CIS 1100

Dictionaries

Python
Fall 2024
University of Pennsylvania

Reminder about Autograder Output

- Gradescope has automatic style deductions
- Take a look sometime today to make sure that you have no automatic style errors!
- Other things that are **your responsibility to check for:**
  - submitting all files
  - compilation issues
- Not an autograder thing, but also remember:
  - When we say pennkey, we mean e.g. `sharry` or `tqmcgaha`
  - In the readme, you have to copy the collaboration statement exactly

If you see 0/40, that's not something you should ignore!

Reminder about OHQ

- Look at the course calendar for TA office hours
  - you do not make an appointment ahead of time
  - when you show up, go to ohq.io and sign yourself up to get help
  - make sure that you select the proper queue (Python!!)

Exam Reminders

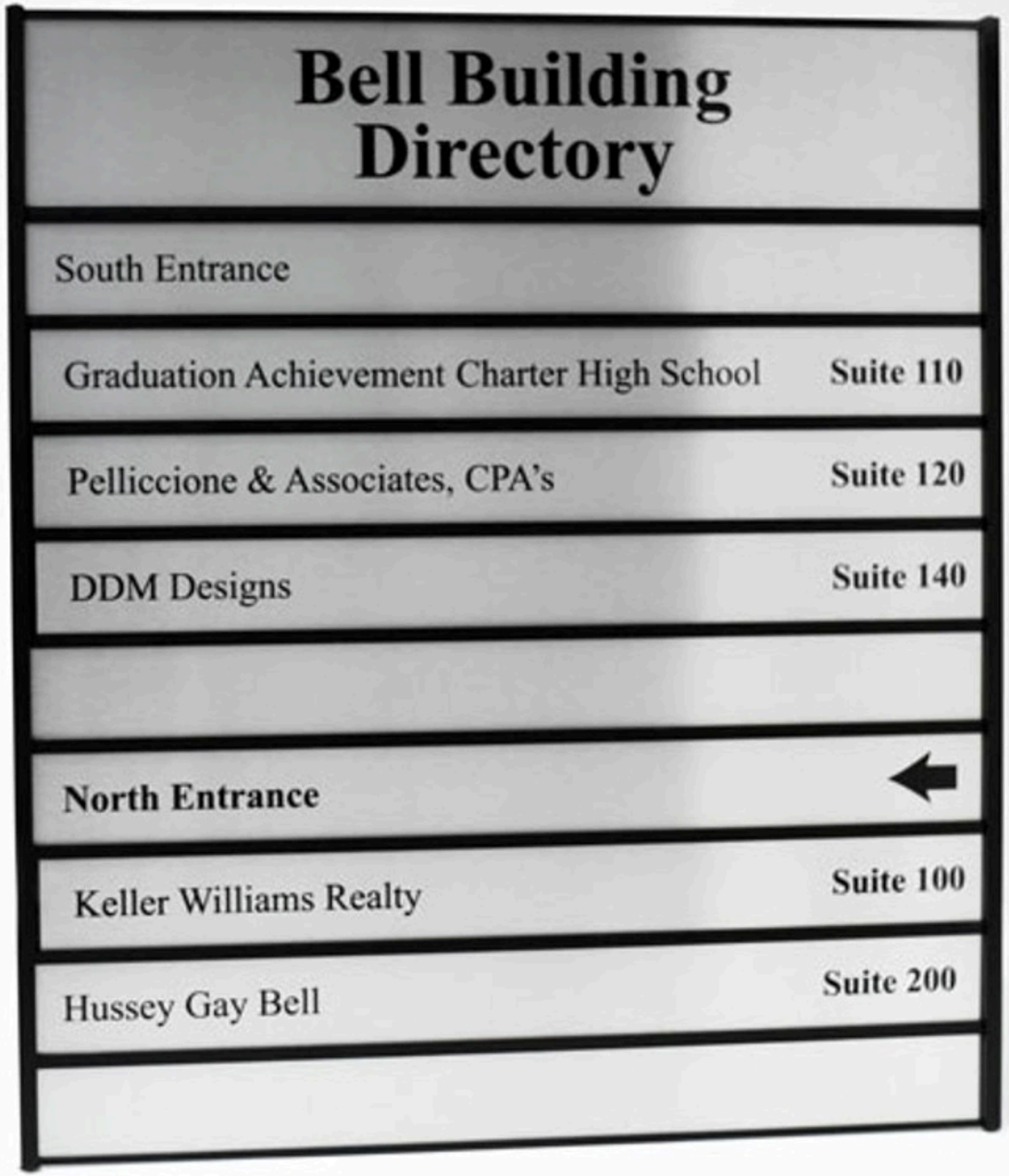
- Plan to take your exam during the section for which you are registered.
- Take a practice exam once you're done with *Caesar*.
- All students who require SDS accommodation to take the exam should schedule their exam through the Weingarten Testing Center ASAP.
  - Any time on October 9th is acceptable.
- The exam only covers material up until sets. Material covered today and Monday will be covered on HW04 and Exam 2.

