

Objectives

- Exam on Friday
- Wrap up lists, dictionaries
- Default parameters
- Writing documentation
- Prep for tomorrow's lab

Mar 10, 2008

Sprengle - CS111

1

Friday's Exam

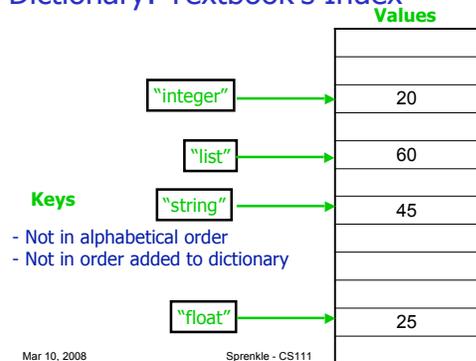
- Exam 2 Prep Document on line
 - Focus: strings, functions, files, lists, dictionaries
 - Know common, useful methods and operations for each data type
 - Slightly more on reading and explaining code
- No class on Wednesday
 - Work on lab
 - Study for exam
- Leaving Wed at noon

Mar 10, 2008

Sprengle - CS111

2

Dictionary: Textbook's Index



Mar 10, 2008

Sprengle - CS111

3

Creating Dictionaries in Python

Syntax:

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

```
empty = {}
```

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

Mar 10, 2008

Sprengle - CS111

4

Accessing Values Using Keys

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

```
if 'z' in ascii:  
    value = ascii['z']
```

Know mapping exists
before trying to access

- Or handle if get default back

```
val = ascii.get('z')  
if val is None:  
    # do something ...
```

Mar 10, 2008

Sprengle - CS111

5

Example Problem

- Modify the previous program to keep track of the number of students of each year

➢ Could we solve this using a list?

Mar 10, 2008

Sprengle - CS111

years_dictionary2.py 6

Analyzing years_dictionary2.py

- Anything useful/general that we could put in a function?

Mar 10, 2008

Sprengle - CS111

7

Defaults for Parameters

- Saw with the **xrange** function
 - Didn't have to specify *start* or *increment* when calling the function
 - Default *start* = 0
 - Default *increment* = 1
- Can assign a default value to a parameter
 - In general, in function header, default parameter(s) should come after all the parameters that *need* to be defined

Mar 10, 2008

Sprengle - CS111

8

Using Default Parameters

- By default, the rollDie function could assume that a die has 6 sides

```
def rollDie(sides=6):  
    return random.randint(1,sides)
```

Assigns a value to *sides*
ONLY IF not passed a parameter

Examples of calling the function:

```
rollDie(6)  
rollDie()  
rollDie(12)
```

Mar 10, 2008

Sprengle - CS111

game.py

9

Problem: Student Majors

- We want to keep track of the number of majors of each type
 - Twist: Not every student has a major (don't declare until sophomore year)

Mar 10, 2008

Sprengle - CS111

majors_dictionary.py 10

Problem: Student Majors, revised

- Students can have more than one major
 - Should count these *separately*
- How can we modify the previous program to do that?

Mar 10, 2008

Sprengle - CS111

11

Why Data File Problems Ad Nauseam?

- **"Parsing"** data files for different purposes if very common in science

Simplified web application access log:

```
128.4.131.54 [09/Aug/2005:14:01:35] GET /dspace/simple-search  
128.4.133.79 [09/Aug/2005:14:13:13] GET /dspace/simple-search  
128.4.133.139 [09/Aug/2005:14:28:20] GET /dspace/simple-search  
128.4.133.139 [09/Aug/2005:14:32:45] GET /dspace/adv-search  
...
```

I write scripts to

- create user sessions (use as test cases)
- analyze user sessions (avg. length, patterns)
- emulate user sessions

Mar 10, 2008

Sprengle - CS111

12

Lists vs. Dictionaries

Lists	Dictionaries
integer <i>positions</i> (0, ...) to any type of value	Map <i>immutable keys</i> (int, float, string) to any type of value
Ordered	Unordered
Slower to find a value (in)	Fast to find a value (use key)
Fast to print in order	Slower to print in order (by key)
Only as big as you make it	Takes up a lot of space (so can add elements in the middle)

Mar 10, 2008

Sprengle - CS111

13

Lists as Parameters to Functions

- If list (passed as a parameter into a function) is modified in a function, the list is modified *outside* the function
 - Lists are **not** passed-by-value
 - Different from immutable types (e.g., numbers, strings)
- Parameter is actually a **pointer** to the list in memory

Mar 10, 2008

Sprengle - CS111

14

Problem: Sort a list of 3 numbers, in descending order

- How with list methods?
- Can we do this using only **3** comparisons?

```
# order list such that list[0] >= list[1] >= list[2]
def descendSort3Nums( list3 ):
```

Called as:

```
list = ...
descendSort3Nums(list)
print list
```

Mar 10, 2008

Sprengle - CS111

[descendSort.py](#)

15

Descend Sort a List w/ 3 elements

```
def descendSort3Nums(list3):
    if list3[1] > list3[0]:
        # swap 'em
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp

    if list3[2] > list3[1]:
        tmp = list3[1]
        list3[1] = list3[2]
        list3[2] = tmp

    if list3[1] > list3[0]:
        tmp = list3[0]
        list3[0] = list3[1]
        list3[1] = tmp
```

```
def main():
    list = [1,2,3]
    descendSort3Nums(list)
    print list
```

Note: Function does not *return* anything. Simply modifies the list3 parameter.

Mar 10, 2008

Sprengle - CS111

16

Getting Documentation

- **dir**: function that returns a list of methods and attributes in an object
 - dir(<type>)
- **help**: get documentation
- In the Python shell
 - help(<type>)
 - import <modulename>
 - help(<modulename>)

Mar 10, 2008

Sprengle - CS111

17

Where is Documentation Coming From?

- Comes from the code itself in “**doc strings**”
 - i.e., “documentation strings”
- Doc strings are simply strings *after* the function header
 - Typically use triple-quoted strings because documentation goes across several lines

```
def verse(animal, sound):
    """ prints a verse of Old MacDonald, filling
    in the strings for animal and sound """
```

Mar 10, 2008

Sprengle - CS111

18

Deal or No Deal Overview

- Have 26 cases with various amounts of money
 - Amounts are known
- Player selects a case (hope has the big jackpot)
- In each round, player opens up cases
 - Reveals amounts that are not in the case they chose
- Banker makes an offer to buy the case
- Player decides if want to take the deal
 - Is the offer more than what is in the case?
 - Make decision based on amounts that haven't been opened yet
- Game ends when only one more case to open (two amounts on board) or player takes the deal.

Mar 10, 2008

Sprenkle - CS111

19