COMP 141

Reading/Writing from/to Files

Rhodes College

1

Announcements

- Solutions to selected problems from Friday's lab are in the Box.com directory (LoopLab.py)
- Reminder:
  - Program 5 due Thursday, March 26th

2

Writing to a File

- For a program to retain data between the times it is run, you must save the data
  - Data is saved to a file, typically on computer disk
  - Saved data can be retrieved and used at a later time
- "Writing data to": saving data on a file
- Output file: a file that data is written to

3

Reading From a File

- "Reading data from": process of retrieving data from a file
- Input file: a file from which data is read
- Three steps when a program uses a file
  - Open the file
  - Process the file
  - Close the file

4

3/19/2020

Types of Files and File Access Methods

- In general, two types of files
  - <u>Text file</u>: contains data that has been encoded as text
  - <u>Binary file</u>: contains data that has not been converted to text
- Two ways to access data stored in file
  - <u>Sequential access</u>: file read sequentially from beginning to end, can't skip ahead
  - <u>Direct access</u>: can jump directly to any piece of data in the file

5

5

Filenames and File Objects

- <u>Filename extensions</u>: short sequences of characters that appear at the end of a filename preceded by a period
  - Extension indicates type of data stored in the file
- <u>File object</u>: object associated with a specific file
  - Provides a way for a program to work with the file: file object referenced by a variable

6

6

Opening a File

- <u>open function</u>: used to open a file
  - Creates a file object and associates it with a file on the disk
  - General format:
    `file_object = open(filename, mode)`
- <u>Mode</u>: string specifying how the file will be opened
  - Example: reading only ('r'), writing ('w'), and appending ('a')

7

7

Specifying the Location of a File

- If `open` function receives a filename that does not contain a path, assumes that file is in same directory as program
- If program is running and file is created, it is created in the same directory as the program
  - Can specify alternative path and file name in the `open` function argument
    - Prefix the path string literal with the letter `r`

8

8

2

Writing Data to a File

- <u>Method</u>: a function that belongs to an object
  - Performs operations using that object
- File object's `write` method used to write data to the file
  - Format: *file_variable*.write(*string*)
- File should be closed using file object `close` method
 - Format: *file_variable*.close()

9

Writing Data to a File

Program 7-1   (file_write.py)

```
1   # This program writes three lines of data
2   # to a file.
3   def main():
4       # Open a file named philosophers.txt.
5       outfile = open('philosophers.txt', 'w')
6
7       # Write the names of three philosphers
8       # to the file.
9       outfile.write('John Locke\n')
10      outfile.write('David Hume\n')
11      outfile.write('Edmund Burke\n')
12
13      # Close the file.
14      outfile.close()
15
16  # Call the main function.
17  main()
```

```
philosophers.txt - ...
File  Edit  Format  View  Help
John Locke
David Hume
Edmund Burke
```
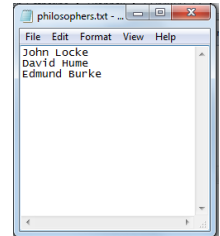
10

Reading Data From a File

- <u>read</u> method: file object method that reads entire file contents into memory
  - Only works if file has been opened for reading
  - Contents returned as a string
- <u>readline</u> method: file object method that reads a line from the file
  - Line returned as a string, including '\n'
- <u>Read position</u>: marks the location of the next item to be read from a file

11

Reading Data From a File

Program 7-2   (file_read.py)

```
1   # This program reads and displays the contents
2   # of the philosophers.txt file.
3   def main():
4       # Open a file named philosophers.txt.
5       infile = open('philosophers.txt', 'r')
6
7       # Read the file's contents.
8       file_contents = infile.read()
9
10      # Close the file.
11      infile.close()
12
13      # Print the data that was read into
14      # memory.
15      print(file_contents)
16
17  # Call the main function.
18  main()
```

```
philosophers.txt - ...
File  Edit  Format  View  Help
John Locke
David Hume
Edmund Burke
```
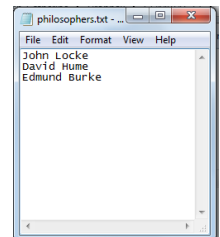
Program Output
```
John Locke
David Hume
Edmund Burke
```

