

# Python COURSE

---

## Q: What is Python?

**A: Python = Easy, Powerful Language that makes computers talk like humans** — simple, easy to read, and powerful.

□ Memory trick: Think of a “python snake” — it slithers smoothly (code runs smoothly).

✓ Example:

```
print("Hello, World!")
```

---

## Q: Key Features of Python?

- High-level → like giving instructions in English, not machine code.
- Interpreted → like a translator reading your instructions line by line.
- General-purpose → can cook anything: web, AI, games, automation.
- Analogy: Like a Swiss Army Knife for coding.

✓ Example:

```
# Python is interpreted → runs line by line
x = 10
y = 5
print(x + y)    # 15
```

---

## Q: Python Programming Applications?

- AI & ML (brains of smart apps)
- Web Development (Django, Flask)
- Data Science (pandas, NumPy)
- Automation (scripts that save time)
- Games, GUI apps, IoT
- Trick: “A-WaDa-Ga” (AI, Web, Data, Games, Automation).

✓ Example:

```
# Simple automation: opening a website
import webbrowser
webbrowser.open("https://www.google.com")
```

---

## Q: What is a Variable?

**A:** A variable is a container to store data.

□ Analogy: Like a jar with a label — you put sugar in one, rice in another.

✓ Example:

```
name = "Bilal"
age = 25
print(name, age)
```

---

## Q: Data Types in Python?

- int → 5 (whole numbers)
- float → 3.14 (decimals)
- str → "Hello" (text)
- bool → True/False (logic)
- list → [1,2,3] (groceries list)
- tuple → (1,2,3) (unchangeable shopping list)
- dict → {"name":"Bilal"} (phonebook)
- Trick: "I Feel So Badly Like The Dull" (int, float, str, bool, list, tuple, dict).

✓ Example:

```
x = 10           # int
pi = 3.14        # float
msg = "Hello"    # str
flag = True      # bool
nums = [1, 2, 3] # list
days = ("Mon","Tue") # tuple
info = {"name":"Bilal", "age":25} # dict
```

---

## Q: Mutability & Use Case?

- Mutable: can change (list, dict, set).
- Immutable: fixed (int, str, tuple).
- Analogy: Mutable = whiteboard (erasable), Immutable = stone carving (permanent).

✓ Example:

```
nums = [1,2,3]
nums[0] = 99 # mutable
print(nums)  # [99, 2, 3]
```

```
name = "Bilal"
# name[0] = "Z" ❌ error → immutable
```

---

## Q: Conditional Statements?

**A:** They let your program decide things.

□ Analogy: If it's raining, take an umbrella.

✓ Example:

```
age = 18
if age >= 18:
    print("You can vote")
else:
    print("Too young")
```

---

## Q: Logical Operators?

- and → both must be true.

- or → at least one true.

- not → reverses.

□ Analogy: Like electricity switches: both ON (and), either ON (or), flip it (not).

✓ Example:

```
x = 5
print(x > 0 and x < 10) # True
print(x > 10 or x == 5) # True
print(not(x == 5))      # False
```

---

## Q: if, if-else, elif Statements in Python?

- if → one road.

- if-else → either this road or that.

- elif → multiple roads to check.

□ Analogy: Choosing food: if pizza → eat it, elif burger → eat that, else → just drink water.

✓ Example:

```
food = "burger"
if food == "pizza":
    print("Eat pizza")
elif food == "burger":
    print("Eat burger")
else:
```

```
print("Drink water")
```

---

## Q: Loops & Types?

- for loop → repeat over a sequence (like reading names in a class roll).
- while loop → repeat until condition false (like jogging until tired).

✓ Example:

```
for i in range(3):  
    print("For loop:", i)  
  
x = 0  
while x < 3:  
    print("While loop:", x)  
    x += 1
```

---

## Q: Iteration with Python Loops?

A: Going through items one by one (like distributing candies to students).

✓ Example:

```
fruits = ["apple", "banana", "mango"]  
for fruit in fruits:  
    print(fruit)
```

---

## Q: range() Function in Python?

A: Creates a sequence of numbers (like a number line).

□ Trick: range(5) → [0,1,2,3,4].

✓ Example:

```
for i in range(2,10,2):  
    print(i)    # 2, 4, 6, 8
```

---

## Q: break Statement?

A: Exits the loop immediately.

□ Analogy: Emergency stop button in a bus.

✓ Example:

```
for i in range(5):  
    if i == 3:  
        break  
    print(i)
```

---

### Q: continue Statement?

**A:** Skips current step, continues loop.

□ Analogy: Skip a song but keep the playlist playing.

✓ Example:

```
for i in range(5):  
    if i == 2:  
        continue  
    print(i)
```

---

### Q: pass Statement?

**A:** Does nothing (placeholder).

□ Analogy: Teacher says “skip this question.”

✓ Example:

```
def future_code():  
    pass
```

---

### Q: Functions in Python?

**A:** A block of reusable code.

□ Analogy: Like a vending machine — input money, get snacks.

✓ Example:

```
def greet(name):  
    return f"Hello, {name}"  
  
print(greet("Bilal"))
```

---

### Q: Keyword Arguments in Python?

**A:** Pass values with names (order doesn't matter).

□ Analogy: Ordering pizza by naming toppings instead of position.

✓ Example:

```
def intro(name, age):  
    print(f"My name is {name}, I am {age} years old")  
  
intro(age=25, name="Bilal")
```

---

## Q: Global vs Local Variable?

- Local → inside function (private, temporary).
- Global → outside function (everyone can access).
- Analogy: Local = food in your plate, Global = food on dining table.

✓ Example:

```
x = "global"  
  
def test():  
    x = "local"  
    print("Inside function:", x)  
  
test()  
print("Outside function:", x)
```