

Objectives

- Dictionaries

March 16, 2022

Sprenkle - CSCI111

1

1

Lab Preparation Suggestions

- Review frequently
 - Learning a new language
 - Better to have some practice every day (rather than every week)
- Review example programs
 - Do you [still] understand them after class?
- Active work in interactive text book
 - Don't just click the boxes
- Focus is on the current week, but we are using tools we learned in the last ~8 weeks.

March 16, 2022

Sprenkle - CSCI111

2

2

LOOKUP ALTERNATIVES

March 16, 2022

Sprenkle - CSCI111

3

3

List/String Lookup

- How do we “lookup” a value in a list or a character in a string?
- Answer:
 - By its index/position
- Requires:
 - Knowing the index where a value is

March 16, 2022

Sprenkle - CSCI111

4

4

Alternative Lookup

- Alternative: look up something by its key
 - Example: When I lookup my friend's phone number in my contacts, I don't know that the number is at position X in my contacts. I look up my friend's number by her *name*.
 - Need a fast way to figure out "given this *key*, what is the *value* associated with it?"
- This type of data structure is known as a **dictionary** in Python
 - Maps a **key** to a **value**
 - Contacts' key: name; value: phone number

March 16, 2022

Sprenkle - CSCI111

5

5

Examples of Dictionaries

Dictionary	Keys	Values
Dictionary		
Textbook's index		
Cookbook		
URL (Uniform Resource Locator)		

- Any other things we've done/used in class?

March 16, 2022

Sprenkle - CSCI111

6

6

Examples of Dictionaries

Dictionary	Keys	Values
Dictionary	Word	Definition
Textbook's index	Keyword	Page number
Cookbook	Food type	Recipes
URL (Uniform Resource Locator)	URL	Web page

- Any other things we've done/used in class?

March 16, 2022

Sprenkle - CSCI111

7

7

Examples of Dictionaries

- Real-world:
 - Dictionary
 - Textbook's index
 - Cookbook
 - URL (Uniform Resource Locator)
- Examples from class
 - Variable name → value
 - Function name → function definition
 - ASCII value → character

March 16, 2022

Sprenkle - CSCI111

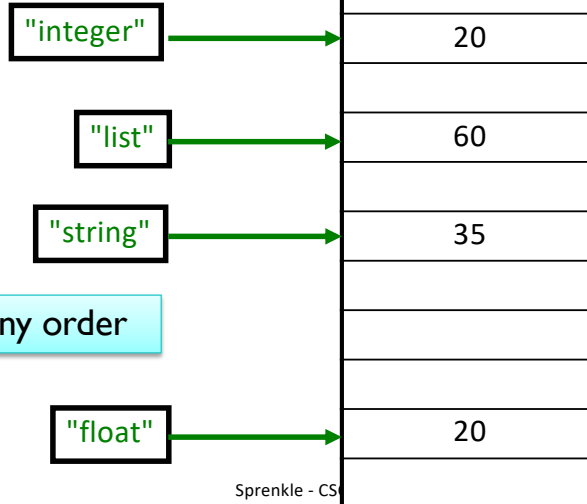
8

8

Example: A Textbook's Index

topicToPageNumber

Values



Lots of empty space to add new entries

Keys are not in any order

Keys are unique; values are not necessarily unique

March 16, 2022

Sprenkle - CS

9

9

Dictionaries in Python

- Map **keys** to **values**
 - Keys are probably **not** alphabetized
 - Mappings are from **one** key to **one** value
 - Keys are **unique**, Values are not necessarily unique
 - Example: student id → last name
 - Keys must be **immutable** (numbers, strings)
- Similar to Hashtables/Hashmaps in other languages

How would we handle if there is more than one value for a given key?

