

Filing Things Away

Self-Review Questions

Self-review 8.1 We can store the contents of dictionaries, lists and tuples in a file and, using `eval`, retrieve them while keeping their type and structure. Does this mean that the operating system specifically supports these types?

Self-review 8.2 Why should we not use the write mode ('w') when we wish to add data to an existing file?

Self-review 8.3 What mode *should* be used when adding data to an existing file?

Self-review 8.4 If you search your computer's hard disk, you may find that many files have the same name. `Readme.txt`, for example, appears many times on our hard disks. These files are distinct and normally have different contents. How is it that the operating system can maintain separate files of the same name?

Self-review 8.5 What is 'wrong' with the histogram drawing algorithm on page ???

Hint: The problem is not in the code structure but in the assumption about coordinates.

The problem here is that we had assumed that the bottom left corner was the origin, (0, 0), and that positive x was rightward and positive y was upward. In fact the origin is the top left-hand corner with positive x rightward and positive y downward. So correcting the histogram drawing for a more expected appearance, we have to make two relatively trivial changes:

```
canvas.create_rectangle (
    (2 * i + 1) * barWidth , height - unitHeight ,
    (2 * i + 2) * barWidth , height - ( data[i][1] + 1 ) * unitHeight ,
    fill = 'black' )
```

Self-review 8.6 What are the special files `.` and `..` for?

Self-review 8.7 What is the 'root' directory in a file system and what is its `..` file?

Self-review 8.8 In the `os` module Python has a variable called `os.sep` which has the value `'/'` on UNIX machines and `'\'` on Windows computers. What do you think this variable is used for?

Self-review 8.9 What are *buffers* and why does Python use them when opening files?

Self-review 8.10 What does the flush method do?

Self-review 8.11 What does the special method `__repr__` do and why is it different to the `__str__` method?

Self-review 8.12 What does Python's eval function do?

Self-review 8.13 What exception is raised when Python tries to open a file which does not exist?

Self-review 8.14 The check digit in ISBNs tells us whether an ISBN is valid. Can you think of any other computing related activities where check digits (or check sums) are used?

Self-review 8.15 Why do some ISBN numbers end with an X?

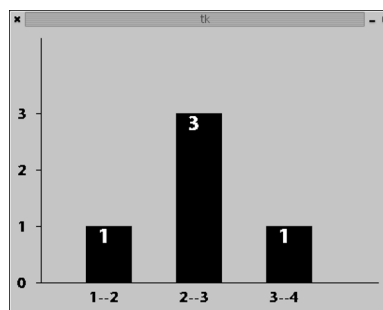
Programming Exercises

Exercise 8.1 Amend the `getData` method so that the parameter is a file object rather than a file name. Change the histogram drawing program removing the closing of the file and execute it to show that the amended `getData` method works as it should.

Exercise 8.2 Amend the `getData` method so that the parameter can be either a string giving the name of the file to be read, or a file object of an already open file.

Hint: Use the `isinstance` function to separate the three cases.

