🐍 Python Basics Course

Topic: Input/Output (I/O) Statements & String Formatting

This note is written in a **simple**, **explained**, and **precise** way for **beginners**. It follows the same easy format so students can **understand quickly** and **remember for exams**.

📘 1. What are I/O Statements?

Meaning of I/O

- **Input (I)** → Taking data from the user
- **Output (O)** → Displaying data on the screen

Python mainly uses:

- `input()` for input
- `print()` for output

📘 2. Input Statement in Python

`input()` Function

The `input()` function is used to **take input from the user**.

Syntax:

```
variable = input("Message")
```

Example:

```
name = input("Enter your name: ")
print(name)
```

📝 Important Points:

- `input()` always takes data as **string**
- User enters data using keyboard

📘 3. Taking Numeric Input

Since `input()` gives string, we must **convert** it into numbers.

Integer Input:

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

Float Input:

```
salary = float(input("Enter your salary: "))
print(salary)
```

📝 Note:

- Use `int()` for whole numbers
- Use `float()` for decimal numbers

📘 4. Output Statement in Python

`print()` Function

The `print()` function is used to **display output** on the screen.

Basic Example:

```
print("Hello Python")
```

Printing Variables:

```
city = "Pokhara"
print(city)
```

📘 5. Printing Multiple Values

Using Comma in `print()`

```
name = "Ram"
age = 21
```

```
print(name, age)
```

Output:

```
Ram 21
```

📝 Note:

- Comma automatically adds space

📘 6. String Formatting in Python

What is String Formatting?

String formatting means **inserting variables inside a string** in a proper and readable way.

Python provides **three main ways** of string formatting.

📘 7. Method 1: f-String (Best & Recommended)

Syntax:

```
f"text {variable}"
```

Example:

```
name = "Sita"
age = 20
print(f"My name is {name} and my age is {age}")
```

📝 Important Points:

- Easy to read
- Most commonly used
- Recommended for beginners

📘 8. Method 2: `format()` Method

Syntax:

```
"string".format(variable)
```

Example:

```
name = "Hari"
age = 22
print("My name is {} and my age is {}".format(name, age))
```

📝 Note:

- Uses `{}` as placeholders

📘 9. Method 3: `%` Formatting (Old Style)

Example:

```
name = "Gita"
age = 19
print("My name is %s and my age is %d" % (name, age))
```

📝 Note:

- `%s` → string
- `%d` → integer
- This method is **old**, used in legacy programs

📘 10. Escape Characters in Print

Escape characters are used to **format output**.

Character	Meaning
`\n`	New line
`\t`	Tab space

Example:

```
print("Hello\nPython")
```

📘 11. Complete Example Program (I/O + Formatting)

```
# Student Details Program

name = input("Enter student name: ")
age = int(input("Enter age: "))
marks = float(input("Enter marks: "))

print("\n--- Student Information ---")
print(f"Name: {name}")
print(f"Age: {age}")
print(f"Marks: {marks}")
```

⭐ Important Exam Notes

✔ `input()` takes input as string
✔ Use type conversion for numbers
✔ `print()` is used for output
✔ f-string is best for formatting
✔ `\n` is used for new line
✔ String formatting improves readability

🎯 Conclusion

In this lesson, students learned:

- Input and Output statements
- Numeric input conversion
- String formatting methods
- Printing formatted output

These concepts are **very important** and are used in **almost every Python program**.

✨ *Practice small programs to gain confidence in I/O and formatting!*