

Functions:

Definition: A **function in Python** is a reusable block of code that performs a specific task. Instead of writing the same code many times, you define a function once and call it whenever you need it.

1. Basic Idea of a Function

Think of a function like a machine:

- You give it **input** (optional)
- It processes something
- It gives you **output** (optional)

2. How to Define a Function

```
def function_name():  
    print("Hello from the function!")
```

Calling the function:

```
function_name()
```

Output:

Hello from the function!

3. Function with Parameters (Inputs)

```
def greet(name):  
    print("Hello", name)
```

Call:

```
greet("Ali")
```

Output:

Hello Ali

4. Function with Return Value

Instead of printing, functions can return a value.

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

Call:

```
result = add(3, 5)  
print(result)
```

👉 Output:

8

5. Function with Default Parameters

```
def greet(name="Guest"):  
    print("Hello", name)
```

Call:

```
greet()  
greet("Sara")
```

👉 Output:

Hello Guest
Hello Sara

6. Multiple Parameters Example

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

Call:

```
student_info("Yacine", 22)
```

7. Practice Examples

Example 1: Even or Odd

```
def check_even(number):
    if number % 2 == 0:
        return "Even"
    else:
        return "Odd"

print(check_even(4))
```

Example 2: Square of a Number

```
def square(n):
    return n * n

print(square(6))
```

Example 3: Simple Calculator

```
def calculator(a, b, operation):
    if operation == "+":
        return a + b
    elif operation == "-":
        return a - b
    elif operation == "*":
        return a * b
    elif operation == "/":
        return a / b
    else:
        return "Invalid operation"

print(calculator(10, 5, "+"))
```

8. Important Notes

- Always use def to create a function
- Indentation (spaces) is **very important**
- Use return if you want to get a result back
- Function names should be clear (like add_numbers, check_age)

9. Small Exercises (Try Yourself)

Exercise 1:

Create a function that returns the **maximum of two numbers**

Exercise 2:

Create a function that prints numbers from 1 to 10

Exercise 3:

Create a function that checks if a number is **positive, negative, or zero**

Bonus (Slightly Advanced)

```
def factorial(n):
```

```
    result = 1
```

```
    for i in range(1, n + 1):
```

```
        result *= i
```

```
    return result
```

```
print(factorial(5))
```

Exercise 1: Maximum of Two Numbers

```
def max_number(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
# Test
```

```
print(max_number(10, 7))
```

👉 Output:

10

✅ **Exercise 2: Print Numbers from 1 to 10**

```
def print_numbers():  
    for i in range(1, 11):  
        print(i)
```

```
# Call the function  
print_numbers()
```

👉 Output:

```
1  
2  
3  
...  
10
```

✅ Exercise 3: Check Positive, Negative, or Zero

```
def check_number(n):  
    if n > 0:  
        return "Positive"  
    elif n < 0:  
        return "Negative"  
    else:  
        return "Zero"
```

```
# Test  
print(check_number(-5))
```

👉 Output:

```
Negative
```

Exercise 1: Sum of Three Numbers

Write a function that takes **3 numbers** and returns their sum.

✅ Solution:

```
def sum_three(a, b, c):  
    return a + b + c
```

```
print(sum_three(2, 4, 6))
```

● Exercise 2: Check Largest of Three Numbers

Create a function that returns the **largest of 3 numbers**.

✓ Solution:

```
def largest(a, b, c):  
    if a >= b and a >= c:  
        return a  
    elif b >= a and b >= c:  
        return b  
    else:  
        return c
```

```
print(largest(3, 9, 5))
```

● Exercise 3: Count Vowels

Write a function that counts how many **vowels** are in a string.

✓ Solution:

```
def count_vowels(text):  
    vowels = "aeiouAEIOU"  
    count = 0  
    for char in text:  
        if char in vowels:  
            count += 1  
    return count
```

```
print(count_vowels("Hello World"))
```

● Exercise 4: Multiplication Table

Create a function that prints the multiplication table of a number.

✓ Solution:

```
def multiplication_table(n):  
    for i in range(1, 11):
```

```
print(n, "x", i, "=", n * i)
```

```
multiplication_table(5)
```

● Exercise 5: Factorial (Using Function)

Write a function that returns the factorial of a number.

✓ Solution:

```
def factorial(n):  
    result = 1  
    for i in range(1, n + 1):  
        result *= i  
    return result
```

```
print(factorial(5))
```

● Exercise 6: Reverse a String

Write a function that returns the **reverse of a string**.

✓ Solution:

```
def reverse_string(text):  
    return text[::-1]
```

```
print(reverse_string("Python"))
```

● Exercise 7: Check Prime Number

Write a function that checks if a number is **prime**.

✓ Solution:

```
def is_prime(n):  
    if n <= 1:  
        return False  
    for i in range(2, n):  
        if n % i == 0:  
            return False
```

```
return True
```

```
print(is_prime(7))
```

● Exercise 8: Sum of List Elements

Write a function that takes a list and returns the sum.

✅ Solution:

```
def sum_list(numbers):  
    total = 0  
    for num in numbers:  
        total += num  
    return total
```

```
print(sum_list([1, 2, 3, 4]))
```

● Exercise 9: Find Even Numbers in a List

Return a list of only even numbers.

✅ Solution:

```
def get_even(numbers):  
    result = []  
    for num in numbers:  
        if num % 2 == 0:  
            result.append(num)  
    return result
```

```
print(get_even([1, 2, 3, 4, 5, 6]))
```