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.

Exercises for Practice

1. Write a program that prints the second color from the list below:

```
colors = ["Red", "Green", "Blue", "Yellow"]
```

2. Modify the list below to change "Cat" to "Tiger", then print the updated list.

```
animals = ["Dog", "Cat", "Elephant"]
```

3. Given the list:

```
sequence = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- Use slicing to extract the elements [3, 4, 5, 6]
- Use slicing to create a list with every third element starting from index 0 (i.e., [0, 3, 6, 9]).
- Print both results.

4. Given the nested list:

```
grid = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

- Print the number 8 (row 2, column 1).
- Replace the number 5 (row 1, column 1) with 50.
- Print the updated grid.

5. Add and Remove Items:
   Start with a list:

```
numbers = [10, 20, 30]
```

- Add 40 to the list.
- Remove 20 from the list.
- Print the final list.

6. Create a list of 5 numbers and sort them in descending order.
7. Write a program that sorts the list below in ascending order and prints it:

```
ages = [25, 19, 30, 22, 18]
```

8. Write a program that combines these two lists into one and prints the result:

```
list1 = ["A", "B", "C"]
list2 = [1, 2, 3]
```
