CODEBOOSTERS TECH

THINK LEARN GRAB

CODEBOOSTERS TECH - LeetCode Training

✅ Day 1 FN Session Plan — Python Basics

🧠 Session Objective:

> Lay a solid foundation in Python programming to confidently solve beginner-level coding problems on LeetCode by afternoon.

📋 Topics Covered :

1. What is Python? Why Python for Problem Solving?
2. Variables & Data Types
3. Type Casting & Type Checking
4. Input & Output
5. Conditional Statements
6. Loops – `for`, `while`, `break`, `continue`
7. Lists & Tuples
8. Strings & String Operations

9. Functions – User-defined, Built-in
10. Extra: Dictionaries, Sets (Intro only)
11. Bonus: Debugging Basics (print/debug strategy)

Now let's go topic-by-topic with deep theory, real-world analogies, hands-on questions, and line-by-line explanations.

🔸 1. What is Python?

✅ Why Python?

- Simple syntax → readable like English
- Large community + tons of libraries
- Used in: Web dev, Data Science, Automation, AI, Competitive Programming
- Perfect for problem solving and LeetCode

🧠 Real-life Analogy:

> If C/C++ is like a manual car (more control, more complexity), Python is an automatic (easy to drive, gets job done fast).

🧪 Micro Exercise:

```python
print("Welcome to Python training!")
```

📌 Output:

```
css

Welcome to Python training!
```

Q: What did we just do here?

A: Used the built-in `print()` function to display a message on the screen.

🔶 2. Variables & Data Types

✅ What is a variable?

A container to store data in memory using a name.

🧠 Analogy:

> Think of a cup labeled "Coffee" — the label is the variable name, and the drink is the value.

🔥 Real-Time Example:

```python

user_name = "Veera"
user_age = 28
user_city = "Hyderabad"
is_active_user = True
```

🧪 Hands-On Drill:

```python

```

```
company = "Google"
founded = 1998
rating = 4.8
is_hiring = True

print(company, "was founded in", founded)
```

🪛 Line-by-Line Explanation:

- `company = "Google"` → Creates a string variable
- `founded = 1998` → Integer
- `is_hiring = True` → Boolean
- `print(...)` → Combines and prints all info

🔶 3. Type Casting & Type Checking

✅ Type Checking:

Use `type()` to check the datatype.

```python
a = 5
b = "5"
print(type(a))  # <class 'int'>
print(type(b))  # <class 'str'>
```

✅ Type Conversion:

```python
a = int("5")  # string to int
b = str(10)   # int to string
```

⚠ Important:

> Input is always `str` type!

🧪 Practice:

```python
num1 = input("Enter a number: ")
num2 = input("Enter another number: ")
print("Sum is:", int(num1) + int(num2))
```

🔸 4. Input & Output

✅ `input()` ⇒ to take user input

✅ `print()` ⇒ to show output

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

🔥 Real-World Example:

> Build a simple interest calculator

```python
p = float(input("Enter principal: ")) r = float(input("Enter rate: ")) t = float(input("Enter time: "))
```

```
si = (p * r * t) / 100
print("Simple Interest =", si)
```

🔶 5. Conditional Statements

✅ if / elif / else – for decisions

```python
age = int(input("Enter your age: "))

if age >= 18:
    print("You can vote!")
elif age > 12:
    print("You are a teenager!")
else:
    print("You are a child!")
```

🧪 Real-World Use:

- Login system (if password correct)
- Online order delivery (if pincode serviceable)
- Show tax slab (if income > certain range)

🔥 Task:

Write a program to print "even" or "odd" for a given number.

```python

```

```
num = int(input("Enter a number: "))
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

🔸 6. Loops – `for` , `while`

✅ `for` loop – iterate over range or collection

```python
for i in range(1, 6):
    print("Number:", i)
```

✅ `while` loop – repeat till condition breaks

```python
x = 5
while x > 0:
    print("Counting down:", x)
    x -= 1
```

🛠️ Special Keywords:

- `break` – stop loop early
- `continue` – skip current iteration

🧠 Real-Time Use:

- Send 10 emails
- Retry login 3 times
- Countdown timer

🔶 7. Lists

✅ Syntax:

```python
items = ["pen", "book", "bottle"]
```

💡 Key Actions:

```python
items.append("phone")
items.remove("book")
print(items[0])          # pen
print(len(items))        # 3
```

🔥 Task:

> Take 5 marks from user and print the average

```python
marks = []
for i in range(5):
```

```
    m = int(input(f"Enter mark {i+1}: "))
    marks.append(m)

average = sum(marks) / len(marks)
print("Average marks:", average)
```

🔸 8. Strings

✅ Strings are sequences of characters

```python
name = "python"
print(name[0])          # p
print(name.upper())     # PYTHON
```

🔥 Use Case:

> Format username into proper case

```python
raw_name = input("Enter your name: ")
print("Welcome,", raw_name.strip().capitalize())
```

🔸 9. Functions

✅ Syntax:

```python
```

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

greet("Veera")
```

🛠️Function with return:

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

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

🔥 Task:

> Write a function to check if a number is prime

```python
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))        # True
```

🔸 10.Dictionaries & Sets (Intro)

✅ Dictionary:

```python
student = {"name": "Veera", "age": 28, "city": "Hyderabad"}
print(student["city"])          # Hyderabad
```

✅ Set:

```python
colors = {"red", "blue", "green", "red"}
print(colors) # No duplicates
```

🔸 11. Debugging Tips

- Use `print()` at each step Break big code into
- smaller parts Check types with type()
- Comment unused `blocks`
- Google error message + try on small input
-

WWW.CODEBOOSTERS.IN