

Objectives

- While Loops
- Dictionaries

Mar 13, 2017

Sprenkle - CSCI111

1

Review

- What is a definite loop? What is an indefinite loop?
- What is the syntax for writing an indefinite loop?
- Which is more powerful—a **for** loop or a **while** loop? Why?

Mar 13, 2017

Sprenkle - CSCI111

2

while Loops: comparing use of **break**

```
# condition says when loop # have to look inside loop to
# will continue           # know when it stops
x=eval(input("Enter number:")) while True :
while x % 2 != 0 :        x = eval(input("Enter number:"))
    print("Error!")      if x % 2 == 0 :
    x = eval(input("Enter # break "breaks" out of a loop
    number: "))         print("Error!")
print(x, "is an even number.") print(x, "is an even number.")
```

Says when to keep going

Says when to stop

Using break statements:
Best when loop has to
execute at least once.

Mar 13, 2017

Sprenkle - CSCI111

3

Flipping Coins

- Problem: How many flips does it take to get 3 consecutive heads?

[consecutiveHeads.py](#)

Mar 13, 2017

Sprenkle - CSCI111

4

Summary: While vs. For Loops

- Any **for** loop can be translated into a **while** loop
 - But **not** vice versa
- **while** loops are more **powerful** than **for** loops
 - Give an example of a **while** loop that can't be converted to a **for**

Mar 13, 2017

Sprenkle - CSCI111

5

LOOKUP ALTERNATIVES

Mar 13, 2017

Sprenkle - CSCI111

6

List/String Lookup

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

Mar 13, 2017

Sprenkle - CSCI111

7

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 can look up my friend's number by her *name*.
 - Have 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: “Friend's name”, value: phone number

Mar 13, 2017

Sprenkle - CSCI111

8

Examples of Dictionaries

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

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

Mar 13, 2017

Sprengle - CSCI111

9

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?

Mar 13, 2017

Sprengle - CSCI111

10

Examples of Dictionaries

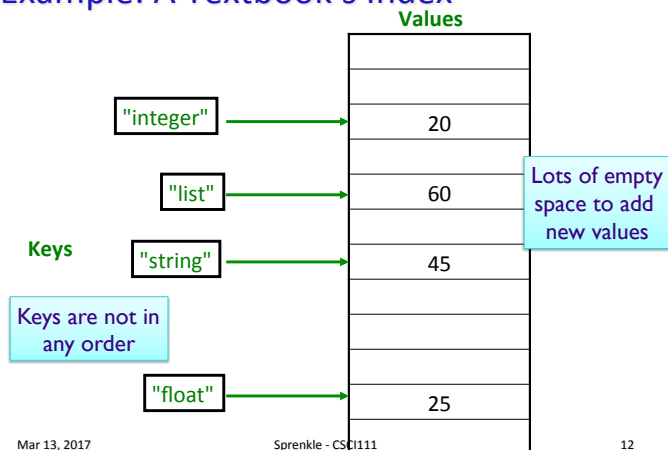
- Real-world:
 - Dictionary
 - Textbook's index
 - Cookbook
 - URL (Uniform Resource Locator)
- Examples from class
 - Variable name → value
 - Function name → function definition
 - ASCII value → character

Mar 13, 2017

Sprengle - CSCI111

11

Example: A Textbook's Index

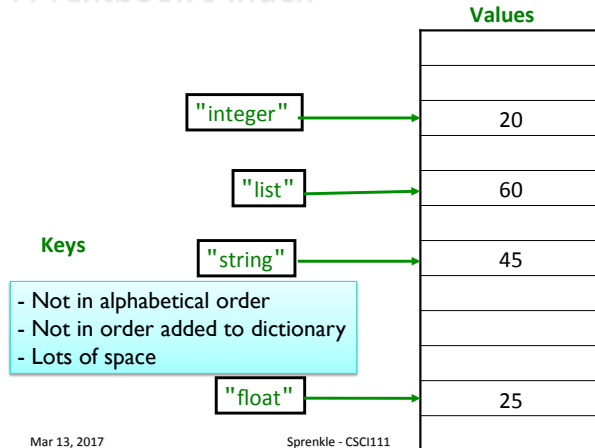


Mar 13, 2017

Sprengle - CSCI111

12

A Textbook's Index



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?

