**Dictionaries**

A **dictionary** in Python is a collection of **key-value pairs**, where each key is unique, and it maps to a corresponding value. It’s similar to a real-life dictionary, where you look up a word (key) to find its meaning (value).

**What is a Python Dictionary?**

* **Unordered**: Dictionaries don’t maintain order (in Python 3.6+, they are ordered).
* **Mutable**: You can modify, add, or remove items.
* **Key-Value Pairs**: Each item is a pair of a **key** and its **value**.
* **Keys are unique** and must be immutable (strings, numbers, or tuples).
* **Values can be any data type** (int, float, list, dictionary, etc.).

### ****Common Methods****

| **Method** | **Description** | **Example** |
| --- | --- | --- |
| dict.get(key) | Returns the value of the specified key | person.get("name") |
| dict.keys() | Returns a list of all keys | person.keys() |
| dict.values() | Returns a list of all values | person.values() |
| dict.items() | Returns a list of key-value pairs | person.items() |
| dict.update() | Updates the dictionary with another dict | person.update({"city": "LA"}) |
| dict.pop(key) | Removes and returns the value of the key | person.pop("name") |
| dict.clear() | Removes all elements from the dictionary | person.clear() |

**Summary**

* A **dictionary** is a powerful and flexible data structure for mapping keys to values.
* It’s ideal for use cases like counting occurrences, grouping data, and fast lookups.
* **Nested dictionaries** allow you to represent complex data structures.