

23 Lexographic ordering, .get, more i/o

Wednesday, March 3, 2021 2:26 PM

Lexographic ordering (alphabetical)

"a" < "b"

"ab" < "ac"

"a" < "aa" Shorter comes 1st

" " < "a"

"A" < "a"

Bellingham vs Bellevue
 i > e (later in alphabet)
 So Bham > Bvue

True/false

Challenge



1: "To be or not to be".find("be") == 13

2: "Boo".replace("o", "O").lower() <= "boo"

3: "no" in "To be or not to be"

4: "stark" not in "Tony Stark"

w21
start +ve

Demo

like strings:

foo = "hello"
 bar = list(foo) $\Rightarrow [h, e, l, l, o]$ Can we go the other way?
~~bar = str(bar)~~ \Rightarrow "hello" ~~Nope~~
~~zab = " ".join(bar) \Rightarrow "[h, e, l, l, o]"~~

W21
start five

Demo

like strings:

foo = "hello"
bar = list(foo) $\Rightarrow [h, e, l, l, o]$ Can we go the other way?
~~baz = str(bar)~~ \Rightarrow 'hello' *Nope*
 $\therefore \text{zab} = \underset{\text{Separator, here it's nothing, but could be comma, etc}}{\text{"."}}.\text{join}(\text{bar}) \Rightarrow ["h", "e", "l", "l", "o"]$

You can:

- Index
- Slice
- len function
- in, not in operators
- + and * operators

i = $[n, a] * 16 + \text{"batman"}$

Dictionaries .get

Wednesday, May 25, 2022 9:55 AM

```
1 list1 = {0:1,2:3}
2
3 #we can get values out like this:
4 print(list1[2])
5
6 #but we can also use get!
7 print(list1.get(2))
8
9 #ok sure, but why?
10 #well, what if you try this?
11 print(list1[4])
12
13 #yeah that crashes.
14 #but if we'd done get...
15 print(list1[4])
16
17 #so less crashing!
18 #that's not all! you can also specify default values
19 print(list1.get(4, "not in database"))
20
21 print(list1.get(2, -1))|
```

This mode is experimental. Use the regular Python Tutor to [get live help](#) and use more features.

Dictionaries Demo
L-23

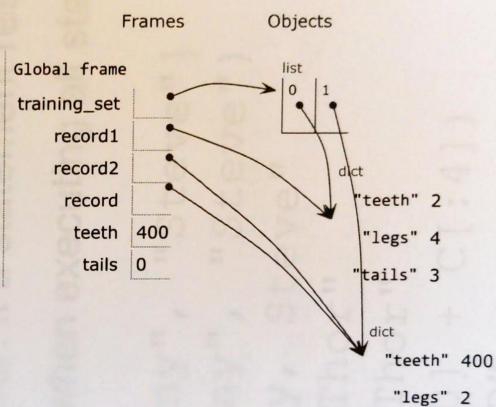
Write code in Python 3.6 ▾

(drag lower right corner to resize code editor)

Print output (drag lower right corner to resize)

```
1 training_set = [ ]
2 record1 = {"teeth": 2, "legs": 4, "tails": 3}
3 record2 = {"teeth":400, "legs":2 }
4 training_set.append(record1)
5 training_set.append(record2)
6
7 for record in training_set:
8     print(record["teeth"])
9
10    teeth = record.get("teeth")
11    tails = record.get("tails", 0)
12
13    print(tails)|
```

```
2
3
400
0
```



— line that just executed
→ next line to execute

<< First < Prev Next > > Last

Done running (16 steps)

```
1 shirt_a = {"color": "red", 0:2, 2:4, 4:6, 6:1, 8:10, 10:100, 12:14}
2 shirt_b = {"color": "blue", 0:3, 2:6, 4:3, 6:8, 8:11, 10:200, 12:140}
3 shirt_c = {"color": "puce", 0:4, 2:3, 4:7, 6:2, 8:13, 10:204, 12:10}
4
5 all_shirts = [shirt_a, shirt_b, shirt_c]
6
7 #print inventory for all shirts for size 10
8 #red: 100
9 #blue: 200
10 #puce: 204
11 to_print = ""
12 for this_shirt in all_shirts:
13     to_print += this_shirt.get("color") + " " + str(this_shirt.get(10)) + "\n"
14
15 print(to_print)
```

