Data Types

int (Integer)

Integer data type stores whole numbers, both positive and negative. It is commonly used for counting and indexing.

Example:

```
age = 25  # Integer
```

float (Floating Point)

Floating-point numbers store decimal values. These are used for scientific calculations and measurements.

Example:

```
height = 5.8  # Float
```

bool (Boolean)

Boolean values store either True or False. This is used for conditions and decision-making.

Example:

```
is_raining = False  # Boolean
```

str (String)

String data type stores text or character data. It is used for labels, names, and categorical values.

Example:

```
name = "Alice"
```

Collection Data Types

list (Ordered, Mutable)

A list stores multiple values in an ordered manner. It is **mutable**, meaning elements can be changed.

Example:

```
scores = [90, 85, 88, 92]
scores.append(95)
print(scores)
```

tuple (Ordered, Immutable)

A tuple is similar to a list but **immutable**, meaning elements cannot be changed after assignment.

Example:

```
coordinates = (10.5, 20.3)
print(coordinates[0])
```

set (Unordered, Unique)

A set stores **unique** values. It is **unordered** and does not allow duplicate elements.

Example:

```
unique_numbers = {1, 2, 3, 3, 4}
print(unique_numbers)
```

dict (Dictionary: Key-Value Pairs)

A dictionary stores data in **key-value pairs**. It allows for fast lookups and structured data storage.

Example:

```
student = {"name": "Alice", "age": 25, "score": 90}
print(student["name"])
```

Data Type	Description	Example
int	Whole numbers	age = 25
float	Decimal numbers	height = 5.8
bool	True/False values	is_raining = False
str	Text data	name = "Alice"
list	Ordered, mutable collection	scores = [90, 85, 88]
tuple	Ordered, immutable collection	coordinates = (10.5, 20.3)
set	Unordered, unique values	unique_numbers = {1, 2, 3}
dict	Key-value pairs	student = {"name": "Alice", "age": 25}

Where Are These Used in Data Science

- **int, float, bool, str** → Used for storing numerical values, labels, and logical conditions in datasets.
- **list, tuple, set, dict** → Used for organizing, filtering, and processing large datasets efficiently.