# 🐍 PYTHON BASICS EXPLAINED WITH TECH SCENARIOS

---

## 1. Variables

- **Definition:** Variables are containers for storing data values.
- **Scenario:** A **food delivery app** stores a customer's name (`str`), the order amount (`float`), and delivery status (`bool`).

```
name = "Shiva"
total = 299.99
delivered = False
```

---

## 2. Data Types

- **Definition:** Python supports different types like `int`, `float`, `str`, `bool`, and more.
- **Scenario:** An **eCommerce app** uses:
  - `int` for product stock count
  - `float` for price
  - `str` for product name
  - `bool` for is_available

---

## 3. Operators

- **Definition:** Operators perform operations on variables and values.
- **Scenario:** In a **billing system**, the total bill is calculated using:

```
price = 150
tax = 18
total = price + tax
```

---

## 4. Input and Output

- **Definition:** `input()` gets user input, `print()` shows output.
- **Scenario:** A **login system** takes a username as input and greets the user.

python

```
username = input("Enter your name: ")
print("Welcome,", username)
```

---

## 5. Conditional Statements (if/else)

- **Definition:** These control what code runs based on conditions.
- **Scenario:** A **banking app** checks if the balance is enough before withdrawing:

python

```
if balance >= amount:
    print("Withdrawal successful")
else:
    print("Insufficient funds")
```

---

## 6. Loops (for, while)

- **Definition:** Loops repeat a block of code.
- **Scenario:** An **email service** sends emails to 100 users:

python

```
for email in user_list:
    send_email(email)
```

---

## 7. Lists

- **Definition:** A list holds multiple items in a single variable.
- **Scenario:** A **shopping cart** stores selected items:

python
```
cart = ["shoes", "watch", "t-shirt"]
print(cart)
```

---

## 8. Dictionaries

- **Definition:** A dictionary stores data in key-value pairs.
- **Scenario:** A **product catalog** with name, price, and stock:

python

```
product = {"name": "Laptop", "price": 75000, "stock": 12}
print(product)
print(product.keys()) # print all the keys
```

## 9.Functions

- **Definition:**A **function** is a reusable block of code that performs a specific task. It helps reduce repetition and makes code clean, organized, and powerful.
- **Scenario:** You create a function that calculates the total fare based on distance.

python

```
def calculate_fare(distance):#function definition,distance is parameter
    return distance * 12
print("Total fare: ₹", calculate_fare(10))# function call,10 is argument
```

## 10.Exception Handling (try-except)

- **Definition:** It handles errors without crashing the program.
- **Scenario:** A **payment gateway** handles wrong card input:

python

```
try:
    A = 10
    print(A)
except:
    print("Payment failed. Try again.")
```