

Ujjawal Kumar Singh

NIU-24-20782

Noida International University

Python Data Types – Explanation with Examples

Python is a dynamically-typed language, which means you don't need to declare the data type of a variable explicitly. Python automatically assigns the appropriate data type based on the value. The main built-in data types in Python include **Numeric**, **String**, **List**, **Tuple**, **Set**, **Dictionary**, and **Boolean**.

Below are explanations and two examples for each data type:

1. Numeric Data Types

Python supports three types of numeric data:

- **int** (Integer): Whole numbers (positive or negative) without decimal points.
- **float** (Floating Point): Numbers with decimal points.
- **complex**: Numbers with a real and imaginary part (e.g., $3 + 4j$).

Examples:

Example 1: Integer

```
marks = 85
```

Example 2: Float

```
temperature = 36.6
```

These numeric values are commonly used for arithmetic operations, loops, and counters in programming.

2. String Data Type

A string is a sequence of characters enclosed within single quotes ('...'), double quotes ("..."), or triple quotes ('''...' or '"""..."""') for multi-line strings.

Examples:

Example 1: Simple string

```
language = "Python"
```

Example 2: Multi-line string

```
message = """Hello,  
Welcome to Python programming!"""
```

Strings are used for storing text data such as names, messages, or any combination of characters.

3. List Data Type

A list is an ordered and mutable (changeable) collection of items. Lists are defined using square brackets [] and can contain elements of different data types.

✓ Examples:

Example 1: List of numbers

```
numbers = [10, 20, 30, 40]
```

Example 2: Mixed-type list

```
profile = ["Alice", 23, True, 88.5]
```

Lists are widely used for storing related data like student names, scores, or items in a cart.

4. Tuple Data Type

A tuple is an ordered and immutable (unchangeable) collection. Tuples are written using parentheses () and are generally faster than lists.

✓ Examples:

Example 1: Tuple of strings

```
colors = ("red", "green", "blue")
```

Example 2: Mixed-type tuple

```
student = ("John", 21, False)
```

Tuples are useful for fixed data that should not be modified, like coordinates or days of the week.

5. Set Data Type

A set is an unordered collection of unique elements. Sets are defined using curly braces {} and automatically remove duplicate values.

✓ Examples:

Example 1: Set of numbers

```
unique_ids = {101, 102, 103, 101}
```

Example 2: Set of characters

```
vowels = {'a', 'e', 'i', 'o', 'u'}
```

Sets are commonly used for operations involving membership tests, removing duplicates, or performing set operations (union, intersection, etc.).

6. Dictionary Data Type

A dictionary is an unordered collection of key-value pairs. It is defined using curly braces {} where each key is unique and maps to a specific value.

✓ Examples:

Example 1: Basic dictionary

```
student_info = {"name": "Sara", "age": 20}
```

Example 2: Product details

```
product = {"name": "Laptop", "price": 45000}
```

Dictionaries are ideal for representing real-world data such as database records, JSON data, and configuration files.

7. Boolean Data Type

Boolean data represents one of two possible values: **True** or **False**. These are often used in conditional statements and loops.

Examples:

Example 1: Boolean True

```
is_registered = True
```

Example 2: Boolean False

```
has_access = False
```

Booleans are essential for decision-making in programs using if, while, or for statements.