20: Dictionaries

Tuesday, February 23, 2021 12:30 PM

Dictionaries part 1

Monday, February 21, 2022 9:59 PM

Sequences we've learned:

Lists
tuples
strings

all ordered from 0 to len-1

Another way to organize data:

dictionaries!

word → definition

key → value

Year_founded = {"www": 1893, "WA": 1889}

Key, Colon, Value

Curly Bracket!

Comma

```
1  #a list to store a person's health information
2  patient_1 = [5,20,125]
3
4  #wait, which number means what?
5  #use a dictionary so you can name the indicies!
6
7  patient_2 = {"height":5, "age":20, "weight":125}
8
9  print("the patient's height is", patient_2["height"], "feet")
```

More examples!

empty dictionary

key

value (a list)

```
1  students = {}
2  students[20891] = ["W0183782", "W0243810"]
3  students[20892] = ["W012345", "W987654"]
```

```
16 print(section[20891]) #print all
17
18 for student in section[20891]: #use in loop
19     print(student)
```

Name	Value
students	{20891: ['W0183782', 'W0243810'], 20892: ['W012345', 'W

Try yourself: Make a dictionary to store this data:

Employee with first name Jane, last name Doe, Type is faculty, and W# is 0987654

```python
employee = {"First": "Jane",
            "Last": "Doe",
            "Type": "Faculty",
            "W#": 98365438}

```

```python
#let's say I wanted to store the number
#of students who got each grade on the
#midterm
grades = {"A":10, "B":18, "C":6, "D":2}

print(grades["A"])
#print(grades["a"]) #key error
grades["E"] = -1 # new mapping!
grades["A"] = 11 #overwrite

#we can check if a key is in a dict
print("F" in grades) #False
print("C" in grades)

#we can remove entries
del grades["E"]
```

Dictionaries in lists

Tuesday, February 22, 2022 11:06 PM

```
7  patient_2 = {"height":5, "age":20, "weight":125}
```

```
11 #what if you had lots of patients? you can't give each one a variable
12 patient_3 = {"name":"Joel","height": 6, "age":22, "weight":212}
13
14 #put the dictionary into a list!
15
16 all_patients = [patient_2, patient_3]
17
18 #then all you need to do to print all patient's heights is:
19 for patient in all_patients:
20     print(patient["name"],"weighs", patient["weight"])
21
```

```
total_weights=0
for patient in all_patients:
    total_weights += patient["weight"]
```

will this run? (any Typos?)

```
    print(patient["name"],"weighs", patient["weight"])
KeyError: 'name'
```

Patient_2 didn't have a key "name"

```
18 #you try!
19 #print out all patients name and height
20 # Joel: 5 tall
21 # Yoel: 6 tall
22
3  for p in all_patients:
4      print(p["name"], ":", p["height"], "tall")
5
```

Iterating in a dictionary

Wednesday, March 3, 2021 11:35 AM

```
dict = {"A":[1,2,3], "B":[4,5,6]}
print(dict["A"][0])
```

Iterating on a dictionary

quake = {"lat": 45, "long": -121.6, "time": "7/27/2016", "mag": 1.43}

four options for a for loop

1. for key in quake
2. for key in quake.keys()

> Can use to get value → quake[key]

3. for val in quake.values() - value but no key
4. for key, val in quake.items() - both

These return an iterable!
dict.keys, dict.values, dict.items

To have it as a list:
list(quake.keys())
list(quake.items()) ← list of tuples [(key, value), (key, value)]

For lab 8:

all_quakes = []

all_quakes.append(quake)

[{k:v, k:v, k:v}, {}, {}]

So you can grab quakes one by one using a for loop & process each one

Exercise: breakout room: 5 min

Problem 2: Write a function to find the most frequently-appearing character in a given string. If there is a tie, return any one of the most frequent characters. You may use the function you wrote in Problem 1.

```
def strmode(in_str):
    """ Return the most frequently-appearing character in
```

