MAIN DATA STRUCTURES IN PYTHON

🔥 Comprehensive Comparison Table - Python Data Structures

Feature	**List**	**Tuple**	**Set**	**Dictionary**
Definition	Ordered, mutable collection of items.	Ordered, immutable sequence of items.	Unordered collection of unique items.	Unordered collection of key-value pairs.
Syntax Symbol	Square Brackets []	Parentheses ()	Curly Braces {}	Curly Braces {} with key-value pairs
Ordered/Unordered	Ordered	Ordered	Unordered	Unordered
Mutable/Immutable	Mutable	Immutable	Mutable	Mutable
Allows Duplicates	Yes	Yes	No	Keys - No, Values - Yes
Indexing Supported?	Yes (0-based)	Yes (0-based)	No (no indexing)	By Key
Best Use Case	Sequential data, where order and flexibility matter	Fixed data (coordinates, config)	Unique collection of items (no duplicates)	Mapping relationships (name to number)
Where Not Suitable	When keys are needed	When modification is needed	When ordering is crucial	When no key-value relation exists
Unique Advantage	Flexible, supports all operations	Memory efficient, faster in loops	Automatically removes duplicates	Super-fast lookups by keys
Key Disadvantage	Slower than tuple	Cannot modify	No order, no duplicates allowed	Slightly more memory-intensive
Duplicate Handling	Allows all	Allows all	Removes duplicates	Keys must be unique
Symbol Confusion	Same as arrays	Same as function args	Same as dict (empty set = set())	Same as set (but with colons)
Crucial Property	Flexible & ordered	Fixed & ordered	Unordered & unique	Key-Value pairs

MAIN DATA STRUCTURES IN PYTHON

Feature	**List**	**Tuple**	**Set**	**Dictionary**
Key Access	By index	By index	No keys	By key
Search Speed	Linear (O(n))	Linear (O(n))	Fast (O(1))	Very Fast (O(1))
Modification	Add, remove, replace	Not possible	Add, remove	Add, remove, update

---

📚 Full Definitions & Code Examples

---

1) LIST:

Full Definition

A **List** is an **ordered**, **mutable** collection that allows duplicate elements. Lists are ideal when you need to store **a sequence of items where order matters and you might need to modify the items**.

Example

```
fruits = ['apple', 'banana', 'cherry']


# Add

fruits.append('orange')


# Modify

fruits[1] = 'mango'


# Remove

fruits.remove('cherry')


print(fruits)
```

MAIN DATA STRUCTURES IN PYTHON

2) TUPLE:

Full Definition

A **Tuple** is an **ordered**, **immutable** collection that allows duplicates. Tuples are ideal when you want to store **a fixed collection of items that should not be modified**, such as coordinates or settings.

Example

```
coordinates = (10, 20)


# Accessing

print(coordinates[0])


# Cannot modify

# coordinates[0] = 15  # This will raise an error
```

---

3) SET:

Full Definition

A **Set** is an **unordered**, **mutable** collection of **unique** items. Sets are best used when you need to **store items without duplicates and order doesn’t matter**.

Example

```
unique_numbers = {1, 2, 3, 2, 1}


# Add

unique_numbers.add(4)


# Remove

unique_numbers.remove(2)


print(unique_numbers)
```

MAIN DATA STRUCTURES IN PYTHON

4) DICTIONARY:

Full Definition

A **Dictionary** is an **unordered**, **mutable** collection of **key-value pairs**, where keys are unique and values can be any data type. Dictionaries are ideal for **mapping relationships between keys and values, such as names to phone numbers**.

Example

```
student_scores = {
  "Alice": 95,
  "Bob": 89
}


# Add
student_scores["Charlie"] = 92


# Update
student_scores["Alice"] = 98


# Remove
del student_scores["Bob"]


print(student_scores)
```

✅ **Bonus — Quick Access Summary Table**

Data Structure	**Key Properties**	**Real-World Example**
List	Ordered, mutable, allows duplicates	Grocery list
Tuple	Ordered, immutable, allows duplicates	GPS coordinates
Set	Unordered, unique items	Set of unique email addresses
Dictionary	Unordered, key-value pairs	Phone book (name to number)

MAIN DATA STRUCTURES IN PYTHON

🚀 Problem-Solving Tip for Each

Data Structure	Example Problem Use Case
List	Store student marks & sort
Tuple	Fixed database record (ID, Name, Date)
Set	Find common attendees between two events
Dictionary	Count word frequency in a document

✍️ Summary in One Line

Data Structure	One-Line Summary
List	Flexible ordered collection of items
Tuple	Fixed ordered collection of items
Set	Unordered collection of unique items
Dictionary	Key-value store for fast lookups