**DAY 6** **[01 - JULY - 2025]**

# DICTIONARY IN PYTHON

Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

Dictionaries are written with curly brackets, and have keys and values.

*Dictionary Items*

Dictionary items are ordered, changeable, and do not allow duplicates.

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

*Changeable*

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

*Duplicates Not Allowed*

Dictionaries cannot have two items with the same key.

*Dictionary Length*

To determine how many items a dictionary has, use the len() function:

*Dictionary Items - Data Types*

The values in dictionary items can be of any data type.

*Get Keys*

The keys() method will return a list of all the keys in the dictionary.

*Get Values*

The values() method will return a list of all the values in the dictionary.

*Change Values*

You can change the value of a specific item by referring to its key name:

### *Update Dictionary*

The update() method will update the dictionary with the items from the given argument.

The argument must be a dictionary, or an iterable object with key:value pairs.

### *Adding Items*

Adding an item to the dictionary is done by using a new index key and assigning a value to it.

### *Removing Items*

There are several methods to remove items from a dictionary.

### *Access Items in Nested Dictionaries*

To access items from a nested dictionary, you use the name of the dictionaries, starting with the outer dictionary.

## *PRACTICE QUESTIONS:*

1. Basic Dictionary Operations:

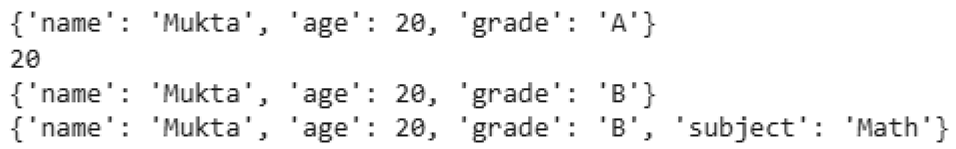
```
# Create a dictionary named student with keys "name," "age," and
"grade," and assign appropriate values.
student = {
    "name": "Mukta",
    "age": 20,
    "grade": "A"
}
print(student)

# Print the value associated with the "age" key.
print(student["age"])

#Update the "grade" to a new value
student["grade"] = "B"
print(student)

#.Add a new key-value pair for "subject" and its corresponding value.
student["subject"] = "Math"
print(student)
```

Output:



```
{'name': 'Mukta', 'age': 20, 'grade': 'A'}
20
{'name': 'Mukta', 'age': 20, 'grade': 'B'}
{'name': 'Mukta', 'age': 20, 'grade': 'B', 'subject': 'Math'}
```

2. Dictionary Manipulation:

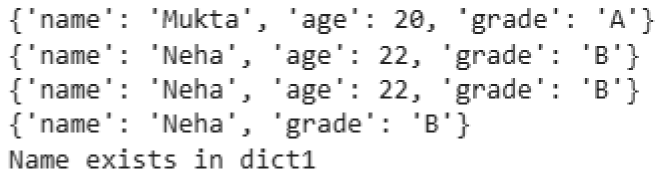
```
# Create two dictionaries, dict1 and dict2, with at least three key-
value pairs each.
dict1 = {
    "name": "Mukta",
    "age": 20,
    "grade": "A"
}
dict2 = {
    "name": "Neha",
    "age": 22,
    "grade": "B"
}
print(dict1)
print(dict2)

#Merge these dictionaries into a new dictionary called merged_dict.
merged_dict = {**dict1, **dict2}
print(merged_dict)

#Remove a key from merged_dict.
del merged_dict["age"]
print(merged_dict)

#Check if a specific key exists in dict1
if "name" in dict1:
    print("Name exists in dict1")
else:
    print("Name does not exist in dict1")
```

Output:



```
{'name': 'Mukta', 'age': 20, 'grade': 'A'}
{'name': 'Neha', 'age': 22, 'grade': 'B'}
{'name': 'Neha', 'age': 22, 'grade': 'B'}
{'name': 'Neha', 'grade': 'B'}
Name exists in dict1
```

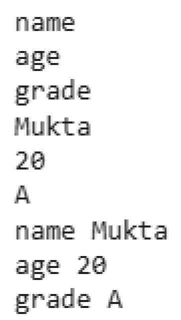
3. Iterating Through a Dictionary:

```
 #Use a loop to print all keys in the student dictionary.
student = {
    "name": "Mukta",
    "age": 20,
    "grade": "A"
}
for key in student:
    print(key)

 #Use another loop to print all values in the student dictionary.
 for value in student.values():
    print(value)

 #Write a loop to print each key-value pair in the student dictionary
 for key, value in student.items():
    print(key, value)
```

