Create and Access a Dictionary

Create a dictionary that stores the names and ages of three people. Then, print the age of a specific person.

```
ages = {"Rohan": 23, "Sid": 26, "Namrata": 24, "Shubham": 24}
n = ages["Namrata"]
print(f"Namrata is {n} years old!")
```

```
Namrata is 24 years old!
```

Add or Update a Value

Add a new person, "Lilly", with an age of 28 to the dictionary. Then, update sid's age to 26. Print the updated dictionary.

```
ages["Lilly"] = 28
ages["Sid"] = 26
print(ages)
```

```
{'Rohan': 23, 'Sid': 26, 'Namrata': 24, 'Shubham': 24, 'Lilly': 28}
```

Check If a Key Exists

Check if the name "Rohan" exists in the dictionary. If it exists, print his age. If not, print a message saying, `"Rohan is not in the dictionary."`

```
if "Rohan" in ages:
  print("Rohan is", ages["Rohan"], "years old!")
else:
  print("Rohan is not in the dictionary.")
```

```
Rohan is 23 years old!
```

Iterate Over a Dictionary

Iterate through the dictionary and print each name and age in the format: `"Name" is "Age" years old`

```
for key,value in ages.items():
  print(key, "is", value, "years old")
```

```
Rohan is 23 years old
Sid is 26 years old
Namrata is 24 years old
Shubham is 24 years old
Lilly is 28 years old
```

Remove a Key from a Dictionary

Remove "Lilly" from the dictionary. After that, print the updated dictionary.

```
ages.pop("Lilly", None) #here None avoids error if Lilly doesn't exist
print(ages)
```

```
{'Rohan': 23, 'Sid': 26, 'Namrata': 24, 'Shubham': 24}
```

Default Values with get

Try to get the age of "Namrata" from the dictionary. If "Namrata" does not exist, return a default message: "Age not found."

```
print(ages.get("Namrata", "Age not found!")) #Namrata is present so will print 24
print(ages.get("Suhas", "Age not found!"))  # Not having Suhas so will print default value.
```

```
24
Age not found!
```

Combine Two Dictionaries

---

new_ages = {"Adveet": 21, "Shanky": 29}

Merge this dictionary into the existing one.

```
new_ages = {"Adveet": 21, "Shanky": 29}
ages.update(new_ages)
print(ages)
```

```
{'Rohan': 23, 'Sid': 26, 'Namrata': 24, 'Shubham': 24, 'Adveet': 21, 'Shanky': 29}
```

Filter a Dictionary

Create a new dictionary containing only the people whose age is greater than or equal to 25 from ages.

```
seniors = dict()
for name, age in ages.items():
  if age >= 25:
    seniors[name] = age
print("Original dict is below -->")
print(ages)
print("New dict with senior people is below -->")
print(seniors)

# #another way to do this
# new_ages = {name: age for name, age in ages.items() if age >= 25}
# print(new_ages)
```

```
{'Sid': 26, 'Shanky': 29}
```

Count Character Frequency

Given a string --> "hello world", count the frequency of each character (ignore spaces) and store it in a dictionary.

```
text = "hello world"
d = dict()
for char in text:
  if char != " ":
    d[char] = d.get(char, 0) + 1
print(d)
```

```
{'h': 1, 'e': 1, 'l': 3, 'o': 2, 'w': 1, 'r': 1, 'd': 1}
```

Reverse a Dictionary

```
reversed_ages = {age: name for name, age in ages.items()}
print(reversed_ages)
```

```
{23: 'Rohan', 26: 'Sid', 24: 'Shubham', 21: 'Adveet', 29: 'Shanky'}
```

Access Nested Dictionary Values

---

student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}, } Print Suhas's CSS score.

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}, }
print("Suhas's CSS marks are" ,student_info["Suhas"]['CSS'])
```

```
Suhas's CSS marks are 88
```

Add Data to a Nested Dictionary

Add a new subject, "Java", with a score of 78 for Rohan. Print the updated dictionary.

