Class 5 Summary: Strings, Lists, Dictionaries, and Sets

This class focused on understanding Python data types through the lens of **state, transitions, and invariants**. Rather than memorizing methods, students were encouraged to reason about what changes, what stays the same, and what assumptions Python enforces.

Core Idea

Every bug is a violated invariant. Programming becomes easier when you can clearly identify the current state of an object, the transition caused by an operation, and the invariants that must always hold.

Strings

- Strings are **immutable**.
- All string methods return **new strings**.
- Indexing reads state; slicing creates new state.
- Methods like replace, split, and join never mutate.

Lists

- Lists are **mutable** and ordered.
- Most mutating list methods return **None**.
- append, extend, insert, remove, and pop change state.
- Mutating a list while relying on indices can break invariants.

Dictionaries

- Dictionaries map **keys** $\rightarrow$ **values** using hashing.
- Keys must be **immutable (hashable)**.
- Methods like update, pop, and clear mutate.
- The | operator creates a **new** merged dictionary.

Sets

- Sets are mutable, unordered collections of **unique, hashable** elements.
- No duplicates are allowed.
- add, remove, discard, and pop mutate the set.
- pop removes an **arbitrary** element.

Rule of Thumb

If a method changes an object's state, it usually returns **None**. If it returns a value, assume a new object was created unless stated otherwise.