Dictionaries in Python

## Definition

A dictionary in Python is an unordered, mutable collection of key-value pairs where each key is unique. It is used to store data values like a map, where keys are unique identifiers for corresponding values.

## Characteristics of Dictionaries

- Unordered (Before Python 3.7): No guaranteed order of elements.  
- Mutable: Items can be changed, added, or removed.  
- Key-Value Pairs: Data is stored as key-value pairs.  
- Keys must be hashable (immutable types like strings, numbers, tuples).  
- Values can be of any data type.

## Properties of Dictionaries

- Defined using curly braces {} or the dict() constructor.  
- Keys are unique within a dictionary.  
- Accessing values is done via keys.

## Dictionary Operations & Methods

## Creation

# Creating dictionaries  
person = {"name": "Alice", "age": 25, "profession": "Engineer"}  
empty\_dict = {}  
another\_dict = dict(name="Bob", age=30)

## Accessing Values

## person = {"name": "Alice", "age": 25} print(person["name"]) # Output: Alice print(person.get("age")) # Output: 25

## Adding or Updating Key-Value Pairs

person = {"name": "Alice", "age": 25}  
person["profession"] = "Engineer" # Adding  
person["age"] = 26 # Updating  
print(person)  
# Output: {'name': 'Alice', 'age': 26, 'profession': 'Engineer'}

## Removing Elements

person = {"name": "Alice", "age": 25, "profession": "Engineer"}  
person.pop("age") # Removes key 'age'  
del person["profession"] # Removes key 'profession'  
print(person) # Output: {'name': 'Alice'}

## Iterating Through Dictionary

person = {"name": "Alice", "age": 25}  
for key in person:  
 print(key, ":", person[key])

## Dictionary Methods

- keys(): Returns all keys  
- values(): Returns all values  
- items(): Returns key-value pairs  
- clear(): Removes all items  
- copy(): Returns a shallow copy of the dictionary  
- popitem(): Removes and returns the last inserted key-value pair

## Dictionary Length

person = {"name": "Alice", "age": 25}  
print(len(person)) # Output: 2

## Example Program: Dictionary Operations

# Example Program to demonstrate dictionary operations  
  
# Creating a dictionary  
person = {"name": "Alice", "age": 25, "profession": "Engineer"}  
  
# Accessing values  
print("Name:", person["name"])  
print("Age:", person.get("age"))  
  
# Adding or updating key-value pair  
person["city"] = "New York"  
person["age"] = 26  
  
# Removing an element  
person.pop("profession")  
  
# Iterating through the dictionary  
print("\nDictionary contents:")  
for key, value in person.items():  
 print(f"{key} : {value}")  
  
# Using dictionary methods  
print("\nKeys:", person.keys())  
print("Values:", person.values())  
print("Items:", person.items())  
  
# Length of the dictionary  
print("\nNumber of items:", len(person))