Lecture07

July 6, 2025

[1]:

```python
fruits = {"apple", "banana", "cherry", "apple"}
print(fruits)

numbers = set([1, 2, 3, 1, 2, 4])
print(numbers)
```

```
{'apple', 'banana', 'cherry'}
{1, 2, 3, 4}
```

[4]:

```python
fruits = {"apple", "banana", "cherry"}

fruits.add("orange")
print(fruits)

fruits.remove("banana")
print(fruits)

fruits.discard("grape")
print(fruits)

remove_item = fruits.pop()
print(remove_item)
print(fruits)

fruits.clear()
print(fruits)
```

```
{'orange', 'apple', 'banana', 'cherry'}
{'orange', 'apple', 'cherry'}
{'orange', 'apple', 'cherry'}
orange
{'apple', 'cherry'}
set()
```

[5]:

```python
set1 = {1, 2, 3}
set2 = {3, 4, 5}

print(set1.union(set2))
print(set1.intersection(set2))
print(set1.difference(set2))
print(set1.symmetric_difference(set2))
```

```
{1, 2, 3, 4, 5}
{3}
{1, 2}
{1, 2, 4, 5}
```

[6]:

```
set1 = {1, 2, 3, 4}
set2 = {3, 4, 5, 6}

union_set = set1 | set2
print("Union", union_set)

intersection_set = set1 & set2
print("Intersection", intersection_set)

difference_set = set1 - set2
print("Difference", difference_set)

sym_diff_set = set1 ^ set2
print("Symmetric Difference", sym_diff_set)
```

```
Union {1, 2, 3, 4, 5, 6}
Intersection {3, 4}
Difference {1, 2}
Symmetric Difference {1, 2, 5, 6}
```

[7]:

```
setA = {1, 2, 3, 4}
setB = set([8, 9, 10])

setA.add(5)
setB.update([6, 7])
Uset = setA | setB
print(Uset)
print(len(Uset))

setB.update("ABCD")
setA.update([6, 7, 8])
print(setB)

print(setA.intersection(setB))
print(setA ^ setB)

setB.remove("B")
setB.discard(10)
print(setB)
print(setA.clear())
```

```python
for val in Uset:
    print(val)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
10
{6, 7, 8, 9, 10, 'B', 'A', 'D', 'C'}
{8, 6, 7}
{1, 2, 3, 4, 5, 9, 10, 'B', 'A', 'D', 'C'}
{6, 7, 8, 9, 'A', 'D', 'C'}
None
1
2
3
4
5
6
7
8
9
10
```

[8]:

```python
def remove_duplicates(lst):
    return list(set(lst))


numbers = [1, 2, 3, 1, 2, 4, 5, 6, 5, 4, 3]
print(remove_duplicates(numbers))
```

```
[1, 2, 3, 4, 5, 6]
```

[13]:

```python
attendance_week = [
    ["Alice", "Bob", "Charlie", "David"],
    ["Alice", "Charlie", "David"],
    ["Alice", "Bob", "David"],
    ["Alice", "David", "Eve"],
    ["Bob", "Charlie", "David"],
]

attendance_sets = [set(day) for day in attendance_week]
print(attendance_sets)

present_every_day = set.intersection(*attendance_sets)
print("Present every day:", present_every_day)

all_students = set.union(*attendance_sets)
absent_at_least_one_day = all_students - present_every_day
print("Absent at least one day:", absent_at_least_one_day)
```

```
first_day_present = attendance_sets[0]
last_day_present = attendance_sets[-1]
first_day_but_not_last = list(first_day_present - last_day_present)
print("First day but not last:", first_day_but_not_last)

unique_students_count = len(all_students)
print("Total unique students:", unique_students_count)
```

```
[{'Alice', 'Charlie', 'Bob', 'David'}, {'Alice', 'Charlie', 'David'}, {'Alice',
'Bob', 'David'}, {'Alice', 'Eve', 'David'}, {'Charlie', 'Bob', 'David'}]
Present every day: {'David'}
Absent at least one day: {'Alice', 'Charlie', 'Eve', 'Bob'}
First day but not last: ['Alice']
Total unique students: 5
```

