Python List vs Tuple - Complete Guide

Definition

List and Tuple are Python data structures used to store multiple items.

- List: [1, 2, 3] -> Mutable (can change)
- Tuple: (1, 2, 3) -> Immutable (cannot change)

Comparison Table

Feature	List	Tuple
Syntax	[1, 2, 3]	(1, 2, 3)
Mutable	Yes	No
Memory usage	Higher	Lower
Speed	Slower	Faster
Methods	Many (append, pop)	Few (count, index)
Dict Key Usage	No	Yes

Mutability Example

List (Mutable):

```
my_list = [1, 2, 3]
my_list.append(4)  # Works
```

Tuple (Immutable):

```
my_tuple = (1, 2, 3)
my_tuple[0] = 10   # Error!
```

Why Tuple is Faster

Python List vs Tuple - Complete Guide

Tuples are faster because:

- Fixed size (no dynamic resizing)
- Fewer operations to support
- More memory efficient

Tuples help optimize performance where data is fixed.

Memory Check Example

```
import sys

a = [1, 2, 3]
b = (1, 2, 3)

print(sys.getsizeof(a))  # List memory
print(sys.getsizeof(b))  # Tuple memory
```

When to Use What

Use List:

- When you need to change, add, or remove items
- For dynamic data (e.g., user input)

Use Tuple:

- When data is fixed (coordinates, DB rows)
- As dictionary keys (immutable & hashable)

Overwriting Concept

Python List vs Tuple - Complete Guide

```
a = [1, 2, 3]
a = (1, 2, 3)  # List is overwritten by tuple

print(type(a))  # Output: <class 'tuple'>
```

Summary

- List: Mutable, slower, uses more memory
- Tuple: Immutable, faster, uses less memory
- Use list for dynamic data, tuple for fixed data
- Variable can be overwritten if reused