Dictionaries in Python

I. Simple Dictionary

1. Dictionary

- A data structure used to store data in key-value pairs.
- Each key is used to access its corresponding value.

2. Create a Dictionary

Syntax:

Dictionary_name = {key1: value1, key2: value2, ..., keyn: valuen}

- Note: The keys must be strings or numbers.

Example:

```
Student = {"Name": "Mohamed", "Age": 21, "Major": "Computer Sciences"}
```

3. Accessing Dictionary Values

Syntax:

Value = Dictionary_name[key]

Example:

```
print(Student["Name"])
print(Student["Age"])
```

Output:

```
Mohamed
21
```

- Note: If the key doesn't exist, Python raises an error.

Using get method:
Syntax:

Value = dictionary_name.get(key, default_value)

Example:

```
print(Student.get("Age"))
print(Student.get("City", "doesn't exist"))
```

Output:

21
doesn't exist

4. Adding and Modifying Elements

- **Modifying an element**
  Syntax:

  Dictionary_name[key] = new_value

- **Adding an element**
  Syntax:

  Dictionary_name[new_key] = value

Example:

```
Student["Name"] = "Abderrahmane"
print(Student)
Student["Gender"] = "male"
print(Student)
```

5. Deleting an Element

1. **Using del**
   Syntax:

   del dictionary_name[key]

   Example:

   ```
   del Student["Gender"]
   print(Student)
   ```

   Output:

   Student = {"Name": "Abderrahmane", "Age": 21, "Major": "Computer Sciences"}

2. **Using pop**
   Syntax:

   dictionary_name.pop(key)

   Example:

   ```
   print(Student.pop("Gender"))
   ```

   Output:

   Student = {"Name": "Abderrahmane", "Age": 21, "Major": "Computer Sciences"}

3. **Using clear**
   To remove all elements in a dictionary.
   Syntax:

   ```
   dictionary_name.clear()
   ```

6. Dictionary Iteration

6.1 Loop Over Keys

Syntax:

```
for key in dictionary_name:
    # code
```

Example:

```
for key in Student:
    print(key)
```

Output:

```
Name
Age
Major
```

6.2 Loop Over Values

Syntax:

```
for value in dictionary_name.values():
    # code
```

Example:

```
for value in Student.values():
    print(value)
```

Output:

```
Abderrahmane
21
Computer Sciences
```

6.3 Loop Over Key-Value Pairs

- **Method 1:**

  ```
  for key, value in dictionary_name.items():
      # code
  ```

  Example:

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

Output:

```
Name Abderrahmane
Age 21
Major Computer Sciences
```

- **Method 2:**

```
for elem in dictionary_name.items():
    # code
```

Example:

```
for elem in Student.items():
    print(elem)
```

Output:

```
('Name', 'Abderrahmane')
('Age', 21)
('Major', 'Computer Sciences')
```

II. Nested Dictionaries

1. Definition

- A dictionary that contains other dictionaries as values.

2. Syntax

```
Dictionary_name = {
    key1: {subkey11: value11, subkey12: value12},
    key2: {subkey21: value21, subkey22: value22}
}
```

Example:

```
students = {
    "student1": {"name": "Ali", "age": 20, "major": "Economics"},
    "student2": {"name": "Sara", "age": 22, "major": "Business"}
}
```

- student1, student2 are main keys of the main dictionary students.
- Name, age, major are inner keys
- Their values are dictionaries themselves.

3. Accessing an Element in a Nested Dictionary

Syntax:

Dictionary_name[main_key][inner_key]

Example:

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

Output:

20

4. Modifying a Value

Syntax:

Dictionary_name[main_key][inner_key] = value

Example:

```
students["student1"]["age"] = 21
print(students["student1"])
```

Output:

{'name': 'Ali', 'age': 21, 'major': 'Economics'}

5. Adding a New Inner Key

Syntax:

Dictionary_name[main_key][new_inner_key] = value

Example:

```
students["student1"]["math"] = 18
print(students["student1"])
```

Output:

{'name': 'Ali', 'age': 21, 'major': 'Economics', 'math': 18}

6. Adding a New Main Key

Syntax:

Dictionary_name[new_main_key] = {inner_key1: value1, inner_key2: value2}

Example:

```
students["student3"] = {"name": "Omar", "age": 23, "major": "marketing"}
print(students["student3"])
```

Output:

{'name': 'Omar', 'age': 23, 'major': 'marketing'}

7. Deleting

- **Deleting a Main Key**
  Syntax:

  ```
  del Dictionary_name[main_key]
  Dictionary_name.pop(main_key)
  ```

  Example:

  ```
  del students["student3"]
  students.pop("student3")
  ```

- **Deleting an Inner Key**
  Syntax:

  ```
  del Dictionary_name[main_key][inner_key]
  ```

  Example:

  ```
  del students["student2"]["age"]
  students["student1"].pop("age")
  ```

8. Looping Through Main Keys

Syntax:

```
for key in dictionary_name:
    print(key)
```

Example:

```
for student in students:
    print(student)
```

Output:

```
student1
student2
student3
```

9. Looping Through Nested Values

Syntax:

```
for value in dictionary_name.values():
    print(value[inner_key])
```

Example:

```
for value in students.values():
    print(value["name"])
```

Output:

Ali
Sara
Omar