



## Lab Work – Dictionary

### Objective

This lab helps develop a practical understanding of Python dictionaries and their applications in monitoring and managing structured data in engineering and AI-related tasks. The focus is on creating, accessing, updating, and deleting dictionary entries, handling nested dictionaries, and using built-in methods such as `.keys()`, `.values()`, `.items()`, `.get()`, and `.pop()`. Through real-world examples like smart home energy meters, solar home systems, student records, weather stations, inventory management, and patient monitoring, the lab strengthens skills in data organization, retrieval, and manipulation in a simple and applied way.

#### Task 1: Smart Home Energy Meter

Reads the dictionary storing energy readings.

Prints the grid power.

Adds a new appliance "TV" to the appliance status.

Updates battery state of charge.

Checks existence of solar power key.

Deletes the "Motor" key from appliance status.

```
In [6]: smart_meter = {
    "time": "18:32",
    "grid_power": 720,
    "solar_power": 340,
    "battery_soc": 67,
    "appliance_status": {"AC": "OFF", "Fridge": "ON", "Motor": "OFF"}
}

print(smart_meter["grid_power"])

smart_meter["appliance_status"]["TV"] = "ON"

smart_meter["battery_soc"] = 72

print("solar_power" in smart_meter)

del smart_meter["appliance_status"]["Motor"]

print(smart_meter)
```

```
720
True
{'time': '18:32', 'grid_power': 720, 'solar_power': 340, 'battery_soc': 72, 'appliance_status': {'AC': 'OFF', 'Fridge': 'ON', 'TV': 'ON'}}
```

### Task 2: University Student Record

Reads a dictionary with student info.

Prints the student's GPA.

Adds a new course with attendance.

Updates attendance for an existing course.

Checks if the key "gpa" exists.

```
In [7]: student_record= {
    "name": "Ali Khan",
    "roll_no": "EE-22-045",
    "gpa": 3.41,
    "courses": ["Circuit Analysis", "Linear Algebra", "Python"],
    "attendance": {"Circuit Analysis": 92, "Linear Algebra": 88, "Python": 96}
}

print(student_record["gpa"])
student_record["courses"].append("Digital Logic")
student_record["attendance"]["Digital Logic"] = 90
student_record["attendance"]["Linear Algebra"] = 91

print("gpa" in student_record)

print(student_record)
```

```
3.41
True
{'name': 'Ali Khan', 'roll_no': 'EE-22-045', 'gpa': 3.41, 'courses': ['Circuit Analysis', 'Linear Algebra', 'Python', 'Digital Logic'], 'attendance': {'Circuit Analysis': 92, 'Linear Algebra': 91, 'Python': 96, 'Digital Logic': 90}}
```

### Task 3: Weather Monitoring Station

Reads a dictionary with weather data.

Prints the wind direction.

Adds a new alert.

Updates wind speed.

Lists all top-level keys.

Removes the humidity key using .pop().

```
In [8]: weather_data={  "hour": 14,
                     "temperature": 33.8,
                     "humidity": 52,
                     "wind": {"speed": 12, "direction": "NE"},
                     "alerts": []}
print(weather_data["wind"]["direction"])
weather_data["alerts"].append("High Temperature")
weather_data["wind"]["speed"]=15
print(weather_data.keys())
weather_data.pop("humidity")
print(weather_data)

NE
dict_keys(['hour', 'temperature', 'humidity', 'wind', 'alerts'])
{'hour': 14, 'temperature': 33.8, 'wind': {'speed': 15, 'direction': 'NE'}, 'alerts': ['High Temperature']}
```

Task 4: Inventory System for Electronics Shop

Reads a dictionary representing an inventory item.

Prints the brand of the controller.

Reduces the stock and updates it.

Adds a new warranty feature.

Checks if price exists.

Merges another dictionary with discount.

```
In [9]: inventory = {
    "item_id": 1042,
    "name": "PWM Solar Charge Controller",
    "stock": 18,
    "price": 2500,
    "features": {"voltage": "12/24V", "current": "20A", "brand": "SolarTech"}
}
print(inventory["features"]["brand"])
inventory["stock"] -= 2
inventory["features"]["warranty"] = "1 year"
print("price" in inventory)
inventory.update({"discount": 10})
print(inventory)

SolarTech
True
{'item_id': 1042, 'name': 'PWM Solar Charge Controller', 'stock': 16, 'price': 2500, 'features': {'voltage': '12/24V', 'current': '20A', 'brand': 'SolarTech'}, 'warranty': '1 year'}, 'discount': 10}
```

## Task 5: Hospital Patient Monitoring System

Reads a dictionary with patient vitals.

Prints the oxygen level.

Adds a new medication.

Updates heart rate.

Retrieves medication schedule using .get() with default.

Prints all key-value pairs using .items().

```
In [10]: patient = {
    "id": 907,
    "name": "Faisal",
    "heart_rate": 82,
    "oxygen_level": 97,
    "medications": {"aspirin": "8AM", "vitamin_D": "12PM"}
}
print(patient["oxygen_level"])
patient["medications"]["antibiotic"] = "6PM"
patient["heart_rate"] = 78
print(patient["medications"].get("insulin", "Not Prescribed"))
print(patient.items())
```

97  
Not Prescribed  
dict\_items([('id', 907), ('name', 'Faisal'), ('heart\_rate', 78), ('oxygen\_level', 97), ('medications', {'aspirin': '8AM', 'vitamin\_D': '12PM', 'antibiotic': '6PM'})])