Date
Page

Day 6 - Dictionaries & Data Structures

Q What is Dictionary?

→ A dictionary is unordered, mutable collection that stores data in key value pairs. In simple words, insted of using index numbers, dictionaries use keys to store and retrieve values.

- Dictionary Creation:

Syntax:

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

eg.

```
student = {
        "name" : "Piyush",
        "age" : 19,
        "Course" : "Engineering"
    }
```

- Rules for keys

- Must be unique
- Immutable (string, number, tuple)
- Lists cannot be keys.

Date
Page

- Accessing Dictionary Values

- Using key Name

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

- Using get() (Safe Method)

```
print(student.get("age"))
```

→ Note : Key difference in dict[key] and dict.get()

- dict[key] → error if key not found
- dict.get(key) → no error, safer if key is not there.

- Updating & Adding Values

→ Updating existing value

```
student["age"] = 20    # Assigning New value to existing key.
```

→ Adding New Key-Value Pair

```
student["city"] = "Mumbai"
```

- Removing Values

```
student.pop("age")      # Removing value of key "age"
del student["course"]
```

→ Dictionary Methods.

Method	Purpose
keys()	Returns all keys
values()	Returns all values
items()	Returns key-value pairs.
update()	Update multiple values
clear()	Remove all items
copy()	Copy dictionary.

Example:

```
print (student.keys())
print (student.values())
```

→ Looping through dictionaries

→ Loop through keys.	→ Loop through values

For key in dict:
  print(key)
	For value in dict.values():
	  print(value)

→ Loop through Key-value pairs

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

This is for iterating over both key & values.

→ Nested Dictionaries.

A nested dictionary is a dictionary inside another dictionary.

Example:

```
Students = {
    "student1": {
        "name": "Piyush",
        "marks": 85
    },
    "Student2": {
        "name": "Amit",
        "marks": 90
    }
}
```

- Accessing Nested Values

```
print(students["student1"]["marks"])
```

Note: This means value of marks from student 1 of student dict.

Date
Page

→ Practical Use Case - Introduction to JSON.

JSON - JavaScript Object Notation

JSON is a data format used to store and exchange data.

• Python Dictionary vs JSON.

Both looks almost same:

→ Python:

```
data = {
        "name" : "Piyush",
        "age" : 19
    }
```

→ JSON:

```
{
  "name" : "Piyush",
  "age" : 19
}
```

→ Why JSON is important

- Universal, used in APIs
- Used in databases, for example: MongoDb atlas.
- Used in web & mobile app.

- Convert dictionary to JSON.

```
import json

json_data = json.dumps(student)
print(json_data)
```

```
# Convert Json to Dictionary

data = json.loads(json_data)
```

→ When to use dictionary?

- Store structured data.
- Represent objects (users, products)
- API responses.
- Configuration files.