📘 Lecture 9: User-Defined Functions in Python

🧠 What Is a Function?

A **function** is a **reusable block of code** designed to do a specific task.

Think of it like a **tool**:
🔧 Input → 🛠️ Function → 📤 Output

🎯 1. Purpose of the Function

- Every function must solve **one clear task**.
- Example: A function to **add two numbers**.

📝 2. Naming the Function

- Use **meaningful names**:
  - ✅ `add_numbers()`
  - ❌ `a()`, `fun1()`

□ 3. Defining Parameters

Syntax:

```
python
CopyEdit
def add(num1, num2):
```

Parameters are **inputs** the function works with.

□ 4. Function Logic + 5. Return Output

Example:

```
python
CopyEdit
def add(num1, num2):
    print("Number 1:", num1)
    print("Number 2:", num2)
    return num1 + num2
```

- Use `return` to **send result** back.
- `print()` just **displays**; doesn't return.

📞 6. Calling the Function

```
python
CopyEdit
result = add(2, 4)
print("Result:", result)  # Output: 6
```

⚠️ Defining ≠ Calling. The function only works when **called**.

⚙️ Types of Function Parameters

Type	Syntax	Example
☐ **Default**	`def greet(name="John")`	`greet()` → John
🏷️ **Keyword**	`greet(first_name="Ali")`	Order doesn't matter
✨ **Variable Length**	`def greet(*names)`	`greet("Ali", "Sara")`

```
python
CopyEdit
def greet(*names):
    for name in names:
        print(f"Hello {name}")
```

🔄 Return vs Print

Feature	`print()`	`return`
Output to Console	✅	❌
Save Value	❌	✅
Reuse Later	❌	✅

Examples:

```
python
CopyEdit
def greet():
    print("Hello Python")   # Only prints

def greet():
    return "Hello Python"   # Can be stored

message = greet()
print(message)
```

📚 Additional Examples

✅ Function with Return

```
python
CopyEdit
def add(num1, num2):
    return num1 + num2

print(add(3, 7))  # 10
```

□ No Parameters

```
python
CopyEdit
def greet():
    print("Welcome to Python")

greet()
```

⚙️ Default Parameters

```
python
CopyEdit
def full_name(first_name="John", last_name="Doe"):
    print(f"Full Name: {first_name} {last_name}")
```

```
full_name()              # John Doe
full_name("Jane")        # Jane Doe
```

✅ Best Practices

- ✔ Use descriptive names
- ✔ Keep functions small & focused
- ✔ Use return when result is needed
- ✔ Avoid unnecessary global variables
- ✔ Combine with loops/conditions for real logic

🧠 Final Thoughts

- ✅ Functions = Building blocks of efficient code
- ✅ Use for **modularity**, **reusability**, and **clarity**
- ✅ Practice defining + calling + returning
- ✅ Know the difference: `print()` vs `return`
- ✅ Mix with conditions/loops to solve real problems

🔁 Revision Table

Topic	Example
Define	`def add(a, b):`
Call	`add(2, 5)`
Default	`def greet(name="Ali")`
Return	`return a + b`
Print	`print(a + b)`
