs Packages

Modules	Packages
Single Python file (.py)	Folder with multiple modules
Can contain code, functions, and classes	Used to organize modules logically

Example:



```python
# Module: interest.py
def calculate_interest(): ...
```

```bash
banking/
├── __init__.py
├── savings.py
├── loans.py
```

Why Use Packages?

Benefits of Packages

- Better code organization
- Promotes reusability
- Helps with maintenance
- Enables collaboration across large projects

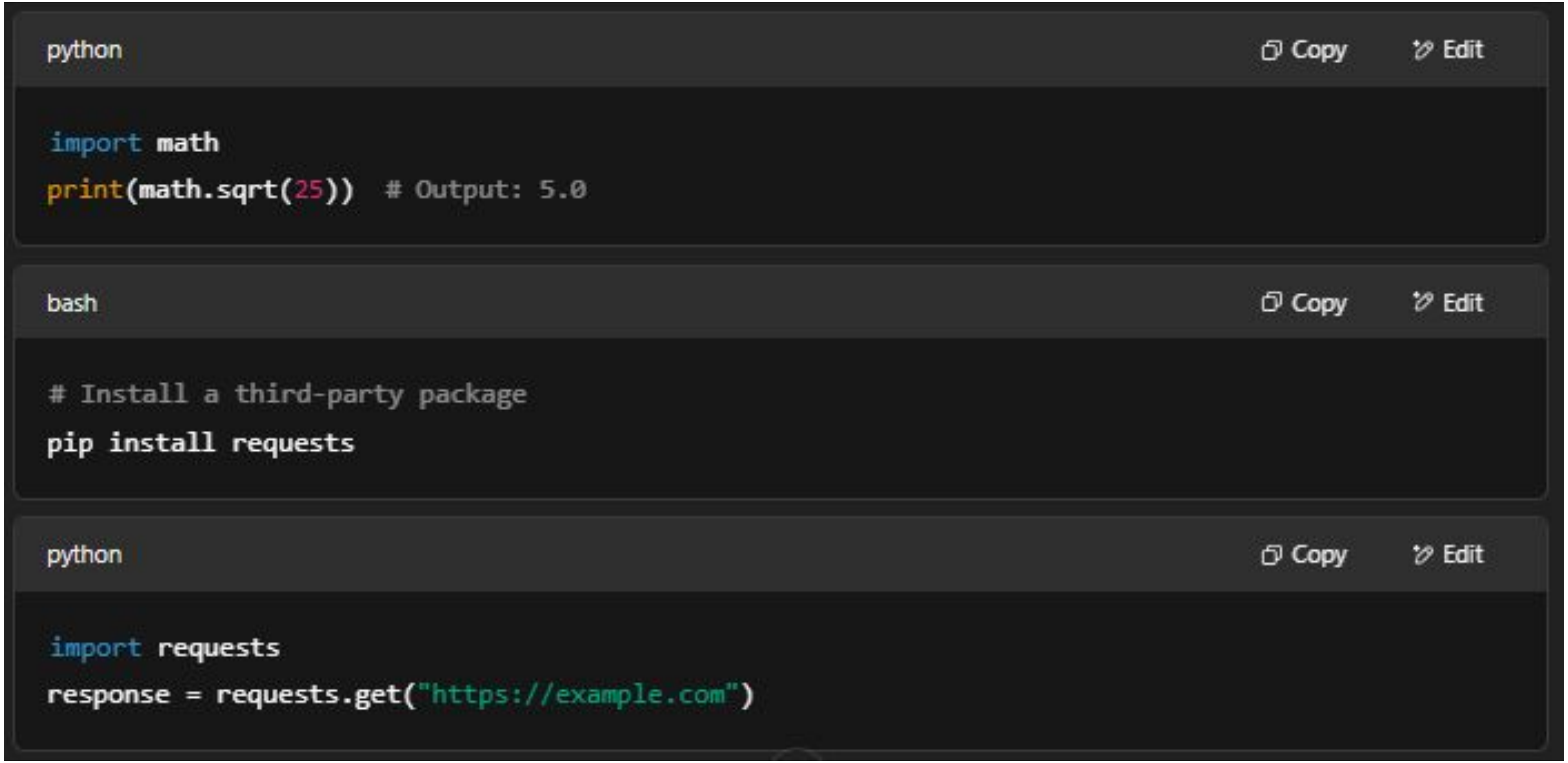
Example: A banking app can separate customer services, transactions, and reports into packages

Types of Packages

1. **Types of Python Packages**
   - Standard Library Packages
2. **Pre-installed (e.g., math, datetime, os)**
   - Third-Party Packages
3. **Install with pip (e.g., numpy, pandas)**
   - Custom Packages
4. **Built by developers for specific needs**

Using Packages in Code

How to Use Packages



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

```bash
# Install a third-party package
pip install requests
```

```python
import requests
response = requests.get("https://example.com")
```

Recap

Summary

1. Packages are directories of related modules
2. Help structure and scale Python projects
3. Can be built-in, third-party, or custom
4. Use import to access package content

Closing Note: Packages help organize, reuse, and maintain code effectively across projects!