

Day 11 – Packages, Modules, Functions & User Input

- Understand how to use Python's built-in modules (especially math)
- Learn various ways to take and process user input

Packages, Modules & Functions

- A package is a folder of modules
- A module is a file that can have functions
- Functions can be built-in or user-defined

Built-in Modules: math - I imported the math module to use functions like `sqrt()` and `pow()`

```
In [11]: # Import the whole math module
import math

# Using math functions with 'math.' prefix
print("Square root of 25:", math.sqrt(25))
print("2 raised to 3:", math.pow(2, 3))
print("Floor of 3.8:", math.floor(3.8))
print("Ceil of 3.1:", math.ceil(3.1))
print("Value of pi:", math.pi)
print("Value of e:", math.e)
```

```
Square root of 25: 5.0
2 raised to 3: 8.0
Floor of 3.8: 3
Ceil of 3.1: 4
Value of pi: 3.141592653589793
Value of e: 2.718281828459045
```

```
In [12]: # Import math with an alias 'm'
import math as m

# ► Access math functions using alias
print("Square root using alias:", m.sqrt(49))
print("Floor using alias:", m.floor(7.9))
```

```
Square root using alias: 7.0
Floor using alias: 7
```

```
In [13]: # Import specific functions directly
from math import sqrt, floor, pow

# ► Now we can call these without 'math.'
print("Square root of 64:", sqrt(64))
print("5 to the power 2:", pow(5, 2))
print("Floor of 9.8:", floor(9.8))
```

```
Square root of 64: 8.0
5 to the power 2: 25.0
Floor of 9.8: 9
```

```
In [14]: # Using round() with math.pow()
print("Rounded power value:", round(pow(2, 3.4)))
```

```
Rounded power value: 11
```

User Input in Python

```
In [18]: # input() always returns a string
a = input("Enter a value: ") # 5
b = input("Enter another value: ") # 5
print(a + b) # Output: '55' (string concatenation)
```

55

```
In [19]: # Converting input to numbers
a = int(input("Enter first number: ")) # 5
b = int(input("Enter second number: ")) # 6
print("Sum is:", a + b) # 11
```

Sum is: 11

```
In [21]: # Indexing and Slicing Input

name = input("Enter your name: ") # Arman
print("Character at index 3:", name[3]) # a
print("Slice from 1 to 3:", name[1:3]) # rm
```

Character at index 3: a

Slice from 1 to 3: rm

```
In [22]: # Slicing directly on input
ch = input("Enter any word or characters: ")[1:3]
print("Characters from index 1 to 2:", ch)
```

Characters from index 1 to 2: rm

```
In [24]: # Evaluating expressions from input using eval()
expr = eval(input("Enter an expression (e.g., 10+5): ")) # 20+30
print("Result:", expr) # 50
```

Result: 50