Step 1: build dictionary

function you wrote in Problem 1.

```
def strmode(in_str):
    """ Return the most frequently-appearing character in
    in_str, or any of the most frequent characters in case of a
    tie. Precondition: in_str is a string with nonzero length.
    Examples: strmode('hahah') # => 'h'
              strmode('who') # => could return 'w', 'h', or 'o'
    """
```

Step 1: build a dictionary of all letters & their frequency

step 2: find the max in the dictionary & return it

PythonScratchpad.py

```
def strmode(in_str):

    """
    """
    all_letters = {}
    for letter in in_str:
        if letter in all_letters.keys():
            all_letters[letter] = all_letters[letter] + 1
        else:
            all_letters[letter] = 1

    max_letter = "?"
    max_letter_count = 0

    for key, val in all_letters.items():
        if val > max_letter_count:
            max_letter = key
            max_letter_count = val

    return(max_letter)

print(strmode("hahaaaaaa"))
```

w22skip

Quick Demo

```
training_set = [ ]
record1 = {"teeth": 2, "legs": 4, "tails": 3}
record2 = {"teeth":400, "legs":2}
training_set.append(record1)
training_set.append(record2)


for record in training_set:
    print(record["teeth"])
    teeth = record.get("teeth")
    tails = record.get("tails", 0)
    print(tails)
```

get returns "none" if not there

if you use [] it crashes

Dictionaries, Functions, challenges

Thursday, March 4, 2021 12:59 PM

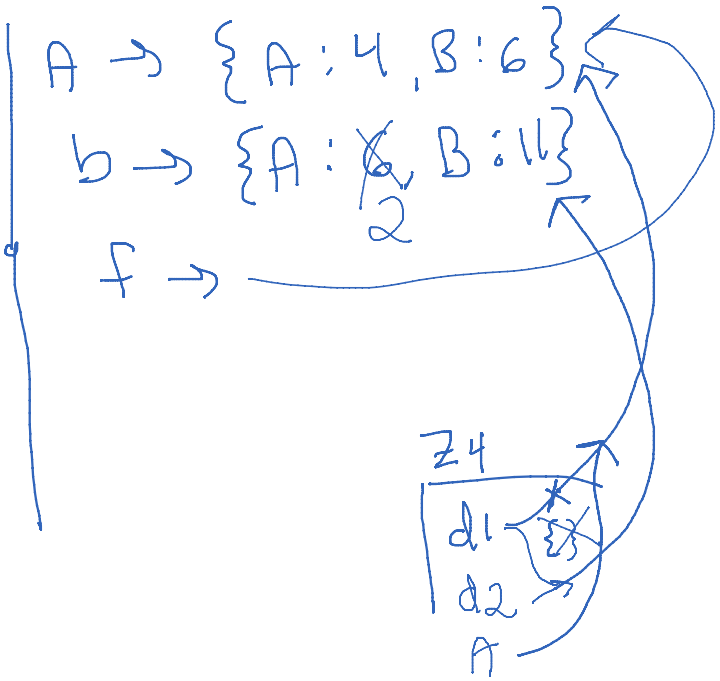
3 min breakout room

challenge

```
def z4(d1, d2):
    a = d1
    d1 = {}
    d1 = d2
    d1["A"] = 2
    return a

a = {"A": 4, "B": 6}
b = {"A": 6, "B": 11}
f = z4(a, b)
print(a["A"], b["A"], f["A"])
```

4 2 4



Then show python tutor

```
def z4(d1, d2):
    a = d1
    d1 = {}
    d1 = d2
    d1["A"] = 2
    return a

a = {"A":4, "B":6}
b = {"A":6, "B":11}

f = z4(a,b)
print(a["A"], b["A"], f["A"])

--------------------------------
def z0(y):
    y[0] = 4
    return y

b = [5,6]
c = z0(b)
print(b[0], c[0])
```