

List, Tuple, Dictionary, Sets

Detailed Explanation:

Feature	List	Tuple	Dictionary	Set
Syntax	<code>my_list = [1, 2, 3]</code>	<code>my_tuple = (1, 2, 3)</code>	<code>my_dict = {'a': 1, 'b': 2}</code>	<code>my_set = {1, 2, 3}</code>
Ordered	Yes	Yes	Yes (Python 3.7+)	No
Mutable	Yes	No	Yes	Yes
Allows Duplicates	Yes	Yes	Keys: No, Values: Yes	No
Indexable	Yes	Yes	No	No
Iteration	Yes	Yes	Yes	Yes
Access Time	O(1) for index access	O(1) for index access	O(1) for key access	O(1) for average case
Key-Value Pairs	No	No	Yes	No
Common Methods	<code>append()</code> , <code>extend()</code> , <code>pop()</code>	<code>count()</code> , <code>index()</code>	<code>keys()</code> , <code>values()</code> , <code>items()</code>	<code>add()</code> , <code>remove()</code> , <code>union()</code>
Use Case	Ordered collection of items	Ordered, immutable collection ↓	Key-value pairs	Unordered collection of unique items

1. List:

- **Syntax:** `my_list = [1, 2, 3]`
- **Ordered:** Yes, items maintain the order in which they are added.
- **Mutable:** Yes, items can be changed after creation.
- **Allows Duplicates:** Yes, lists can contain multiple instances of the same value.
- **Indexable:** Yes, items can be accessed via their index.
- **Common Methods:** `append()`, `extend()`, `insert()`, `remove()`, `pop()`, `clear()`, `index()`, `count()`, `sort()`, `reverse()`.
- **Use Case:** Ideal for ordered collections that may need to change.

2. Tuple:

- **Syntax:** `my_tuple = (1, 2, 3)`
- **Ordered:** Yes, items maintain the order in which they are added.
- **Mutable:** No, items cannot be changed after creation.

- **Allows Duplicates:** Yes, tuples can contain multiple instances of the same value.
 - **Indexable:** Yes, items can be accessed via their index.
 - **Common Methods:** `count()`, `index()`.
 - **Use Case:** Ideal for fixed collections of items that should not change.
3. **Dictionary:**
- **Syntax:** `my_dict = {'a': 1, 'b': 2}`
 - **Ordered:** Yes (as of Python 3.7+, insertion order is preserved).
 - **Mutable:** Yes, key-value pairs can be changed after creation.
 - **Allows Duplicates:** Keys: No, each key must be unique. Values: Yes, values can be duplicated.
 - **Indexable:** No, items are accessed via keys, not indices.
 - **Common Methods:** `keys()`, `values()`, `items()`, `get()`, `update()`, `pop()`, `popitem()`, `clear()`.
 - **Use Case:** Ideal for collections of key-value pairs.
4. **Set:**
- **Syntax:** `my_set = {1, 2, 3}`
 - **Ordered:** No, items do not maintain order.
 - **Mutable:** Yes, items can be changed after creation.
 - **Allows Duplicates:** No, each item must be unique.
 - **Indexable:** No, items cannot be accessed via index.
 - **Common Methods:** `add()`, `remove()`, `discard()`, `pop()`, `clear()`, `union()`, `intersection()`, `difference()`.
 - **Use Case:** Ideal for collections of unique items and membership testing.

Summary

- **Lists** are versatile, ordered, and mutable collections, suitable for general use cases.
- **Tuples** are ordered and immutable, useful for fixed data collections.
- **Dictionaries** store key-value pairs, allowing for fast lookups by key.
- **Sets** are unordered collections of unique items, ideal for membership tests and eliminating duplicates.