```
student_info["Rohan"]["Java"] = 78
print(student_info)
```

```
{'Rohan': {'python': 85, 'CSS': 92, 'Java': 78}, 'Suhas': {'python': 90, 'CSS': 88}}
```

Iterate Over a Nested Dictionary

Iterate through student_info and print each student's name along with their subject scores.

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
for name, subjects in student_info.items():
  print("This is", name, "'s Report -->")
  for sub, marks in subjects.items():
    print(sub, "=", marks)
  print()
```

```
This is Rohan 's Report -->
python = 85
CSS = 92
Java = 78

This is Suhas 's Report -->
python = 90
CSS = 88
```

Start coding or generate with AI.

Write a Python script to sort (ascending and descending) a dictionary by value

```
dict = {"PYTHON": 88, "JAVA": 78, "HTML": 99, "CSS": 45}
sorted_dict = {}
list = []
for key,value in dict.items():
  list.append((value,key))
for key, value in sorted(list, reverse = False):
  sorted_dict[value] = key
print("Sorted dict by values is", sorted_dict)
```

```
Sorted dict by values is {'CSS': 45, 'JAVA': 78, 'PYTHON': 88, 'HTML': 99}
```

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

Start coding or generate with AI.

```
#updating dict
student_info = {"Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
student_info["Rohan"]["Java"] = 78

print(student_info)
```

```
{'Rohan': {'python': 85, 'CSS': 92, 'Java': 78}, 'Suhas': {'python': 90, 'CSS': 88}}
```

```
#deleting data using pop() method
dictionary = {'a': 1, 'b': 2, 'c': 3}
print(dictionary.pop('b'))
print(dictionary)
```

```
2
{'a': 1, 'c': 3}
```

```
#deleting data using pop() method
dictionary = {'a': 1, 'b': 2, 'c': 3}
print(dictionary.pop('b'))
print(dictionary)
```

```
2
{'a': 1, 'c': 3}
```

Double-click (or enter) to edit

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
css_score = student_info["Suhas"].pop("CSS")
print(student_info)
```

```
{'Rohan': {'python': 85, 'CSS': 92}, 'Suhas': {'python': 90}}
```

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
css_score = student_info["Suhas"].pop("HTML", "not present")
print(student_info)
```

```
{'Rohan': {'python': 85, 'CSS': 92}, 'Suhas': {'python': 90, 'CSS': 88}}
```

```
Start coding or generate with AI.
```

Deleting values using Del keyword

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
del student_info["Suhas"]["CSS"]
print(student_info)
```

```
{'Rohan': {'python': 85, 'CSS': 92}, 'Suhas': {'python': 90}}
```

method of dict

`keys()`

Keys method will return all keys from dict having

```
print(student_info.keys())
```

```
dict_keys(['Rohan', 'Suhas'])
```

```
dict = {}
print(dict.keys())
```

```
dict_keys([])
```

values() method

```
print(student_info.values())
```

```
dict_values([{'python': 85, 'CSS': 92}, {'python': 90, 'CSS': 88}])
```

items() mthod

```
print(student_info.values())
```

```
dict_values([{'python': 85, 'CSS': 92}, {'python': 90, 'CSS': 88}])
```

```
print(student_info.values())
```

```
dict_values([{'python': 85, 'CSS': 92}, {'python': 90, 'CSS': 88}])
```

```
student_info = { "Rohan": {"python": 85, "CSS": 92}, "Suhas": {"python": 90, "CSS": 88}}
for name, subjects in student_info.items():
  for sub, marks in subjects.items():
    print(sub, "=", marks)
```

```
python = 85
CSS = 92

python = 90
CSS = 88
```

```
students = {"rollno1": 22, "rollno2": 33, "rollno3": 88}
for rollno in students.keys():
  students[rollno] = students[rollno] + 1
for rollno, marks in students.items():
  print(rollno,marks)
```

```
rellno1 23
rollno2 34
rollno3 89
```

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit

Double-click (or enter) to edit