comparison between **List**, **Tuple**, **Set**, and **Dictionary** in Python in **Markdown** format:

**1. List**

A **List** is an ordered, mutable collection that allows duplicate elements.

* **Syntax**: list\_name = [1, 2, 3, 4]
* **Ordered**: The order of elements is preserved.
* **Mutable**: You can modify the elements (add, remove, change).
* **Allows duplicates**: Multiple identical items can be stored in a list.
* **Indexed**: Elements can be accessed by index.
* **Can store heterogeneous data**: Lists can hold items of different types (strings, integers, etc.).

**Example:**

python

Copy

my\_list = [1, 2, 2, 3, 4]

my\_list[0] # Output: 1

**2. Tuple**

A **Tuple** is an ordered, immutable collection that allows duplicate elements.

* **Syntax**: tuple\_name = (1, 2, 3, 4)
* **Ordered**: The order of elements is preserved.
* **Immutable**: Once created, elements cannot be modified (you cannot add, remove, or change items).
* **Allows duplicates**: Multiple identical items can be stored in a tuple.
* **Indexed**: Elements can be accessed by index.
* **Can store heterogeneous data**: Tuples can hold items of different types.

**Example:**

python

Copy

my\_tuple = (1, 2, 3, 3, 4)

my\_tuple[1] # Output: 2

**3. Set**

A **Set** is an unordered collection of unique elements.

* **Syntax**: set\_name = {1, 2, 3, 4}
* **Unordered**: The order of elements is not preserved.
* **Mutable**: You can add or remove elements, but you cannot modify an individual element.
* **No duplicates**: A set automatically removes duplicate values.
* **Not indexed**: Elements cannot be accessed by index.
* **Can store heterogeneous data**: Sets can hold items of different types.

**Example:**

python

Copy

my\_set = {1, 2, 3, 3, 4}

print(my\_set) # Output: {1, 2, 3, 4}

**4. Dictionary**

A **Dictionary** is an unordered collection of key-value pairs, where each key is unique.

* **Syntax**: dict\_name = {"key1": "value1", "key2": "value2"}
* **Unordered**: In Python versions before 3.7, dictionaries were unordered. As of Python 3.7+, dictionaries maintain insertion order.
* **Mutable**: You can modify the values (add, remove, change).
* **No duplicates**: Each key must be unique. If a key repeats, its value is updated.
* **Indexed by key**: You access values by their corresponding keys, not by index.
* **Can store heterogeneous data**: Keys and values can be of any data type.

**Example:**

python

Copy

my\_dict = {"name": "Alice", "age": 25}

my\_dict["name"] # Output: Alice

**Comparison Table:**

| **Feature** | **List** | **Tuple** | **Set** | **Dictionary** |
| --- | --- | --- | --- | --- |
| **Ordered** | Yes | Yes | No | No (Before Python 3.7) |
| **Mutable** | Yes | No | Yes | Yes |
| **Duplicates** | Yes | Yes | No | No (keys must be unique) |
| **Indexed** | Yes | Yes | No | Yes (by key) |
| **Heterogeneous** | Yes | Yes | Yes | Yes |
| **Syntax** | [] | () | {} | {key: value} |

**Summary:**

* **Lists** are ordered, mutable collections that can store duplicates.
* **Tuples** are ordered, immutable collections that can store duplicates.
* **Sets** are unordered collections of unique items and do not support indexing.
* **Dictionaries** store key-value pairs and ensure that keys are unique, and the values are mutable.