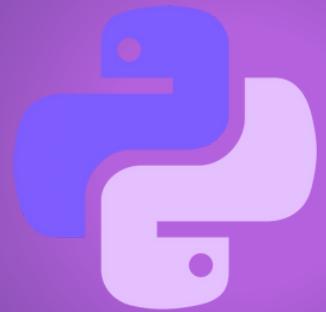


PYTHON

MODULES & PACKAGES



SWIPE

Modules and packages are essential components of Python programming that enable code organization, reusability, and modularity. Understanding how to leverage modules and packages effectively can take your Python development skills to the next level!

◆ **Modules:**

- Modules in Python are simply Python files (with a **.py** extension) that contain Python code, including functions, classes, and variables.
- They serve as containers for reusable code and help organize related functionality into separate files.
- By importing modules into your Python scripts or interactive sessions, you can access and utilize the code defined within them.



Example: Using the math module



Python

```
# to calculate the square root
import math

result = math.sqrt(25)
print(result)
```

Output:



5.0

In this example, we import the **math** module and use the **sqrt()** function defined within it to calculate the square root of 25.

< SWIPE

◆ Packages:

- Packages in Python are directories that contain multiple Python modules, along with a special `__init__.py` file that indicates to Python that the directory should be treated as a package.
- Packages allow you to organize related modules into hierarchical structures, making it easier to manage and maintain large projects.
- By importing packages, you can access the modules and functionality contained within them.

Example: using Package

```
my_package/  
    __init__.py  
    module1.py  
    module2.py
```

To use modules from the **my_package** package:



```
# Example: Importing modules from a package
from my_package import module1, module2

module1.function1()
module2.function2()
```

In this example, we import **module1** and **module2** from the **my_package** package and call functions defined within them.

Key Benefits:

- 1 Code Organization:** Modules and packages help organize code into logical units, making it easier to manage and maintain.
- 2 Code Reusability:** By encapsulating code into modules and packages, you can reuse functionality across multiple projects or scripts.
- 3 Modularity:** Modules and packages promote modular design principles, allowing you to build complex systems by combining smaller, reusable components.



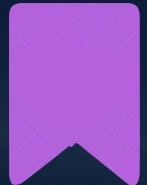
**DID YOU FIND
THIS HELPFUL?**



Like



Comment



Save



Share

www.linkedin.com/in/neethubabu921