[74]:

```
survey_result = [
    ["Python", "JavaScript", "C++"],
    ["Python", "JavaScript", "C#"],
    ["Python", "Java"],
    ["Python", "C++", "JavaScript"],
    ["Python", "JavaScript", "C++", "Java"],
]

merge_participants = []
for participant in survey_result:
    merge_participants += participant
unique_language = set.union(*[set(participant)
                              for participant in survey_result])

choices_sets = [set(participant) for participant in survey_result]
print(choices_sets)

common_languages = set.intersection(*choices_sets)
print("Languages chosen by all participants:", common_languages)

only_one_diff_languages = set([])
for language in unique_language:
    if merge_participants.count(language) == 1:
        only_one_diff_languages.add(language)
print("Languages chosen by only one participant:", only_one_diff_languages)

print("number of unique language:", len(unique_language))

same_participant_can_write_two_languages = set([])
for language in unique_language:
    if merge_participants.count(language) == 2:
        same_participant_can_write_two_languages.add(language)
```

```
print(
    "Languages chosen by exactly two participants:",
    same_participant_can_write_two_languages,
)

set_identical_language = set([])
for i in range(len(choices_sets)):
    for j in range(len(choices_sets) - 1, i, -1):
        if choices_sets[i] == choices_sets[j]:
            set_identical_language.add((i + 1, j + 1))
print("Participants with identical choices:", list(set_identical_language))
```

```
[{'Python', 'JavaScript', 'C++'}, {'Python', 'JavaScript', 'C#'}, {'Python',
'Java'}, {'Python', 'JavaScript', 'C++'}, {'Python', 'JavaScript', 'C++',
'Java'}]
Languages chosen by all participants: {'Python'}
Languages chosen by only one participant: {'C#'}
number of unique language: 5
Languages chosen by exactly two participants: {'Java'}
Participants with identical choices: [(1, 4)]
```

[33]:

```
student = dict(name="Alice", age=25, grade="A")
print(student)

student = dict([("name", "Alice"), ("age", 25), ("grade", "A")])
print(student)
```

```
{'name': 'Alice', 'age': 25, 'grade': 'A'}
{'name': 'Alice', 'age': 25, 'grade': 'A'}
```

[31]:

```
student = {}

student["name"] = "Alice"
student["age"] = 25
student["grade"] = "A"
print(student)
```

```
{'name': 'Alice', 'age': 25, 'grade': 'A'}
```

[32]:

```
student = {"name": "Alice", "age": 25, "grade": "A"}
print(student["name"])
print(student["age"])
print(student["grade"])
```

```
Alice
25
A
```

[34]:

```
phonebook = {"Anirach": "777-1111", "Mickey": "777-2222",
             "Donald": "777-3333"}

print(phonebook)

print(phonebook["Mickey"])
print(phonebook.get("Donald"))

key = "Pluto"
if key in phonebook:
    print(phonebook["Pluto"])
else:
    print(key + "not in phonebook")

phonebook["Simpson"] = "777-4567"
phonebook["Pluto"] = "777-4444"
phonebook["Mickey"] = "777-2122"
print(phonebook)

del phonebook["Simpson"]
print(phonebook)
```

```
{'Anirach': '777-1111', 'Mickey': '777-2222', 'Donald': '777-3333'}
777-2222
777-3333
Plutonot in phonebook
{'Anirach': '777-1111', 'Mickey': '777-2122', 'Donald': '777-3333', 'Simpson':
'777-4567', 'Pluto': '777-4444'}
{'Anirach': '777-1111', 'Mickey': '777-2122', 'Donald': '777-3333', 'Pluto':
'777-4444'}
```

[37]:

```
student = {"name": "Alice", "age": 25, "grade": "A",
           "major": "Computer Science"}

for key in student:
    print(f"{key}: {student[key]}")
```

```
name: Alice
age: 25
grade: A
major: Computer Science
```

