7. Working with Dictionaries

- **Iterating over a dictionary using loops.**

Iterating over a dictionary using loops is a fundamental concept in Python and other programming languages. Below is a theoretical explanation of how it's done, focusing on Python:

Why Iterate Over a Dictionary?

- Access all keys or values
- Modify items
- Search or filter data
- Perform operations based on conditions

Ways to Iterate Over a Dictionary in Python

1. Iterating Over Keys:

```
for key in my_dict:
    # do something with key
    print(key)
```

2. **Iterating Over Values**

```
for value in my_dict.values():
    # do something with value
    print(value)
```

- **Merging two lists into a dictionary using loops or zip().**

Merging two lists into a dictionary is a common task in Python. This involves using one list for keys and another for values. There are two main ways to do this

1. **Using zip()**

The zip() function is used to combine two or more iterables (like lists, tuples) into pairs.

```
for a, b in zip(list1, list2):
    # do something with a and b
    print(a, b)
```

How It Works:

- The zip() function pairs elements from two lists into tuples.
- The dict() function then converts these tuples into key-value pairs.

2. Using Loop

```
for x, y in zip(sequence1, sequence2):
    # use x and y
    print(x, y)
```

- **Counting occurrences of characters in a string using dictionaries.**

Using a dictionary is one of the most efficient ways to count how many times each character appears in a string.

How It Works:

1. **Initialize an empty dictionary.**
    - This will store characters as keys and their counts as values.
2. **Iterate through each character in the string.**
3. **For each character:**
    - Check if the character already exists in the dictionary.
        - If it does, increment its count.
        - If it doesn't, add it to the dictionary with a count of 1.

Key Concepts:

- Dictionaries allow fast lookups and updates, which makes them efficient for frequency counting.
- The logic is based on conditional checking:
    - Whether a character has been seen before (exists as a key).
- This method works for any string, regardless of its length or content.