**What is a Dictionary?**

A dictionary in Python is a **collection of key-value pairs**.

* Each key is **unique** and **immutable** (cannot be changed).
* Values can be any data type and can be duplicated.
* Dictionaries are **mutable**, meaning we can change, add, or remove items.

**Syntax:**

my\_dict = {

"name": "Talha",

"age": 22,

"city": "Lahore"

}

**2. Accessing Values**

You can access values using keys.

python

CopyEdit

print(my\_dict["name"]) # Talha

print(my\_dict.get("age")) # 22 (doesn't throw error if key missing)

**3. Adding or Updating Items**

python

CopyEdit

my\_dict["email"] = "talha@example.com" # Adds new key-value

my\_dict["age"] = 23 # Updates existing key

print(my\_dict)

**4. Removing Items**

my\_dict.pop("age") # Removes 'age' key

my\_dict.popitem() # Removes the last inserted item

del my\_dict["city"] # Removes 'city' key

my\_dict.clear() # Clears all items

**5. Iterating Through Dictionary**

for key in my\_dict:

print(key, my\_dict[key])

for value in my\_dict.values():

print(value)

for key, value in my\_dict.items():

print(key, value)

**6. Dictionary Built-in Functions**

**a) len()**

Returns the number of items:

print(len(my\_dict))

**b) keys()**

Returns all keys:

print(my\_dict.keys()) # dict\_keys(['name', 'age'])

**c) values()**

Returns all values:

print(my\_dict.values())

**d) items()**

Returns key-value pairs:

print(my\_dict.items())

**e) update()**

Adds another dictionary or key-value pairs:

my\_dict.update({"age": 25, "city": "Karachi"})

print(my\_dict)

**f) copy()**

Returns a shallow copy:

new\_dict = my\_dict.copy()

**g) fromkeys()**

Creates a dictionary from keys with the same value:

keys = ["a", "b", "c"]

default\_value = 0

new\_dict = dict.fromkeys(keys, default\_value)

print(new\_dict) # {'a': 0, 'b': 0, 'c': 0}

**h) setdefault()**

Returns the value if the key exists, otherwise adds the key with a default value:

print(my\_dict.setdefault("name", "Unknown"))

print(my\_dict.setdefault("country", "Pakistan"))

print(my\_dict)

**7. Nesting Dictionaries**

Dictionaries can contain other dictionaries:

students = {

"student1": {"name": "Ali", "age": 20},

"student2": {"name": "Talha", "age": 22}

}

print(students["student1"]["name"]) # Ali

**8. Dictionary Comprehension**

squares = {x: x\*x for x in range(5)}

print(squares) # {0: 0, 1: 1, 2: 4, 3: 9, 4: 16}

**9. Why Use Dictionaries?**

* Fast **lookup** of data using keys.
* Flexible and can store various data types.
* Useful for storing structured data (like JSON).