Exercise 8.3 Evolve the histogram drawing code (your corrected version from Self-review 8.5 rather than the faulty version on page ??) so that the bars have labels that show which bin the data is for, and so that there are indications of what the lengths of the bars represent. One approach would be to draw an axis down the side with tick marks and values. It may also be sensible to write the actual count of items in the bin above the bar, perhaps resulting in something like:



```
import Tkinter
```

```
def plot ( data ) :
```

```
    numberOfBuckets = len ( data )
    maximumValue = max ( [ datum[1] for datum in data ] )
    root = Tkinter.Tk ( )
    width , height = 400 , 300
    canvas = Tkinter.Canvas ( root , width = width , height = height )
    canvas.pack ( )
    originX , originY = 30 , height - 30
    canvas.create_line ( originX , originY , width - 10 , originY )
    canvas.create_line ( originX , originY , originX , 10 )
    barWidth = ( width - 2 * originX ) / ( 2 * numberOfBuckets + 1 )
    unitHeight = 300 / ( maximumValue + 2 )
    fontName = 'stonesans' # Use X names.
    axesFontDescriptor = ( fontName , 12 , 'bold' )
    numberOfTicks = 6
    for i in range ( numberOfTicks + 1 ) :
        value = i * maximumValue / numberOfTicks
        y = originY - value * unitHeight
        canvas.create_line ( originX - 5 , y , originX , y )
        canvas.create_text ( originX - 20 , y , text = str ( value ) , font = axesFontDescriptor )

    for i in range ( numberOfBuckets ) :
        x0 = ( 2 * i + 1 ) * barWidth + originX
        x1 = x0 + barWidth
        y0 = height - 30
        y1 = y0 - data[i][1] * unitHeight
        canvas.create_rectangle ( x0 , y0 , x1 , y1 , fill = 'black' )
        canvas.create_text ( x0 + 0.4 * barWidth , y0 + 15 , text = data[i][0] , font = axesFontDescriptor )
        canvas.create_text ( x0 + 0.4 * barWidth , y1 + 10 , text = data[i][1] , fill = 'white' , font = ( fontName , 12 ) )
    root.mainloop ( )
```

```
if __name__ == '__main__' :
```

```
    plot ( [
        [ '1--2' , 1 ] ,
        [ '2--3' , 3 ] ,
        [ '3--4' , 1 ]
    ] )
```

Exercise 8.4 Look at the following program:

```
file = open ( 'words.dat' , 'w' )
word = ""
while word != 'END' :
    word = raw_input ( 'Enter a word (enter END to quit): ' )
    file.write ( word + '\n' )
file.close ( )
```

This is a very simple program for storing a list of words in a file. When executed it expects the user to enter some words, one per line, finishing with the word "END".

1. What is the name of the file containing the words?
2. How many words are stored in the file?

3. Write a program to read the data file and display a numbered list of the contents, such as:
 - 1: chicken
 - 2: apple
 - 3: fox
 - 4: aubergine
 - 5: END
4. Note that “END” is always the last word displayed. This isn’t really the behavior we want, since “END” is meant as a command, not a word. Change the program so that it no longer stores “END” in the file.
5. There’s still a problem: we can’t put the word “END” into the file even if we wanted to. Maybe we should use the empty string (“”) to signal the user’s desire to quit? Modify the program to use the empty string to terminate user input.

Exercise 8.5 Using your answer to the previous question as a starting point, write a program that counts the number of times each word appears in the data file. Its output might look like this:

```
Carrot: 3
Cheese: 1
Egg: 6
Tomato: 2
```

Exercise 8.6 It would be nice if we had a graphical display of word frequency for the previous exercise. Use the histogram module from earlier in the chapter to display the word frequencies as a histogram.

Exercise 8.7 Write a program to say how many lines of text there are in a text file the name of which is obtained by prompting and reading input from the user.

Exercise 8.8 Write a program to read the contents of a text file and display it in a Tkinter text box.

Exercise 8.9 Write a program to store a list of contact names and telephone numbers, similar to the contact lists you might find on a mobile phone. The data needs to be stored in a file so that it is *persistent* – that is, the data available at the beginning of a new execution of the program is the same as at the end of the previous execution.

Exercise 8.10 Extend the program written for the previous exercise so that as new contacts are added, the program checks for duplicate names and numbers. If the new contact’s name is already in the database, the message “This person is already entered.” should appear. If the telephone number is already in the database, the program should display “This person’s telephone number is already in the database. This could be an error, or a second contact at the same company. Add anyway?” If the user enters “Y”, the contact is added.

Challenges

Challenge 8.1 Write a program to display a simple form of digital book. “Books” are text files in which each block (page) of text is followed by a double dash (--).

When a book is displayed, the first block of text is shown and the program should wait for the user to press the enter key before displaying the next.

Challenge 8.2 Extend the previous challenge so that it is possible to skip forward by an arbitrary number of pages. This should be achieved by allowing the user to enter a number before pressing the enter key. If the number is positive, the given number of pages are skipped. If there is no number, the next page is displayed.

Challenge 8.3 Further extend the book reader so that it can accept negative numbers for skipping pages. Entering -1 should go back to the previous page. There are many ways to achieve this.