Output:



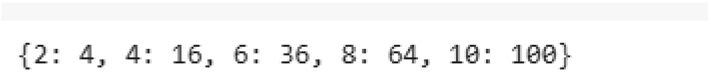
```
name
age
grade
Mukta
20
A
name Mukta
age 20
grade A
```

4. Dictionary Comprehension:

```
 #Create a dictionary comprehension to generate a dictionary of squares
from 1 to 10.
 dict = {num: num**2 for num in range(1, 11)}

 #Filter the above dictionary to include only even squares.
 dict = {num: num**2 for num in range(1, 11) if num**2 % 2 == 0}
 print(dict)
```

Output:



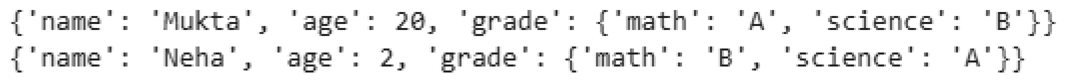
```
{2: 4, 4: 16, 6: 36, 8: 64, 10: 100}
```

5. Nested Dictionaries:

```
#Create a dictionary named school with multiple students (nested
dictionaries).
school = {
    "student1": {
        "name": "Mukta",
        "age": 20,
        "grade": {
            "math": "A",
            "science": "B"
        }
    },
    "student2": {
        "name": "Neha",
        "age": 2,
        "grade": {
            "math": "B",
            "science": "A"
        }
    }
}

#Access and print information about a specific student within the
school dictionary.
print(school["student1"])
print(school["student2"])
```

Output:



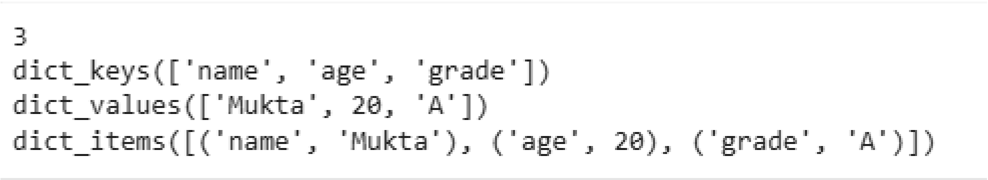
```
{'name': 'Mukta', 'age': 20, 'grade': {'math': 'A', 'science': 'B'}}
{'name': 'Neha', 'age': 2, 'grade': {'math': 'B', 'science': 'A'}}
```

6. Dictionary Functions:

```
#Use the len() function to find the number of items in a dictionary.
student = {
    "name": "Mukta",
    "age": 20,
    "grade": "A"
}
print(len(student))

#Use the keys(), values(), and items() methods on a dictionary and
print the results
print(student.keys())
print(student.values())
print(student.items())
```

Output:

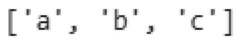


```
3
dict_keys(['name', 'age', 'grade'])
dict_values(['Mukta', 20, 'A'])
dict_items([('name', 'Mukta'), ('age', 20), ('grade', 'A')])
```

7. Sorting a Dictionary:

```
#Create a dictionary with unsorted keys.
unsorted_dict = {
    "c": 3,
    "a": 1,
    "b": 2
}
#Use the sorted() function to print the keys in alphabetical order.
print(sorted(unsorted_dict))
```

Output:



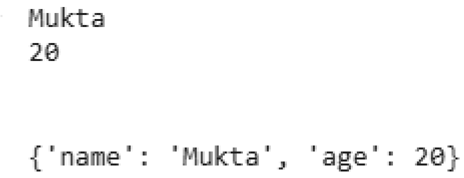
```
['a', 'b', 'c']
```

8 . Dictionary Methods:

```
# Use the get() method to retrieve a value with a default if the key
doesn't exist.
student = {
    "name": "Mukta",
    "age": 20,
    "grade": ""
}
print(student.get("name"))
print(student.get("age"))
print(student.get("grade"))

#Use the pop() method to remove and return a specific key-value pair.
print(student.pop("grade"))
print(student)
```

Output:



```
Mukta
20


{'name': 'Mukta', 'age': 20}
```