Objectives

- Continuing with dictionaries

1

Review: Dictionaries

- What is a dictionary in Python?

- What is the syntax for creating a new dictionary?

- How do we access a key's value from a dictionary? (2 ways)
  - ➢ What happens if there is no mapping for that key?

- How do we create a key →value mapping in a dictionary?

- How can we iterate through a dictionary?

- What other operations can we perform on dictionaries?

- Review/think through the "Towards a Solution" slide on the handouts

2

Review: Creating Dictionaries in Python

Syntax:

`{<key>:<value>, …,`
`  <key>:<value>}`

```
empty = {}
charToAscii = { 'a':97, 'b':98, …, 'z':122 }
```

3

Review: Dictionary Operations

Indexing	`<dict>[<key>]`
Length (# of keys)	`len(<dict>)`
Iteration	`for <key> in <dict>:`
Membership	`<key> in <dict>`
Deletion	`del <dict>[<key>]`

Unlike strings and lists, doesn't make sense to do slicing, concatenation, repetition for dictionaries

4

Review: Dictionary Methods

Method Name	Functionality
`<dict>.clear()`	Remove all items from dictionary
`<dict>.keys()`	Returns a copy of dictionary's keys (a set-like object)
`<dict>.values()`	Returns a copy of dictionary's values (a set-like object)
`<dict>.get(x [, default])`	Returns `<dict>[x]` if x is a key; Otherwise, returns None (or default value)

5

Review: Accessing Values Using Indexing

- Syntax:
  `<dictionary>[<key>]`
- Examples:
  ```
  charToAscii['z']

  nameToPhoneNum['friendname']
  ```

- **KeyError** if key is not in dictionary
  - ➤ Runtime error; exits program

6

3

Review: Adding/Modifying Key-Value Pairs

- Syntax:

  `<dictionary>[<key>] = <value>`

- Example:

  `nameToPhoneNum['registrar'] = 8455`

  ➤ Adds mapping for `'registrar'` to 8455

  OR

  ➤ If mapping already existed, *modifies* old mapping to 8455

7

Review: Problem

Example file:

- Given a file (data/roster.dat) of the form

  `<firstname> <gradyear>`

- Goal: quickly find the classyear of a particular student

  ➤ Specifically, want to
  - Repeatedly prompt user for a first name of a student (given)
  - Display that student's graduation year

```
Whose class year? Bobby
Bobby is in the class of 2024
```

- Consider
  ➤ How would we solve this before learning dictionaries?
  ➤ How would we solve this with dictionaries?
    - What is the key? What is the value?
  ➤ If that dictionary existed, how would we implement the user input part?
  ➤ How do we parse the file to create the dictionary?

  `years_dictionary.py`

8

Towards a Solution

`years_dictionary.py`

- Data Structure: dictionary
  - Key: name
  - Value: class year
- Part 1: Pretend already have the dictionary
  - Repeatedly prompt user for the first name of the student
  - Display the student's graduation year
- Part 2: Parse the file to generate the dictionary
  - Given a file of the form
    - \<firstname> \<gradyear>

9

Parsing Algorithm

- Create an empty dictionary
- Read in the file line by line
  - Split the line
  - From the split, get the name and the year
  - Add a mapping of the name to the year in the dictionary
    - (*accumulate* the data in the dictionary)
- Process the data in the dictionary, e.g.,
  - Display it, in sorted order (to verify it)
  - Integrate into user input part to get answers

10

Another Problem

Example file:

- Given a file of the form
  - \<firstname\> \<classyear\>
- Goal: Report the *number* of students in each graduation year
  - How do we want to model the data?
  - What is the key? What is the value?
- Problem-solving Approach:
  - Pretend you are the computer, how would you solve this problem?

11

Equivalent Solutions:
A Dictionary of Accumulators

```
if myKey not in myDictionary :
    myDictionary[myKey] = 1
else:
    count = myDictionary[myKey] + 1
    myDictionary[myKey] = count
```

```
if myKey not in myDictionary :
    myDictionary[myKey] = 1
else:
    myDictionary[myKey] += 1
```

12

6

Looking Ahead

- Pre Lab 9
  - Dictionaries
  - Classes
  - Fewer exercises, fewer opportunities to confirm your understanding
- Friday – Exam 2
  - Preparation document online

13