


File I/O

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*

Built-in function
“constructs” a file object



- `<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")`

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

Reading from a File

- 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

file_read2.py

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?

Data Types of Loop Variables

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

```
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 ...
```

Data Types of Loop Variables

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

```
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)

File I/O

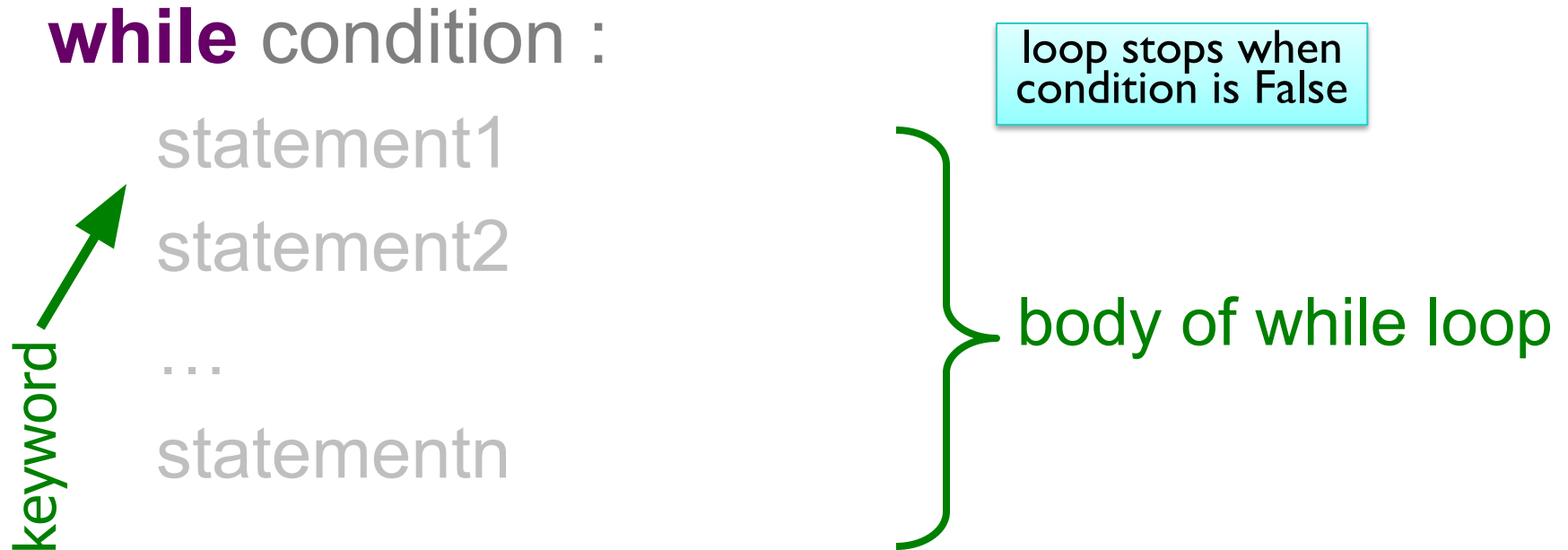
More details

Reading Files with a while loop

Review

- How can we make something repeat when some condition is true?
- True or False: Every **for** loop can be converted into a **while** loop
- True or False: A **while** loop is more powerful than a **for** loop

Review: While Loop Syntax



- Like a looped **if** statement
 - Execute statements **only** when condition is true

A Very Simple Therapist

- Whenever a user tells the computer/program what they think, the program asks, "How does that make you feel?"
- Ends when user enters nothing ("")
- Partial example output:

Tell me what is bothering you.

There is too much going on in my life.

How does that make you feel?

I feel like I am out of control and can't juggle it all.

How does that make you feel?

Really stressed and tired.

How does that make you feel?

Thank you! Come again!

Design Pattern: Sentinel Loop

- Sentinel: when to stop
 - “guard” to the loop

```
value = initialize
while value != sentinel :
    process value
    value = updated value
```

Another Way to Read from a File

```
FILENAME="data/years.dat"

dataFile = open(FILENAME, "r")

line = dataFile.readline()

while line != "":
    line = line.rstrip()
    print(line)
    line = dataFile.readline()

dataFile.close()
```

file_read.py

Writing files

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?

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?

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**

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

avgData.py

Problem

- Given a file of the form
 - <lastname> <year>
- Goal: I want to quickly find out what a student's class year is
 - How do we want to model the data?
 - What is the key? What is the value?
 - How to display the mapping in a pretty way?
 - What order is the data printed in?

years_dictionary.py

Problem

- Modify the previous program to keep track of the *number* of students of each year
 - How do we want to model the data?
 - What is the key? What is the value?
- Could we solve this using a list?

years_dictionary2.py

Analyzing years_dictionary2.py

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

Equivalent Solutions

```
if key not in dictionary :  
    dictionary[key] = 1  
else:  
    value = dictionary[key] + 1  
    dictionary[key] = value
```

```
if key not in dictionary :  
    dictionary[key] = 1  
else:  
    dictionary[key] += 1
```

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.

`writeSumReport.py`