Recap

Dictionaries associate keys with values.

When you're trying to find your doctor in an office using a building directory, the names are _____ and the rooms are _____ (M1)

- A: keys, values
- B: keys, keys
- C: values, values
- D: values, keys



Describe the Issue:

We want to write a function that turns a list of names into a dictionary where the *keys* are the first initials and the *values* are lists of names that start with that letter.

```
["Susan", "Sally", "Paul", "Nico"]
⬇️
{"S" : ["Susan", "Sally"],
 "P" : ["Paul"],
 "N" : ["Nico"]
}
```

What's wrong with this implementation? (L11; describe, don't fix)

```python
def group_names_by_letter(names):
    names_by_letter = dict()
    for name in names:
        first_letter = name[0]
        names_by_letter[first_letter].append(name)
    return names_by_letter
```

Worked Example: Books!

I'm vain, so we're going to use my personal data: my collection of books read from Goodreads. (follow me?)

We'll use this data to build a recommender system

- "I heard about this author, can you recommend me her best book?"
- "What's the best book from last year?"

Understanding the Data

What's going on here? What do we have to work with?

```
Voices in the Evening
Natalia Ginzburg
1952 170 3.76
The Dry Heart
Natalia Ginzburg
1947 88 3.99
Childhood / Youth / Dependency (The Copenhagen Trilogy, #1-3)
Tove Ditlevsen
1967 371 4.36
In the Eye of the Wild
Nastassja Martin
2019 128 3.96
Kudos
Rachel Cusk
2018 236 3.91
Jack (Gilead, #4)
Marilynne Robinson
2020 309 3.86
```

Understanding the Data

For each data point (Book), we have:

- title
- author
- year, page count, rating

How many lines of the file do we need to read to process all the data for a single book? (S7)

```
Voices in the Evening
Natalia Ginzburg
1952 170 3.76
The Dry Heart
Natalia Ginzburg
1947 88 3.99
Childhood / Youth / Dependency (The Copenhagen Trilogy, #1-3)
Tove Ditlevsen
1967 371 4.36
In the Eye of the Wild
Nastassja Martin
2019 128 3.96
Kudos
Rachel Cusk
2018 236 3.91
Jack (Gilead, #4)
Marilynne Robinson
2020 309 3.86
```

Understanding the Data

For each data point (Book), we have:

- title
- author
- year, page count, rating

What are the types for each of these properties? (S8)

```
Voices in the Evening
Natalia Ginzburg
1952 170 3.76
The Dry Heart
Natalia Ginzburg
1947 88 3.99
Childhood / Youth / Dependency (The Copenhagen Trilogy, #1-3)
Tove Ditlevsen
1967 371 4.36
In the Eye of the Wild
Nastassja Martin
2019 128 3.96
Kudos
Rachel Cusk
2018 236 3.91
Jack (Gilead, #4)
Marilynne Robinson
2020 309 3.86
```

Understanding the Data

For each data point (Book), we have:

- title
  - `str`
- author
  - `str`
- year, page count, rating
  - `int`, `int`, `float`

Process Books

```
def process_book_file(filename):
    """Returns a dictionary d mapping
    book titles to tuples of book information.
    """
    ...
    return d
```

What is the type of `d`? (S9)

Process Books

```
def process_book_file(filename):
    """Returns a dictionary d mapping
    book titles to tuples of book information.
    """
    ...
    return d
```

What are the types of `d`'s keys? What are the types of its values? **Be as detailed as possible.** (S10)

Process Books

```
def process_book_file(filename):
    """Returns a dictionary d mapping
    book titles to tuples of book information.
    """
    ...
    return d
```

Implement `process_book_file` in (C12)

Together: Print all books by author

```
def find_by_author(d, author):
    """Return a dictionary containing
    just those books by the given author.
    """
    ...
```

You: Find Best Book by Author

```
def best_by_author(d, author):
    """Return the name of the book that
    has the highest rating by that author.
    """
    ...
```

(C14)