Python Dictionary - Complete Explanation

A dictionary in Python is an unordered, mutable collection of key-value pairs.
Each key must be unique and immutable, while values can be of any type.

Basic Syntax:
my_dict = {'name': 'Gaurav', 'age': 21}

Key Features:
• Mutable and dynamic
• Unordered (but maintains insertion order in Python 3.7+)
• Accessed using keys instead of indexes
• Keys must be unique

Creation Methods:
1. Using {}: my_dict = {'a': 1, 'b': 2}
2. Using dict(): my_dict = dict(a=1, b=2)
3. Empty dict: my_dict = {}

Accessing Data:
print(my_dict['a'])
print(my_dict.get('b', 'default'))

Adding & Updating:
my_dict['c'] = 3
my_dict['a'] = 10

Removing:
pop(), popitem(), del, clear()

Looping:
for k, v in my_dict.items():
print(k, v)

Common Methods:
keys(), values(), items(), get(), update(), pop(), copy()

Dictionary Comprehension:
squares = {x: x**2 for x in range(5)}

Nested Dictionaries:
students = {'101': {'name': 'Gaurav', 'branch': 'CSE'}}

Merging Dictionaries (Python 3.9+):
merged = dict1 | dict2

Word Frequency Example:
text = 'python is fun and python is powerful'
freq = {}
for word in text.split():
freq[word] = freq.get(word, 0) + 1

Summary:
Create: d = {'a': 1}
Access: d['a']
Safe Access: d.get('a', 0)
Add/Update: d['b'] = 2
Delete: del d['a']
Loop: for k, v in d.items():
Keys: d.keys()

Values: d.values()
Copy: d.copy()
Merge: d1 | d2