

**Course & Test Series**

-  **CBSE**
-  **ICSE**
-  **NTSE**

-  **Banking & Insurance**
-  **Central Govt. Service**

-  **State Govt. Services**

-  **LAW Entrance**

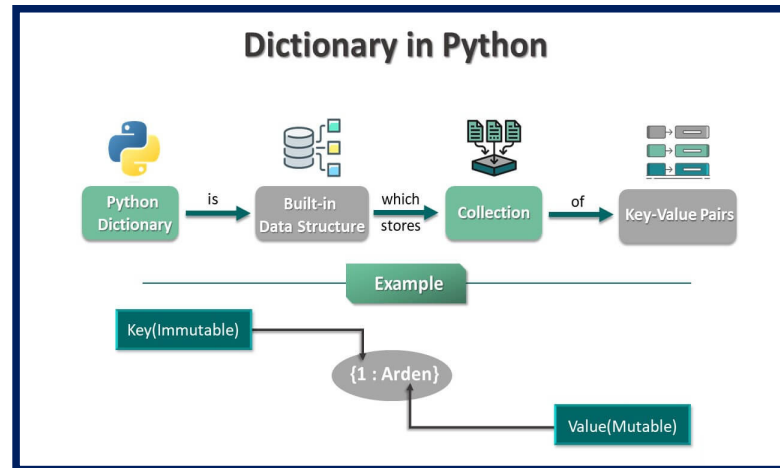
-  **MBA Entrance**

-  **Railways & Metro Services**

**...many more**  
**abhyasonline.in**

**Dictionary in Python**

In Python, a dictionary is a built-in data type that stores key-value pairs. It's used to organize and access data efficiently by keys rather than by index (like in lists).



**Syntax:**

```
dictionary_name = {
    key1: value1,
    key2: value2,
    key3: value3
}
```

**Example:**

```
my_dict = {
    "name": "Alice",
    "age": 25,
    "city": "New York"
}
```

- Keys are unique
- Values can be of any type
- Keys must be immutable types (like strings, numbers, tuples)

**Why do we use Dictionary in Python?**

- Key-value pairs: Store data in pairs for fast and direct access using keys instead of indices.
- Descriptive keys: Keys can be meaningful (like 'name', 'age'), making code more readable and self-documenting.
- Mutable: You can change, add, or remove key-value pairs after the dictionary is created.

Course  
&  
Test Series



CBSE



ICSE



NTSE



Banking &  
Insurance



Central Govt.  
Service



State Govt.  
Services



LAW  
Entrance



MBA  
Entrance



Railways & Metro  
Services

...many more

abhyasonline.in

Dictionary in Python

- Flexible data structure: Supports storing any data type as values – including lists, other dictionaries, etc.
- Useful for real-world data modeling: Ideal for representing JSON-like data structures or objects.
- Easy data mapping: Efficient for implementing maps, hash tables, symbol tables, etc.
- Foundation for other data structures: Used heavily in implementations of objects, databases, caches, etc.

Solved Example 1: Basic Dictionary

```
student = {  
    "name": "Ravi",  
    "age": 16,  
    "class": "12th",  
    "marks": 88  
}
```

```
print(student)
```

Output:

```
{'name': 'Ravi', 'age': 16, 'class': '12th', 'marks': 88}
```

Explanation:

Each piece of data (like *name*, *age*) is stored as a **key-value pair**.

Solved Example 2: Accessing Values

```
print(student["name"])  
print(student["marks"])
```

Output:

```
Ravi  
88
```

Explanation:

Use the **key name inside square brackets []** to get its value.

Solved Example 3: Adding or Updating Items

```
student["school"] = "Abhyas School" # Add a new key-value pair  
student["marks"] = 92 # Update marks  
print(student)
```

Output:

```
{'name': 'Ravi', 'age': 16, 'class': '12th', 'marks': 92, 'school': 'Abhyas School'}
```



Course  
&  
Test Series

Dictionary in Python

 CBSE

**Explanation:**

We can easily add or update entries in the dictionary using assignment (=).

**Solved Example 4: Removing Items**

```
student.pop("age") # Removes the key 'age'  
print(student)
```

**Output:**

```
{'name': 'Ravi', 'class': '12th', 'marks': 92, 'school': 'Abhyas School'}
```

 ICSE

**Explanation:**

pop() removes the specified key and its value.

 NTSE

**Solved Example 5: Looping Through a Dictionary**

```
for key, value in student.items():  
    print(key, ":", value)
```

**Output:**

```
name : Ravi  
class : 12th  
marks : 92  
school : Abhyas School
```

 Banking & Insurance

 Central Govt. Service

**Explanation:**

The .items() method returns all key-value pairs, which can be looped through easily.

 State Govt. Services

**Solved Example 6: Dictionary Example in Real Life**

```
phonebook = {  
    "Raghav": 9876543210,  
    "Ravi": 9123456789,  
    "Sita": 9988776655  
}  
print("Ravi's Number:", phonebook["Ravi"])
```

**Output:**

```
Ravi's Number: 9123456789
```

 LAW Entrance

 MBA Entrance

**Explanation:**

Just like a phonebook – names are keys, phone numbers are values.

 Railways & Metro Services

...many more

abhyasonline.in

Common Methods of Dictionary

Method	Description	Example
dict.get(key)	Returns the value for the key, or None if not found	d.get("name")
dict.keys()	Returns a view of all keys	d.keys()
dict.values()	Returns a view of all values	d.values()
dict.items()	Returns a view of key-value pairs	d.items()
dict.update()	Updates the dictionary with key-value pairs from another	d.update({"age": 30})
dict.pop(key)	Removes the item with the specified key	d.pop("name")
dict.popitem()	Removes and returns the last inserted key-value pair	d.popitem()
dict.clear()	Removes all items from the dictionary	d.clear()

Assignment

Ques 1: Create a dictionary called employee with the following keys and values: "name", "designation", "salary", "department".

- Print all the keys using .keys()
- Print all the values using .values()

Ques 2: Store 3 students and their total marks in a dictionary.

- Then use a for loop to print the name and marks of each student in this format:  
Ravi : 85

 CBSE

 ICSE

 NTSE

 Banking & Insurance

 Central Govt. Service

 State Govt. Services

 LAW Entrance

 MBA Entrance

 Railways & Metro Services

...many more

abhyasonline.in