12

9

10

11

12

3

Reading Data From a File

```
# This program reads the contents of the
# philosophers.txt file one line at a time.
def main():
    # Open a file named philosophers.txt.
    infile = open('philosophers.txt', 'r')

    # Read three lines from the file
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()

    # Close the file.
    infile.close()

    # Print the data that was read into
    # memory.
    print(line1)
    print(line2)
    print(line3)

# Call the main function.
main()
```

Program Output

John Locke

David Hume

Edmund Burke

13

Concatenating a Newline to and Stripping it From a String

- In most cases, data items written to a file are values referenced by variables
  - Usually necessary to concatenate a '\n' to data before writing it
    - Carried out using the + operator in the argument of the `write` method
- In many cases need to remove '\n' from string after it is read from a file
  - `rstrip` method: string method that strips specific characters from end of the string

14

Stripping Newline from a String

```
# This program reads the contents of the
# philosophers.txt file one line at a time.
def main():
    # Open a file named philosophers.txt.
    infile = open('philosophers.txt', 'r')

    # Read three lines from the file
    line1 = infile.readline()
    line2 = infile.readline()
    line3 = infile.readline()

    # Strip the \n from each string.
    line1 = line1.rstrip('\n')
    line2 = line2.rstrip('\n')
    line3 = line3.rstrip('\n')

    # Close the file.
    infile.close()

    # Print the data that was read into
    # memory.
    print(line1)
    print(line2)
    print(line3)

# Call the main function.
main()
```

Program Output

John Locke
David Hume
Edmund Burke

15

Writing Data to a File

Program 7-1   (file_write.py)

```
1   # This program writes three lines of data
2   # to a file.
3   def main():
4       # Open a file named philosophers.txt.
5       outfile = open('philosophers.txt', 'w')
6
7       # Write the names of three philosphers
8       # to the file.
9       outfile.write('John Locke\n')
10      outfile.write('David Hume\n')
11      outfile.write('Edmund Burke\n')
12
13      # Close the file.
14      outfile.close()
15
16  # Call the main function.
17  main()
```

16

13

14

15

16

Appending Data to an Existing File

- When open file with 'w' mode, if the file already exists it is overwritten
- To append data to a file use the 'a' mode
  - If file exists, it is not erased, and if it does not exist it is created
  - Data is written to the file at the end of the current contents

17

17

Writing and Reading Numeric Data

- Numbers must be converted to strings before they are written to a file
- <u>str</u> function: converts value to string
- Number are read from a text file as strings
  - Must be converted to numeric type in order to perform mathematical operations
  - Use int and float functions to convert string to numeric value

18

18

Reading Numbers from a File

```python
# This program demonstrates how numbers that are
# read from a file must be converted from strings
# before they are used in a math operation.

def main():
    # Open a file for reading.
    infile = open('numbers.txt', 'r')

    # Read three numbers from the file.
    num1 = int(infile.readline())
    num2 = int(infile.readline())
    num3 = int(infile.readline())

    # Close the file.
    infile.close()

    # Add the three numbers.
    total = num1 + num2 + num3

    # Display the numbers and their total.
    print('The numbers are:', num1, num2, num3)
    print('Their total is:', total)

# Call the main function.
main()
```

19

19

Writing Numbers to a File

```python
# This program demonstrates how numbers
# must be converted to strings before they
# are written to a text file.

def main():
    # Open a file for writing.
    outfile = open('numbers.txt', 'w')

    # Get three numbers from the user.
    num1 = int(input('Enter a number: '))
    num2 = int(input('Enter another number: '))
    num3 = int(input('Enter another number: '))

    # Write the numbers to the file.
    outfile.write(str(num1) + '\n')
    outfile.write(str(num2) + '\n')
    outfile.write(str(num3) + '\n')

    # Close the file.
    outfile.close()
    print('Data written to numbers.txt')

# Call the main function.
main()
```

20

20

5

Your practice

- Write a program that writes a series of random numbers to a file. Each random number should be in the range of 1 through 100. Write at least 5 random numbers to the file – 1 number/line.

- Call your output file `randomNums.txt`

21

21