Diary

Difficulty: Intermediate

This exercise shows how to use files to store data that the computer remembers after the program has finished.

It builds up a program to keep a diary that keeps its data in files, so it is not lost when the program terminates.

# Introduction

In computer programs it’s generally a good idea to keep *data* separate from *code*. This is because the code in your program doesn’t change (unless you’re in the middle of writing it!), but data usually changes all the time. For example, in a word processor you’re always writing new documents, but the word processor itself isn’t changing with each document you write.

Computer games follow the same pattern: graphic artists create the textures and images for a game and store them in files that are then loaded by the code (the game engine) to display on the screen.

So it’s important to understand how to read and write files. Fortunately, in Python, it’s really easy!

## Reading and Writing to Files

When programming, it can be helpful to think of files as being like cardboard folders. If you want to get at the paper inside you *open* the file first, then *read* or *write* data, thenwhen you’ve finished, you *close* the file.

The simple program below shows how to read and write to a file. Type it in and experiment running it by first writing to a file, then reading it again.

def readDiary():

day = input("What day do you want to read? ")

file = open(day, "r")

line = file.read()

print(line)

file.close()

def writeDiary():

day = input("What day is your diary for? ")

file = open(day, "w")

line = input("Enter entry: ")

file.write(line)

file.close()

operation = input("Read entries or write entries (R/W)? ")

if (operation == "R"):

readDiary()

elif (operation == "W"):

writeDiary()

else:

print("Sorry, enter a R (for read) or W (for write) only.")

print("=== All done ===")

Before you move on, read the code and make sure you understand what each line does. What do you think the “r” and “w” parameters to the open function do?

Add some comments to your file (use the # character to tell Python that everything after it on the same line is a comment.

Note that in programming, it’s very important to close a file after it’s been opened. This is because the computer can only open a certain number of files at once. If you forget to close files, strange things might happen later!

## Multiple Lines in Each Diary Entry

In the program so far, we can only read or write a single line of text in a diary entry. Changing the program to add extra lines requires us to update both the readDiary and writeDiary functions.

Making readDiary read all the lines in a file only requires the following changes:

def readDiary():

day = input("What day do you want to read? ")

file = open(day, "r")

line = file.read()

while line!="":

print(line)

line = file.read()

file.close()

Examine the code above and update your readDiary function so it matches this one. Look at the lines that have changed. Make sure that the new code works with diary entries that you’ve created so far.

How does readDiary know that it’s read all the lines?

Now we need to do something similar in writeDiary, to allow the user to keep entering lines. The code will keep reading lines from the user until they enter a blank line.

def writeDiary():

day = input("What day is your diary for? ")

file = open(day, "w")

while True:

line = input("Enter entry (enter a blank line to finish): ")

file.write(line)

if line=="":

break

file.close()

Again, read through the code and make sure you understand what’s going on.

Try out your code. What happens when you read back an entry with multiple lines? Can you fix the problem by making a change to one line in writeDiary?