[38]:

```
student = {"name": "Alice", "age": 25, "grade": "A",
            "major": "Computer Science"}

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

```
Alice
```

```
25
A
Computer Science
```

[40]:

```python
student = {"name": "Alice", "age": 25, "grade": "A",
            "major": "Computer Science"}

for key, value in student.items():
    print(f"{key}: {value}")
```

```
name: Alice
age: 25
grade: A
major: Computer Science
```

[41]:

```python
student = {"name": "Alice", "age": 25, "grade": "A"}

student["age"] = 26
student["major"] = "Computer Science"
print(student)

del student["grade"]
print(student)

remove_major = student.pop("major")
print(remove_major)
print(student)
```

```
{'name': 'Alice', 'age': 26, 'grade': 'A', 'major': 'Computer Science'}
{'name': 'Alice', 'age': 26, 'major': 'Computer Science'}
Computer Science
{'name': 'Alice', 'age': 26}
```

[42]:

```python
phonebook = {"Anirach": "777-1111", "Mickey": "777-2222",
              "Donald": "777-3333"}

phonebook["Bart"] = [1, 3, 5]

elements = len(phonebook)
print("There are ", elements, " names in phonebook")

for key in phonebook:
    print(key, "phone number is: ", phonebook[key])

phonebook["Bart"][1] = 9
print(phonebook)
```

```
There are  4  names in phonebook
Anirach phone number is:  777-1111
```

```
Mickey phone number is:  777-2222
Donald phone number is:  777-3333
Bart phone number is:  [1, 3, 5]
{'Anirach': '777-1111', 'Mickey': '777-2222', 'Donald': '777-3333', 'Bart': [1,
9, 5]}
```

[52]:

```python
student = {"name": "Alice", "age": 25, "grade": "A",
            "major": "Computer Science"}
print(student.keys())
print(student.values())
print(student.items())
print(student.get("name"))
print(student.get("grade", "Not Found"))
major = student.pop("major")
print(major)
print(student)
last_item = student.popitem()
print(last_item)
print(student)
student.clear()
print(student)
```

```
dict_keys(['name', 'age', 'grade', 'major'])
dict_values(['Alice', 25, 'A', 'Computer Science'])
dict_items([('name', 'Alice'), ('age', 25), ('grade', 'A'), ('major', 'Computer
Science')])
Alice
A
Computer Science
{'name': 'Alice', 'age': 25, 'grade': 'A'}
('grade', 'A')
{'name': 'Alice', 'age': 25}
{}
```

[65]:

```python
phonebook = {
    "Anirach": "777-1111",
    "Mickey": "777-2222",
    "Donald": "777-3333",
    "Pluto": "777-4444",
}

heroesdict = {}
heroesdict["Hulk"] = "888-1111"
heroesdict["Iron Man"] = "888-2222"
print(heroesdict.get("Halk", "Key not found"))
print(heroesdict.get("Hulk", "Key not found"))

for key, value in phonebook.items():
    print(key, value)

print(phonebook.keys())
print(phonebook.values())

print(phonebook.pop("Mick", "Element not found"))
print(phonebook.pop("Mickey", "Element not found"))
print(phonebook)
print(phonebook.popitem())
print(phonebook)
phonebook.clear()
print("After clear")
print(phonebook)
```

```
Key not found
888-1111
Anirach 777-1111
Mickey 777-2222
Donald 777-3333
Pluto 777-4444
dict_keys(['Anirach', 'Mickey', 'Donald', 'Pluto'])
dict_values(['777-1111', '777-2222', '777-3333', '777-4444'])
Element not found
777-2222
{'Anirach': '777-1111', 'Donald': '777-3333', 'Pluto': '777-4444'}
('Pluto', '777-4444')
{'Anirach': '777-1111', 'Donald': '777-3333'}
After clear
{}
```

[12]:

