29th April 21

Previous Day

- Tuples
- Dictionary
- all dictionary operations: add a single element, how to add multiple elements, we also learned how to merge 2 dictionaries update(), we also learned how to delete all the items in the dictionary, delete the entire dictionary including the variable which stores the dictionary.

Problem Solving

Lecture Flow

```
# Values can be of any type in a dictionary
# Keys can be of any immutable type in a dictionary: basically everything
except list and dictionary

# The data types of keys in a dictionary can also be different. However,
the key should be unique, else the old value saved in the same key gets
overwritten.

# Let us say we have to add 5 key:value pairs into a dictionary

# we need a sequence of size 5 that contains all the keys.
# we need a sequence of size 5 that contains all the values.
```

Program

```
21  newKeys = ["Sumuthra", "Sankalp", "Sravan", "Amol", "Abhishek"]
22  newValues = [23, 24, 25, 26, 27]
23
24  key_to_be_added = newKeys[0]
25  value_to_be_added = newValues[0]
26  AgeLog[key_to_be_added] = value_to_be_added
27
28  key_to_be_added = newKeys[1]
29  value_to_be_added = newValues[1]
30  AgeLog[key_to_be_added] = value_to_be_added
31
32  key_to_be_added = newKeys[2]
33  value_to_be_added = newValues[2]
34  AgeLog[key_to_be_added] = value_to_be_added
35
36  key_to_be_added = newKeys[3]
37  value_to_be_added = newValues[3]
38  AgeLog[key_to_be_added] = value_to_be_added
39
40  key_to_be_added = newKeys[4]
41  value_to_be_added = newValues[4]
42  AgeLog[key_to_be_added] = value_to_be_added
43
44  print(AgeLog)
```

Output

```
~/RichBiodegradableLicenses$ python3 lecture19_notes.py
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sa
nkalp': 24, 'Sravan': 25, 'Abhishek': 27}
~/RichBiodegradableLicenses$
```

```
General format:
key_to_be_added = newKeys[idx]
value_to_be_added = newValues[idx]
AgeLog[key_to_be_added] = value_to_be_added
variable here: idx
starting: 0
ending: 4
range(0,5)
```

Start thread

Program

```
21  newKeys = ["Sumuthra", "Sankalp", "Sravan", "Amol", "Abhishek", "Durjoy"]
22  newValues = [23, 24, 25, 26, 27, 28]
```

```
53  for idx in range(0,5):
54      key_to_be_added = newKeys[idx]
55      value_to_be_added = newValues[idx]
56      AgeLog[key_to_be_added] = value_to_be_added
57
58  for idx in range(0,5):
59      AgeLog[newKeys[idx]] = newValues[idx]
60
61  print(AgeLog)
```

Output

```
~/RichBiodegradableLicenses$ python3 lecture19_notes.py
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sa
nkalp': 24, 'Sravan': 25, 'Abhishek': 27}
```

Program

```
for idx in range(0,len(newKeys)):
  AgeLog[newKeys[idx]] = newValues[idx]

print(AgeLog)
```

Output

```
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sa
nkalp': 24, 'Sravan': 25, 'Abhishek': 27, 'Durjoy': 28}
~/RichBiodegradableLicenses$
```

```
66   # this is how we add multiple values to a dictionary
```

Program

```
68   keyNames = ["Durjoy", "Va
69
70   # key_to_delete = keyNames[0]
71   # del AgeLog[key_to_delete]
72
73   # key_to_delete = keyNames[1]
74   # del AgeLog[key_to_delete]
75
76   # key_to_delete = keyNames[idx]
77   # del AgeLog[key_to_delete]
78   # variable: idx, start value: 0, end value: 1 => range(0,2)
79
80   for idx in range(0,2):
81     key_to_delete = keyNames[idx]
82     del AgeLog[key_to_delete]
83
84   print(AgeLog)
```

Output

```
{'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sr
avan': 25, 'Abhishek': 27}
~/RichBiodegradableLicenses$
```

Program

```
100  names = input().split()
101
102  ages = input().split()
103
104  # 0,1,2, ...., len-1: range(0, len)
105
106  UserInputAges = {}
107
108  for idx in range(0,len(names)):
109      UserInputAges[names[idx]] = int(ages[idx])
110
111  print(UserInputAges)
```

Output

```
{'Vaibhav': 30, 'Priyesh': 25, 'Megha': 27, 'Shubham': 28}
\~/RichBiodegradableLicenses$
```

Online Judges

```
# Online judges: company coding tests and sometimes even durign interviews

# normally companies will provide you a link like this/ test link:
https://www.hackerrank.com/robin-may-retest-set-1

# 1. First you forward your CV to a company
# 2. Then the company screens those CVs
# 10000 people => 1000 people
# 3. Conduct an online problem solving test for these people.
# For this purpose they use online platforms called judges to evaluate
your code in terms of correctness(syntax and logic) and efficiency
# 1000 people => 50/80 people

# mostly=> 90% you will give the test on either hackerrank or interviewbit
# familiarity with hackerrank.com can give you a huge advantage
```

MCQs

On the give dictionary what is the output of:
AgeLog = {'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27, 'Durjoy': 28}

Attempted - 32 (61.54%) EASY

keyNames = ["Durjoy", "Vaibhav", "Megha"]
for idx in range(0,2):
del AgeLog[keyNames[idx]]

- [x] {'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27} 56.25%
- [ ] {'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27} 25%
- [ ] {} 18.75%

What is the output of:

```
names = input().split()
ages = input().split()
UserInputAges = {}
for idx in range(0,len(names)):
  UserInputAges[names[idx]] = list(ages[idx])

print(UserInputAges)
```

2

User inputs are:
P Q R S
10 11 12 13

- {"P": 10, "Q": 11, "R": 12, "S": 13}
- {"P": "10", "Q": "11", "R": "12", "S": "13"}
- ✓ {"P": ["10"], "Q": ["11"], "R": ["12"], "S": ["13"]} 69.23%
- error

On the give dictionary what is the output of:

```
AgeLog = {'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23,
'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27, 'Durjoy': 28}

keyNames = ["Durjoy", "Vaibhav", "Megha"]
for idx in range(0,2):
 del AgeLog[keyNames[idx]]
```

Attempted - 33 (63.46%)

EASY

- ✓ {'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27} 57.58%
- {'Amol': 26, 'Priyesh': 25, 'Sumuthra': 23, 'Sankalp': 24, 'Sravan': 25, 'Abhishek': 27} 24.24%
- {} 18.18%

What is the output of:

```
names = input().split()
ages = input().split()
UserInputAges = {}
for idx in range(0,len(names)):
  UserInputAges[names[idx]] = list(ages[idx])

print(UserInputAges)
```

Attempted - 36 (69.23%) EASY

User inputs are:

P Q R S

10 11 12 13

{"P": 10, "Q": 11, "R": 12, "S": 13}	11.11%
{"P": "10", "Q": "11", "R": "12", "S": "13"}	11.11%
{"P": ["10"], "Q": ["11"], "R": ["12"], "S": ["13"]}	69.44%
error	8.33%