Difference between Lists and Tuples in Python

Feature	List	Tuple
Mutability	Mutable - You can modify (add, remove, or change elements).	Immutable - Once created, elements cannot be modified.
Syntax	Defined with square brackets: my_list = [1, 2, 3]	Defined with parentheses: my_tuple = (1, 2, 3) (or just commas: 1, 2, 3)
Methods available	Many methods: .append(), .remove(), .sort(), .reverse(), etc.	Very few methods: .count() and .index() only.
Memory usage	Uses more memory (because of mutability).	Uses less memory (because it's immutable and lightweight).
Performance (speed)	Slower than tuples (because mutable data structures require extra handling).	Faster in iteration and lookup (Python can optimize tuples better).
Use cases (when to use)	When you need a collection of items that can change - e.g., a list of students in a class.	When you need a fixed collection of items - e.g., coordinates (x, y), RGB colors, or data that shouldn't change.
Hashable (dict key?)	No (lists are unhashable).	Yes, if it contains only immutable elements - tuples can be dictionary keys or set elements.
Element access	Same in both: via indexing and slicing - my_list[0], my_tuple[1:3]	Same as list.

Quick summary rule:

Use list when you need something that changes.

Use tuple when the data should stay constant and be faster or hashable.