```
performance_data = {
    "Sales": {
        "Alice": [80, 85, 88, 90],
        "Alex": [80, 85, 88, 90],
        "Bob": [70, 75, 78, 80],
        "Charlie": [60, 65, 70, 72],
    },
    "Engineering": {
        "David": [90, 92, 94, 95],
        "Eve": [85, 88, 87, 90],
        "Frank": [88, 87, 86, 85],
    },
    "HR": {
        "Grace": [70, 72, 74, 76],
        "Heidi": [65, 68, 70, 73],
        "Ivan": [60, 62, 64, 66],
    },
```

```python
}
average_scores = {}
for department, employees in performance_data.items():
    average_scores[department] = {}
    for employee, scores in employees.items():
        average = sum(scores) / len(scores)
        average_scores[department][employee] = average
# print(average_scores)

for department in average_scores.keys():
    department_scores_list = []
    # print(department)
    for scores in average_scores[department].values():
        # print(scores)
        department_scores_list.append(scores)
    average_score = sum(department_scores_list)/len(department_scores_list)
    average_scores[department]["average_scores"] = average_score

for department in average_scores.keys():
    department_scores_list = []
    # print(department)
    for scores in average_scores[department].values():
        # print(scores)
        department_scores_list.append(scores)
    max_score = max(department_scores_list)
    best_member = []
    for member, value in list(average_scores[department].items()):
        if value == max_score:
            best_member.append(member)
    average_scores[department]["top_performer"] = [max_score]
    average_scores[department]["top_performer"].extend(best_member)
# print(average_scores)

list_top_department = []
for department, values in list(average_scores.items()):
    # print(values["average_scores"])
    list_top_department.append(values["average_scores"])
max_department_score = max(list_top_department)
# print(max_department_score)

top_department = []
for department, values in list(average_scores.items()):
    if values["average_scores"] == max_department_score:
        top_department.append(department)
average_scores['top_department'] = top_department
# print(average_scores)
```

```
print("Summary Report:")
for key, values in average_scores.items():
    if key not in ["top_department", "average_scores", "top_performer"]:
        print(f"Department: {key}")
        for member in values.keys():
            if member not in ["top_department", "average_scores",
                               "top_performer"]:
                print(f"  {member}: Average Score = ",end="")
                print(f"{average_scores[key][member]}")
        # print(average_scores)
        str_members = ""
        # print(average_scores[key]["top_performer"][1:])
        for top_perfomer in average_scores[key]["top_performer"][1:]:
            if str_members == "":
                str_members = top_perfomer
            else:
                str_members += ", " + top_perfomer
        print(f"Top Performer: {str_members} with Average score = ",end="")
        print(f"{average_scores[department]['top_performer'][0]}")

# print(average_scores["top_department"])
str_department = ""
for department in average_scores["top_department"]:
    if str_department == "":
        str_department = department
    else:
        str_department += ", " + department

list_best_department = []
for department, value in average_scores.items():
    if department not in ["top_department"]:
        list_best_department.append(value["average_scores"])

max_best_department = max(list_best_department)
best_department = ""
for department, value in average_scores.items():
    # print(value)
    if department not in ["top_department", "average_scores",
                           "top_performer"]:
        if value["average_scores"] == max_best_department:
            if best_department == "":
                best_department = department
            else:
                best_department += ", " + department
print()
print(f"Best Department: {best_department} with Average score = ",end="")
print(f"{max_best_department:.2f}")
```

```
Summary Report:
Department: Sales
  Alice: Average Score = 85.75
  Alex: Average Score = 85.75
  Bob: Average Score = 75.75
  Charlie: Average Score = 66.75
Top Performer: Alice, Alex with Average score = 73.0
Department: Engineering
  David: Average Score = 92.75
  Eve: Average Score = 87.5
  Frank: Average Score = 86.5
Top Performer: David with Average score = 73.0
Department: HR
  Grace: Average Score = 73.0
  Heidi: Average Score = 69.0
  Ivan: Average Score = 63.0
Top Performer: Grace with Average score = 73.0

Best Department: Engineering with Average score = 88.92
```