March 16, 2022

10

10

Creating Dictionaries in Python

Syntax:

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

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

March 16, 2022

Sprenkle - CSCI111

11

11

Dictionary Operations

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

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

March 16, 2022

Sprenkle - CSCI111

12

12

Accessing Values Using Indexing

- Syntax:

`<dictionary>[<key>]`

- Examples:

```
charToAscii['z']
```

```
nameToPhoneNum['friendname']
```

- **KeyError** if key is not in dictionary

- Runtime error; exits program

March 16, 2022

Sprenkle - CSCI111

13

13

Dictionary Methods

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

March 16, 2022

Sprenkle - CSCI111

14

14

Accessing Values Using get Method

- Syntax: `<dict>.get(x [,default])`
 - Semantics: Returns `<dict>[x]` if `x` is a key
Otherwise, returns `None` (or default value)

- Examples:

```
charToAscii.get('z')  
nameToPhoneNum.get('friendname')
```

- If no mapping, **None** is returned instead of **KeyError**

March 16, 2022

Sprenkle - CSCI111

15

15

Accessing Values: Look Before You Leap

- Typically, you will check if dictionary has a key before trying to access the key

```
if 'friend' in nameToPhoneNum :  
    number = nameToPhoneNum['friend']
```

Know mapping exists
before trying to access

- Or handle if `get` returns default

```
number = nameToPhoneNum.get('friend')  
if number is None:  
    # do something ...
```

No phone number exists

March 16, 2022

Sprenkle - CSCI111

16

16

Recall: Special Value **None**

- Special value we can use
 - E.g., Return value from function when there is an error
- If you execute

```
list = list.sort()
print(list)
```

 - Prints **None** because `list.sort()` does **not return** anything

March 16, 2022

Sprenkle - CSCI111

17

17

Example Using **None** as an Error

```
def encryptLetter( letter, key ):
    """
    Pre: letter is a single lowercase letter, ...
    returns the lowercase letter encoded by the key.
    If letter is not a lowercase letter, returns None
    """
    if letter < 'a' or letter > 'z':
        return None
    #As usual ...
```

```
# example use
encLetter = encryptLetter(char, key)
if encLetter is None:
    print("Can't encrypt character", char, "in message: ")
```

March 16, 2022

Sprenkle - CSCI111

18

18

Inserting Key-Value Pairs

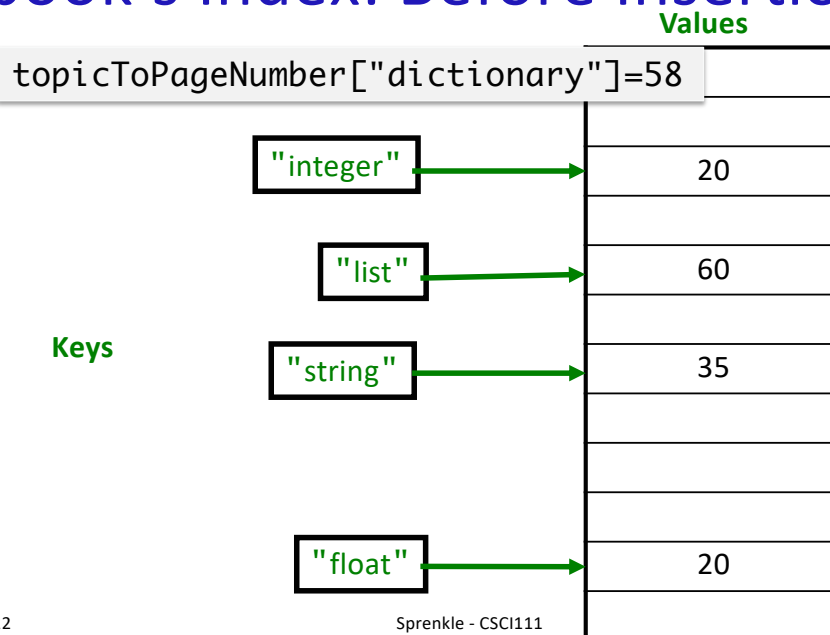
- Syntax:

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

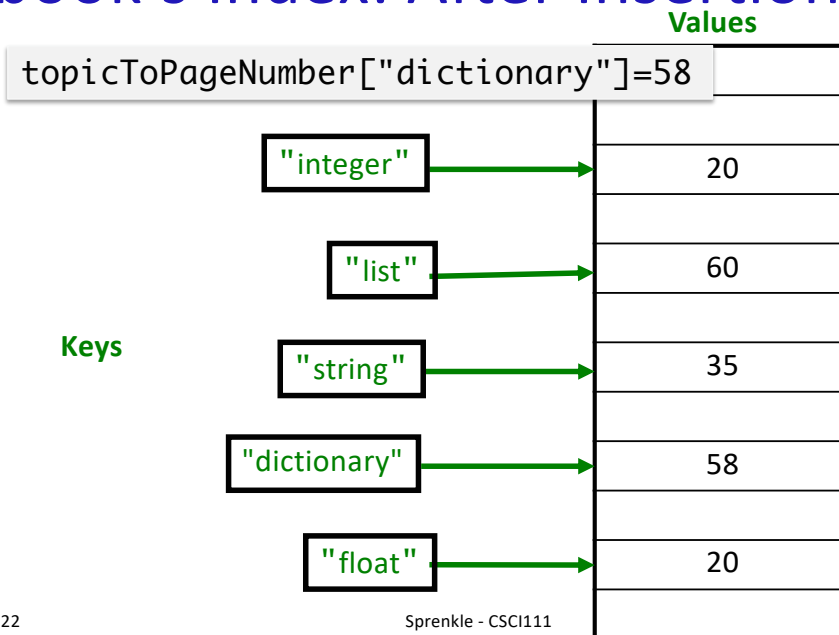
- `charToAscii['a'] = 97`

➤ Creates new mapping of 'a' → 97

Textbook's Index: Before Insertion



Textbook's Index: After Insertion



March 16, 2022

Sprenkle - CSCI111

21

21

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

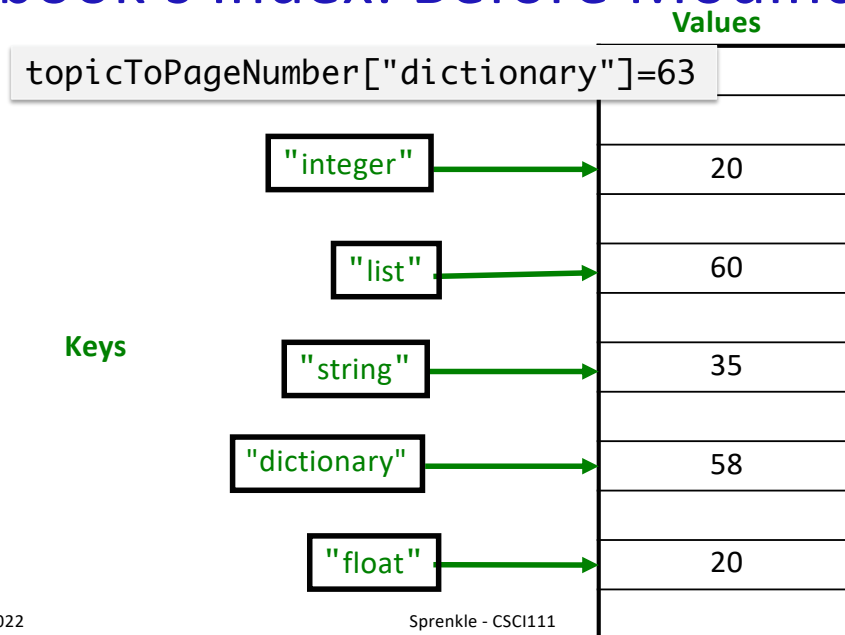
March 16, 2022

Sprenkle - CSCI111

22

22

Textbook's Index: Before Modification



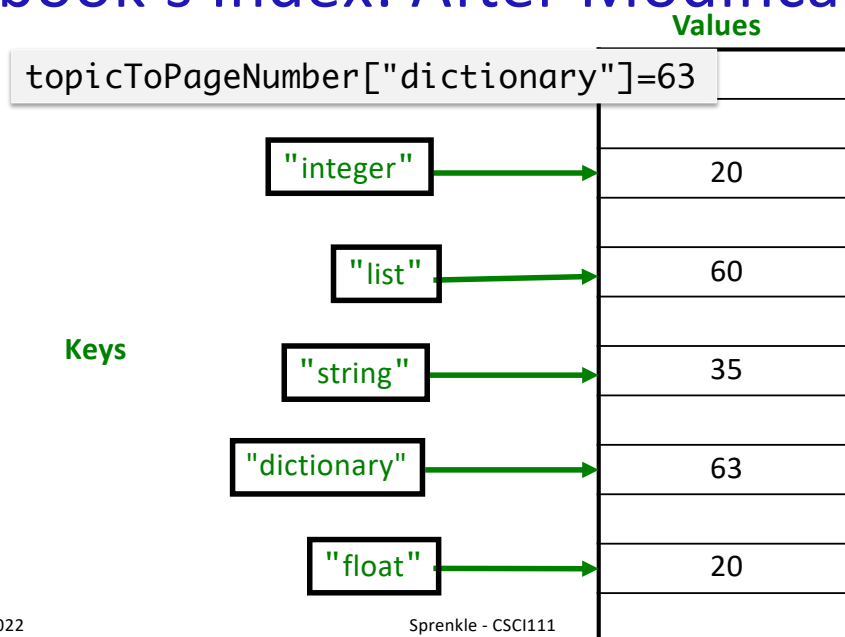
March 16, 2022

Sprenkle - CSCI111

23

23

Textbook's Index: After Modification



March 16, 2022

Sprenkle - CSCI111

24

24

Methods `keys()` and `values()`

- Don't return a list object
- But can be used similarly to a list
- If you want to make them into a list, use list converter:

```
keys = list(mydict.keys())
```

March 16, 2022

Sprenkle - CSCI111

25

25

Using Dictionaries

`using_dictionary.py`

- Demonstrates lots of operations, methods, etc. in using dictionaries

March 16, 2022

Sprenkle - CSCI111

26

26

Problem

- Given a file (data/roster.dat) of the form
 <firstname> <gradyear>
- Goal: quickly find out what a particular student's class is.
- Specifically,
 - Repeatedly prompt user for a first name of a student (given)
 - Display that student's graduation year
- Consider
 - How would we solve this pre-dictionaries?
 - How would we solve this with dictionaries?
 - What is the key? What is the value?

March 16, 2022

Sprenkle - CSCI111

[years_dictionary.py](#)

27

27

Solutions: Pre Dictionaries

- Lots of possibilities
- One possibility:
 - Read through the file, looking for name; stop when found
- Another possibility:
 - Create two lists: one for first names, one for class years
 - Read the file, split each line of the file, add the first name and class year to the appropriate lists
 - Find the first name in the list → index of element in list
 - Use that index to find the class year in the other list

March 16, 2022

Sprenkle - CSCI111

28

28

Analyzing Pre-dictionaries Solution

- Not ideal because
 - Reading file multiple times
 - Keeping track of two lists
 - If remove/add people, need to add/remove from both lists to keep in sync
 - find is a relatively expensive operation
 - Has to look through each element: “Are you my element?” until find the match

March 16, 2022

Sprenkle - CSCI111

29

29

Looking Ahead

- Lab 8 due Friday
- BI: Cryptocurrency due Thursday
- Today's Office Hours: 1:45-3 p.m.
 - Email me with questions

March 16, 2022

Sprenkle - CSCI111

30

30