Shirt_inventory file I/O and list of dictionaries

Friday, May 27, 2022 3:09 AM

https://www2-my.sharepoint.com/:g/personal/hardinc3_nwu_edu/ef15b_nsJNmXopcoA4tjYBmrePaKTLrwuM2SbjZEyw?e=vIFFFF

| A | B | C | D | E | F | G | H |
|---------|---|---|---|---|----|-----|----|
| 1 color | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
| 2 red | 2 | 4 | 6 | 1 | 10 | 100 | 14 |
| 3 puce | 4 | 3 | 7 | 2 | 13 | 204 | 10 |

```
1 #shirt_a = {"color": "red", 0:2, 2:4, 4:6, 6:1, 8:10, 10:100, 12:14}
2 #shirt_c = {"color": "puce",0:4, 2:3, 4:7, 6:2, 8:13, 10:204, 12:10}
3 #all_shirts = [shirt_a, shirt_c]
4 #KEY_NAMES = ["color", 0,2,4,6,8,10,12]
5
6 all_shirts = []#[shirt_a, shirt_c] #[shirt_a, shirt_b, shirt_c]
7 KEY_NAMES= []
8
9 with open("shirt_inventory.csv", "r") as infile:
10     #first line is header
11     first_line = True
12     for line in infile:
13         line = line.strip(" \n")
14         |
15         line_list = line.split(",")
16         line_list = line_list[:-1]
17
18         if first_line:
19             for key in line_list:
20                 if len(key) > 0:
21                     KEY_NAMES.append(key)
22             first_line = False
23         else:
```

```

25     new_shirt = {}
26     for i in range(len(line_list)):
27         #print(i, KEY_NAMES[i])
28         new_shirt[KEY_NAMES[i]] = line_list[i]
29         #print(new_shirt[KEY_NAMES[i]])
30     all_shirts.append(new_shirt)
31 #
32 #read in other lines
33
34
35 #detect if three even numbered inventories in a row for one shirt color
36 pattern = 0
37 for shirt in all_shirts:
38     for size in list(shirt.keys())[1:]:
39         print("size:", size, "number", shirt[size])
40     if int(shirt[size]) % 2 == 0:
41         pattern += 1
42     else:
43         pattern = 0
44     if pattern == 3:
45         print("three in a row! in ", shirt["color"])
46 pattern = 0
47
48 with open("shirt_inventory_out.csv", "w") as outfile:
49     #headings
50     for key in KEY_NAMES:
51         outfile.write(str(key) + ",")
52     outfile.write("\n")
53
54     for shirt in all_shirts:
55         for value in shirt:
56             outfile.write(str(shirt[value])+",")
57
58
59
60 #print inventory for all shirts for size 10
61 #red: 100
62 #blue: 200
63 #puce: 204
64 to_print = ""
65 for this_shirt in all_shirts:
66     to_print += this_shirt.get("color", "none found") + " " + str(this_shirt.get(10)) + "\n"
67
68 print(to_print)

```

Finding the biggest horse

Friday, November 18, 2022 2:02 PM

```
1 all_horses = []
2 KEY_NAME =[]
3 with open('horses.csv', 'r') as infile:
4
5     first_line = True
6     for line in infile:
7         line = line.strip("\n")
8
9         line_list = line.split(",")
10        #print(line_list) #todo for debugging
11        #line_list = line_list[:-1]
12        for l in range(len(line_list)):
13            #print(l)
14
15            line_list[l] = line_list[l].strip()
16        #print(line_list) #todo for debugging
17
18        if first_line: #first line has the key values
19            # print(line_list)
20
21            for key in line_list:
22                KEY_NAME.append(key)
23            first_line = False
24            KEY_NAME[0] = KEY_NAME[0][1:]
25        else:
26            this_horse = {}
27            for i in range(len(line_list)):
28                this_horse[KEY_NAME[i]] = line_list[i]
29            # print(this_horse)
30            all_horses.append(this_horse)
31
32        #print(all_horses)
33        #print(KEY_NAME)
34
35    #make weights integers
36    for h in all_horses:
37        h["weight"] = int(h["weight"])
38    print(all_horses)
39
40    #find biggest horse
41    biggest = ""
42    weight = 0
43    for h in all_horses:
44        if h["weight"] > weight:
45            biggest = h["name"]
46
47    print(biggest)
48
```

LGBTQ+ Western proudly presents: Trans Week

These sessions are open for all, including students, faculty, staff, and community members centering our trans and nonbinary community at Western.



Follow us on Instagram to stay updated on events including partnerships with student organizations @lgbtqwestern



Trans Week Events:

Grab & Go Clothing Swap:

M Nov 14th, 2-5 pm VU 723 | LGBTQ+ Western Office

Trans Tea: T Nov 15th, 4-5pm VU 462

Trans Takeover Open Mic Night: T Nov 15th, 7-9 pm, Viking Union Underground Coffee House, 3rd floor

Black LGBTQ+ Thriving Collective Kickoff & Jeopardy:

W Nov 16th, 6-8pm VU Multicultural Center Kitchen

Info Table and Story Sharing:

Th Nov 17th, 11-2 VU Lobby near Bookstore

Boundaries Workshop with Counseling & Wellness:

Th Nov 17th, 3-4:30pm VU Multicultural Center 735

Trans Day of Remembrance Vigil:

Fri Nov 18th, 10:30-12:30pm

VU Multicultural Center 735

Co-Program with Roots: Adoption Month Celebration

Fri Nov 18th VU MPR

10-3 (Presentation about LGBTQ+ Adoption at 1pm)



AA/EQ

For disability accommodations and event questions,
please contact lgbtq@wwu.edu