Datatypes, Input/Output, and Printing Methods in Python

What is a Datatype?

In Python, a **datatype** defines the kind of value a variable can hold.
It helps the interpreter understand how data should be stored and what operations can be performed on it.

Types of Datatypes in Python with Examples

1. Numeric Types

- **int** → Integer numbers
  Example: x = 10
- **float** → Decimal numbers
  Example: y = 3.14
- **complex** → Complex numbers
  Example: z = 2 + 3j

---

2. Sequence Types

- **str** → String (text)
  Example: name = "Python"
- **list** → Ordered, changeable collection
  Example: numbers = [1, 2, 3]
- **tuple** → Ordered, unchangeable collection
  Example: coords = (10, 20)

---

3. Set Types

- **set** → Unordered collection of unique items
  Example: s = {1, 2, 3}
- **frozenset** → Immutable set
  Example: fs = frozenset([1, 2, 3])

---

4. Mapping Type

- **dict** → Key-value pairs
  Example: student = {"name": "Gaurav", "age": 20}

---

5. Boolean Type

- **bool** → True or False
  Example: flag = True

---

6. None Type

- **None** → Represents absence of value
  Example: x = None

Input and Output Functions

Input Function

- Used to take user input (always returns a string).

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

print("Hello", name)
```

Output Function

- The print() function is used to display output.

```
print("Welcome to Python")
```

Printing Methods in Python

1. f-string (formatted string literals) – Modern & Recommended

```
name = "Gaurav"

age = 20

print(f"My name is {name} and I am {age} years old.")
```

2. .format() method

```
name = "Gaurav"

age = 20

print("My name is {} and I am {} years old.".format(name, age))
```

Summary

- Datatypes define the type of data a variable holds.

- Python uses input() for user input and print() for output.
- Printing methods:
  f-string → Best modern way
  .format() → Flexible