

Objectives

- Continuing with Files
- Defining Functions

Feb 27, 2012

Sprenkle - CSCI111

1

Midway Check: Parts of an Algorithm

- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

- Which of these have we covered?
- How do we implement them in Python?

Feb 27, 2012

Sprenkle - CSCI111

2

Midway Check: Parts of an Algorithm

- Primitive operations
 - What data you have, what you can do to the data
- Naming
 - Identify things we're using
- Sequence of operations
- Conditionals
 - Handle special cases
- Repetition/Loops
- Subroutines
 - Call, reuse similar techniques

where most of the rest of the semester focuses

No longer primitive

One more loop structure

Defining our own

Feb 27, 2012

Sprenkle - CSCI111

3

Review

- Why files?
- How do we create a new file object?
- What are two ways to read from a file?
- What should we do after we're done reading from a file?

Feb 27, 2012

Sprenkle - CSCI111

4

Review: Files

- Conceptually, a file is a **sequence** of data stored in memory
- To use a file in a Python script, create an object of type **file**
 - **file** is a *data type*
 - `<varname> = open(<filename>, <mode>)`
 - `<filename>`: string
 - `<mode>`: string, "r" for read, "w" for write, "a" for append (and others)
 - Ex: `dataFile = open("years.dat", "r")`

Built-in function
"constructs" a file object

Feb 27, 2012

Sprenkle - CSCI111

5

Review: Common File Methods

Method Name	Functionality
<code>read()</code>	Read the entire content from the file, returned as a string object
<code>readline()</code>	Read one line from file, returned as a string object (which includes the "\n"). If it returns "", then you've reached the end of the file
<code>write(string)</code>	Write a string to the file
<code>close()</code>	Close the file. Must close the file after done reading from/writing to a file

Feb 27, 2012

Sprenkle - CSCI111

6

Review: Reading from a File

- Recall that a file is a **sequence** of data
- Can use a **for** loop to iterate through a file

A line (of type **str**) from the file (includes **\n**)

file object

```
for line in dataFile:
    print(line)
```

➤ Read as: for each line in the file, do something

Feb 27, 2012

Sprenkle - CSCI1111

file_read2.py

7

Data Types of Loop Variables

What are the data types of the loop variable **x**?
What does **x** represent?

```
string = "some string"
dataFile = open("years.dat", "r")

for x in range(len(string)):
    # loop body ...

for x in string:
    # loop body ...

for x in dataFile:
    # loop body ...
```

Feb 27, 2012

Sprenkle - CSCI1111

8

Data Types of Loop Variables

What are the data types of the loop variable **x**?
What does **x** represent?

```
string = "some string"
dataFile = open("years.dat", "r")
```

```
for x in range(len(string)):
    # loop body ...
```

integer

```
for x in string:
    # loop body ...
```

string → single characters

```
for x in dataFile:
    # loop body ...
```

string → line (include **\n**)

Feb 27, 2012

Sprenkle - CSCI1111

9

Review: Processing a File

- file_search.py

Feb 27, 2012

Sprenkle - CSCI1111

10

Writing to a File

- Create a file object in **write** mode:
➤ `myFile = open("years.txt", "w")`
- Example: create a file from user input
➤ `file_write.py`

What happens if you execute the program again with different user input?

Feb 27, 2012

Sprenkle - CSCI1111

11

Handling Numeric Data

- We have been dealing with reading and writing **strings** so far
➤ Read from a file: get a string
➤ Write to file: use a string
- What do we need to do to **read numbers** from a file?
- How can we **write numbers** to a file?

Feb 27, 2012

Sprenkle - CSCI1111

12

Handling Numeric Data

- We have been dealing with reading and writing *strings* so far
 - Read from a file: get a string
 - Write to file: use a string
- What do we need to do to **read numbers** from a file?
 - Cast as a numeric type, e.g., `int` or `float`
- How can we **write numbers** to a file?
 - Cast number as a `str`

Feb 27, 2012

Sprenkle - CSCI111

13

Problem: Temperature Data

- **Given:** data file that contains the daily high temperatures for last year at one location
 - Data file contains one temperature per line
 - Example: `data/florida.dat`
- **Problem:** What is the average high temperature (to 2 decimal places) for the location?

Rule of Thumb: Always look at data file before processing it

Feb 27, 2012

Sprenkle - CSCI111

`avgData.py`

14

Problem: Create a Summary Report

- **Given:** a file containing students names and their years (first years, sophomore, junior, or senior) for this class
- **Problem:** create a report (in a file) that says the year and how many students from that year are in this class, on the same line.
 - Again, we want to ignore comments in the file

Do we need to start this program from scratch?
Have code we can use or repackage?

`writeSumReport.py`

Feb 27, 2012

Sprenkle - CSCI111

15

This Week

- Tuesday: Lab 6
 - Lists
 - Files
 - Functions
- I'm going to a conference after Wed's class
 - Plan accordingly
 - Email access
- No class Friday
 - **Still have Broader Issue write up**
 - A few different questions because no in-class discussion
 - See description on Sakai for more information

Feb 27, 2012

Sprenkle - CSCI111

16