Chapter 2: Data Structures

Lesson 1: Lists

- **Definition**: A list is an ordered, mutable collection of items in Python, defined with square brackets `[]` (e.g., `my_list = [1, "hello", 3.14]`).
- **Creating Lists**: Use `[]` for direct creation, or `list()` constructor (e.g., `list("hello")` → `['h', 'e', 'l', 'l', 'o']`).
- **Key Properties**: Indexed (starts at 0), allows duplicates.
- **Accessing Elements**:
  - Positive indexing: `list[0]` (first item).
  - Negative indexing: `list[-1]` (last item).
  - Slicing: `list[start:stop:step]` (e.g., `[0, 1, 2, 3][1:3]` → `[1, 2]`).
- **Modifying Lists**:
  - Change: `list[1] = "new"`.
  - Add: `append()`, `insert(index, item)`.
  - Remove: `remove(item)`, `pop(index)`, `clear()`.
- **Common Methods**:
  - `len(list)`: Length.
  - `count(item)`: Count occurrences.
  - `index(item)`: First index of item.
  - `sort()`: Sort ascending.
  - `reverse()`: Reverse order.
- **Nested Lists**: Lists within lists (e.g., `[[1, 2], [3, 4]]` → `matrix[0][1]` = `2`).
- **Operations**:
  - Concatenation: `list1 + list2`.
  - Repetition: `list * n`.
  - Membership: `item in list`.
- **Practical Example**: Managing a to-do list with add/remove/sort operations.