🧠 Topic 2: Functions in Python

🔹 Definition

A **function** in Python is a block of reusable code that performs a specific task. You define a function once and can use (or call) it many times throughout your program.

It helps make your code **modular, clean, and reusable**.

🔹 Terminologies

- **Function Definition:** The block of code created using the def keyword.
- **Function Name:** The unique name you assign to the function.
- **Parameters:** Variables that accept input values inside the function definition.
- **Arguments:** Actual values you pass into the function when calling it.
- **Return Statement:** Used to send back a result from the function to the caller.
- **Calling a Function:** The act of executing a function by using its name followed by parentheses ().

🔹 Explanation

A function helps you avoid writing the same code repeatedly.
When Python runs your program, it will skip a function's body until the function is called.

We use functions to:
✅ Break large problems into smaller parts.
✅ Reuse code easily.
✅ Improve readability and debugging.
✅ Handle input/output logic separately.

🔹 Syntax

```
def function_name(parameters):
    # code block
    return value
```

◆ **Examples + Outputs**

Example 1

```
def greet():
    print("Hello, welcome to Python!")
greet()
```

Output:

```
Hello, welcome to Python!
```

Example 2

Using **parameters and arguments**:

```
def greet_user(name):
    print("Hello", name)
greet_user("Maverick")
```

Output:

```
Hello Maverick
```

Example 3

Using **return statement**:

```
def add(a, b):
    return a + b

result = add(5, 3)
print("Sum:", result)
```

Output:

```
Sum: 8
```

Example 4

Using **default parameters**:

```
def power(base, exponent=2):
    return base ** exponent


print(power(5))
print(power(5, 3))
```

Output:

```
25
125
```

🔹 Benefits of Functions

✅ Reusability – write once, use many times.
✅ Organization – code becomes structured and readable.
✅ Scalability – easy to expand or modify.
✅ Debugging – fix issues in one place instead of everywhere.

🔹 20 Challenges (Practice Problems)

✅ Solved Challenge 1

Question: Create a function that prints your name and age.

```
def info(name, age):
    print("Name:", name)
    print("Age:", age)
```

```
info("Maverick", 25)
```

Output:

```
Name: Maverick

Age: 25
```

✅ Solved Challenge 2

Question: Create a function that returns the area of a rectangle.

```
def rectangle_area(length, width):
    return length * width


print(rectangle_area(5, 10))
```

Output:

```
50
```

🔸 Your 18 Challenges

1. Create a function that prints “Hello, World!”.
2. Write a function that adds three numbers.
3. Create a function that takes a string and prints it in uppercase.
4. Write a function that returns the square of a number.
5. Create a function that checks if a number is even or odd.
6. Write a function that returns the sum of all numbers in a list.
7. Create a function that finds the largest number in a list.
8. Write a function that converts Celsius to Fahrenheit.
9. Create a function that takes a name and prints a greeting message.
10. Write a function that counts how many vowels are in a string.

11. Create a function that returns the factorial of a number.
12. Write a function that reverses a string.
13. Create a function that multiplies all elements in a list.
14. Write a function that checks whether a word is a palindrome.
15. Create a function that returns the average of three numbers.
16. Write a function that prints numbers from 1 to 10 using a loop.
17. Create a function that prints multiplication table of a number.
18. Write a function that prints the length of a string.