This module gives overview of more sophisticated Python collections, with focus on operations, memory models, and coding styles.

**Previous puzzles**

Collection Operations

List Methods: index, count, copy, clear, pop, remove, append, extend, insert, reverse, sort.

Set Methods: add, clear, remove, discard, pop, issubset, difference.

Dict Methods: get, pop, update, clear, copy.

**Flexible Function Arguments**: \*args (variable positional args) and \*\*kwargs (variable keyword args).

**Converting Between Iterables**: tuple(), list(), set().

**Looping Techniques**: Looping through elements or indices of strings, lists, tuples, sets, and dictionaries.

**Memory Model**

Assignment vs. Copy:

Assignment (b = a) shares the same memory address.

Copy (b = a.copy()) creates a new object.

**Mutable vs. Immutable Objects**:

Lists and dictionaries (mutable) can be modified in functions, which can affect the original.

Tuples (immutable) cannot be modified except by reassigning them.

**Coding Style**

Do not mix data types in collections for readability and error prevention.

Use list comprehensions (e.g., [i\*i for i in range(50)]) where possible for conciseness.