

## Fundamental Data Types vs Collection Data Types:

In programming, data types classify different types of data that can be used in a program. They define the operations that can be performed on the data and the way the data is stored in memory.

Two broad categories of data types are:

### 1. Fundamental Data Types:

- Fundamental data types represent single values.
- They are basic building blocks and include primitive data types such as integers, floating-point numbers, characters, and boolean values.
- In many programming languages, fundamental data types are also called primitive data types.

#### Here are some examples:

1. **Integer:** Represents whole numbers (e.g., 1, 10, -5).
2. **Float:** Represents floating-point numbers with decimal values (e.g., 3.14, -0.5, 1e3).
3. **Character:** Represents single characters (e.g., 'a', 'X', '@').
4. **Strings :** Represents group or collection of characters (e.g., 'python', 'JSRao' )
5. **Boolean:** Represents true or false values.

Fundamental data types are typically have fixed memory allocations and are immutable, meaning their values cannot be changed after they are created.

### 2. Collection Data Types:

- Collection data types, also known as composite data types or aggregate data types, represent collections or groups of values.
- They allow you to store multiple values in a single variable and are used to manage and manipulate collections of data more efficiently.

#### Collection data types include:

- **Lists:** Ordered collections of items, where each item can be of any data type. Lists are mutable, meaning you can change their elements after they are created.
- **Tuples:** Similar to lists but immutable, meaning their elements cannot be changed after they are created.
- **Sets:** Unordered collections of unique items. Sets do not allow duplicate elements.
- **Dictionaries:** Key-value pairs where each value is associated with a unique key. Dictionaries are mutable and allow you to access, modify, and delete items by their keys.

Collection data types provide various methods and operations for manipulating and accessing their elements, such as indexing, slicing, appending, and removing.

**In summary, fundamental data types represent single values with fixed memory allocations, while collection data types represent collections or groups of values and provide more complex data structures and operations for managing and manipulating collections of data.**