# Lecture 8: Functions in Python

---

## Real-Life Analogy: The Chef

Imagine a chef getting **repeated orders**:

- ❌ **Option 1**: Rewrite the recipe each time
- ✅ **Option 2**: **Write once**, reuse many times (like a **function**!)

🧠 **Moral**: Instead of repeating code, define a function and call it when needed.

---

## What is a Function?

A **function** is a block of code that runs **only when called**. It may take **input(s)** and give **output**.

Like a **juicer**:

- Input: Fruit
- Output: Juice

---

## Benefits of Using Functions

| Benefit | Description |
|---|---|
| **Reusability** | Write once, use many times |
| **Modularity** | Organize your program into sections |
| **Clean Code** | Easier to understand & manage |
| **Less Errors** | Reuse avoids mistakes from rewriting |

---

## Built-in Python Functions

Python comes with **built-in tools**. Here are some:

| Function | Description | Example | Output |
|---|---|---|---|
| `len()` | Length of string/list | `len("Python")` | `6` |
| `max()` | Max value | `max([1, 2, 3])` | `3` |
| `sorted()` | Sort list | `sorted([3, 1, 2])` | `[1, 2, 3]` |
| `range()` | Generates sequence | `range(5)` | `0 to 4` |
| `id()` | Memory address | `id(x)` | `e.g., 139452932` |
| `eval()` | Evaluates string as code | `eval("5 + 3 * 2")` | `11` |

---

# 📥 Input & 📤 Output Example

```
python
CopyEdit
name = input("Enter your name: ")
print("Hello", name)
```

📝 Also:

```
python
CopyEdit
print(f"The length of the string is {len(name)}")
```

---

# 🔄 Sorting

```
python
CopyEdit
ratings = [4.5, 3.0, 5.0, 4.0]
print(sorted(ratings))               # [3.0, 4.0, 4.5, 5.0]
print(sorted(ratings, reverse=True)) # [5.0, 4.5, 4.0, 3.0]
```

---

# 🧮 Using `eval()` for Math Strings

```
python
CopyEdit
expression = "5 + 3 * 2"
result = eval(expression)
print(result)  # 11
```

⚠ Use `eval()` carefully — it runs code from strings!

---

# □ Types of Functions

| Type | Description | Examples |
|---|---|---|
| **Built-in** | Already in Python | `print()`, `input()`, `len()`, `eval()` |
| **User-defined** | You create them! | `def greet():` |

---

## ✍ User-Defined Function Example

```
python
CopyEdit
def greet(name):
    print(f"Hello, {name}!")
```

📞 Call it like this:

```
python
CopyEdit
greet("Alice")  # Output: Hello, Alice!
```

---

# 🧠 Best Practices

✅ Give functions **meaningful names**
✅ Use functions to **avoid repetition**
✅ Keep them **short and specific**
✅ Use comments and **docstrings** to explain purpose

---

# 🧠 Final Thoughts

🔷 Functions = Backbone of clean, reusable code
🔷 Built-ins make your life easier
🔷 User-defined let you **customize behavior**
🔷 PRACTICE = MASTERING FUNCTIONS

---

# 🔗 Quick Review Table

| Concept | Example |
| --- | --- |
| Function Definition | `def greet():` |
| Function Call | `greet()` |
| With Parameter | `def greet(name):` |
| Built-in | `print(),max()` |
| Custom | `def square(x): return x*x` |

---