Mar 13, 2017

Sprenkle - CSCI111

14

Creating Dictionaries in Python

Syntax:

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

```
empty = {}  
ascii = { 'a':97, 'b':98, 'c':99, ..., 'z':122 }
```

Mar 13, 2017

Sprenkle - CSCI111

15

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

Mar 13, 2017

Sprenkle - CSCI111

16

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)

Mar 13, 2017

Sprenkle - CSC1111

17

Accessing Values Using Keys

- Syntax:
`<dictionary>[<key>]`
- Examples:

```
ascii['z']  
contacts['friendname']
```

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

Mar 13, 2017

Sprenkle - CSC1111

18

Accessing Values Using `get` Method

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

```
ascii.get('z')  
directory.get('friendname')
```

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

Mar 13, 2017

Sprenkle - CSC1111

19

Accessing Values

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

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

Know mapping exists
before trying to access

- Or handle if `get` returns default

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

No phone number exists

Mar 13, 2017

Sprenkle - CSC1111

20

Recall: Special Value None

- Special value we can use
 - E.g., Return value from function when there is an error
- Similar to **null** in Java
- If you execute

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

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

Mar 13, 2017

Sprenkle - CSC1111

21

Example Using None

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

```
# example use
encLetter = translateLetter(char, key)
if encLetter is None:
    print("Error in message: ", char)
```

Mar 13, 2017

Sprenkle - CSC1111

22

Inserting Key-Value Pairs

- Syntax:

```
<dictionary>[<key>] = <value>
```
- `ascii['a'] = 97`
 - Creates new mapping of 'a' → 97

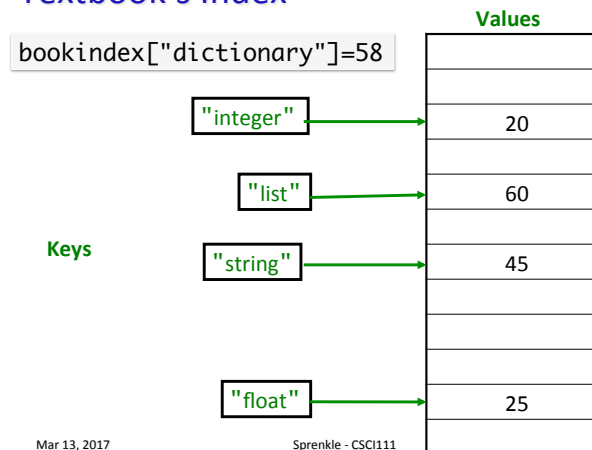
`ascii_dictionary.py`

Mar 13, 2017

Sprenkle - CSC1111

23

Textbook's Index



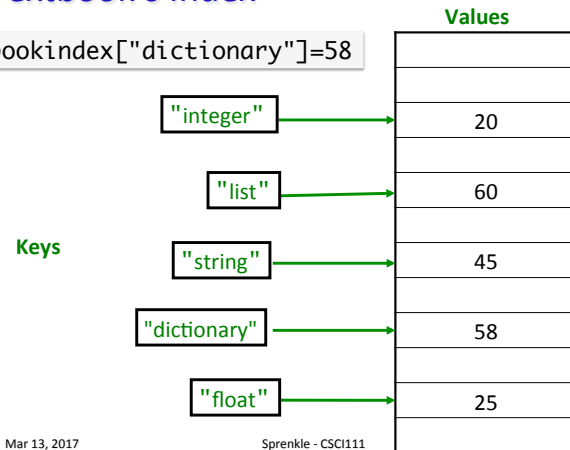
Mar 13, 2017

Sprenkle - CSC1111

24

Textbook's Index

```
bookindex["dictionary"]=58
```



Mar 13, 2017

Sprenkle - CSCI111

25

This Week

- Lab 8
- SSA on Friday: no Broader Issue
- Exam 2 next Friday

Mar 13, 2017

Sprenkle - CSCI111

26