

Exercises chapter 10: Files and exceptions

Saul SL

June 2023

1 Exercise 10-1 Learning Python

Open a blank file in your text editor and write a few lines summarizing what you've learned about Python so far. Start each line with the phrase In Python you can. . . . Save the file as `learning_python.txt` in the same directory as your exercises from this chapter. Write a program that reads the file and prints what you wrote three times. Print the contents once by reading in the entire file, once by looping over the file object, and once by storing the lines in a list and then working with them outside the with block.

```
1  import datetime
2  import sys
3  import os
4  os.chdir('/home/saul/Documents/Apuntes/Python/Matthes-2019/Ch-10')
5  ifile = 'learning_python.txt'
6
7  # Read the entire file
8  with open(ifile) as fh:
9      whole_txt = fh.read()
10     print(whole_txt)
11
12     # size of the object
13     isize = sys.getsizeof(whole_txt)
14     print(isize, "bytes")
15
16     # Read line by line
17     with open(ifile) as fh:
18         for line in fh:
19             print(line.strip())
20
21     # Store the lines in an object
22     with open(ifile) as fh:
23         lines = fh.readlines()
24
25     for line in lines:
26         print(line.strip())
27
28     isize2 = sys.getsizeof(lines)
29     print(isize2, "bytes")
```

2 Exercise 10-2 Learning C

You can use the `replace()` method to replace any word in a string with a different word. Here's a quick example showing how to replace 'dog' with 'cat' in a sentence:

```
>>> message = "I really like dogs."
>>> message.replace('dog', 'cat')
'I really like cats.'
```

Read in each line from the file you just created, `learning_python.txt`, and replace the word Python with the name of another language, such as C. Print each modified line to the screen.

```

1 with open(ifile) as fh:
2     lines2 = fh.readlines()
3
4 for line in lines2:
5     print(line.strip().replace('Python', 'R'))

```

3 Exercise 10-3 Guest

Write a program that prompts the user for their name. When they respond, write their name to a file called `guest.txt`.

```

1 u_name = input("Please write your name: ")
2 out_file = 'guests.txt'
3 with open(out_file, 'w') as fh:
4     fh.write(u_name)

```

4 Exercise 10-4 Guest Book

Write a while loop that prompts users for their name. When they enter their name, print a greeting to the screen and add a line recording their visit in a file called `guest_book.txt`. Make sure each entry appears on a new line in the file.

```

1 out_file = 'guests_book.txt'
2 with open(out_file, 'w') as fh:
3     fh.write("List of guests\n")
4
5 print("Please write your name.\nType 'q' to exit")
6 while True:
7     iname = input("Name: ")
8     if iname == 'q':
9         break
10    else:
11        with open(out_file, 'a') as fh:
12            now_time = datetime.datetime.now()
13            itime = now_time.strftime("%Y-%m-%d_%H:%M:%S")
14            fh.write(f"- {itime}:\t {iname} \n")

```

5 Exercise 10-5 Programming Poll

Write a while loop that asks people why they like programming. Each time someone enters a reason, add their reason to a file that stores all the responses.

```

1 out_file = 'programming_poll.txt'
2 with open(out_file, 'w') as fh:
3     fh.write("Why do you like programming?\n")
4
5 print("Why do you like programming?\n")
6 print("Please write your answer.\nType 'q' to exit")
7 while True:
8     ianswer = input("Answer: ")
9     if ianswer == 'q':
10        break
11    else:
12        with open(out_file, 'a') as fh:
13            now_time = datetime.datetime.now()

```

```

14         itime = now_time.strftime("%y-%m-%d_%H:%M:%S")
15         fh.write(f"- {itime}:\t {ianswer} \n")

```

6 Exercise 10-7 Addition Calculator

Wrap your code from Exercise 10-6 in a while loop so the user can continue entering numbers even if they make a mistake and enter text instead of a number.

```

1  print("Insert two numbers. Their sum will be calculated")
2  print("Type 'q' to exit")
3
4  while True:
5      inum1 = input("1st Number: ")
6      inum2 = input("2nd Number: ")
7      if inum1 == 'q' or inum2 == 'q':
8          break
9
10     try:
11         inum1 = int(inum1)
12         inum2 = int(inum2)
13
14     except ValueError:
15         print('The program needs a numeric input')
16         print('Please try again')
17
18     else:
19         isum = inum1 + inum2
20         print(f"{inum1} + {inum2} = {isum}")

```

7 Exercise 10-8 Cats and Dogs

Make two files, cats.txt and dogs.txt. Store at least three names of cats in the first file and three names of dogs in the second file. Write a program that tries to read these files and print the contents of the file to the screen. Wrap your code in a try-except block to catch the FileNotFoundError error, and print a friendly message if a file is missing. Move one of the files to a different location on your system, and make sure the code in the except block executes properly.

```

1  in_files = ['Cats.txt', 'dogs.txt']
2  for ifile in in_files:
3      try:
4          with open(ifile) as fh:
5              lines = fh.readlines()
6
7      except FileNotFoundError:
8          print(f"File '{ifile}' not found")
9
10     else:
11         print(f"'{ifile}' contents:")
12         for line in lines:
13             print(line.strip())
14         print("")
15
16 # # 10-9
17 in_files = ['Cats.txt', 'dogs.txt']
18 for ifile in in_files:
19     try:
20         with open(ifile) as fh:
21             lines = fh.readlines()

```

```

22
23     except FileNotFoundError:
24         pass
25
26     else:
27         print(f"{ifile} contents:")
28         for line in lines:
29             print(line.strip())
30         print("")

```

8 Exercise 10-10 Common Words

Visit Project Gutenberg (<https://gutenberg.org/>) and find a few texts you'd like to analyze. Download the text files for these works, or copy the raw text from your browser into a text file on your computer. You can use the `count()` method to find out how many times a word or phrase appears in a string. For example, the following code counts the number of times 'row' appears in a string:

```

>>> line = "Row, row, row your boat"
>>> line.count('row')
2
>>> line.lower().count('row')
3

```

Notice that converting the string to lowercase using `lower()` catches all appearances of the word you're looking for, regardless of how it's formatted. Write a program that reads the files you found at Project Gutenberg and determines how many times the word 'the' appears in each text. This will be an approximation because it will also count words such as 'then' and 'there'. Try counting 'the ', with a space in the string, and see how much lower your count is.

```

1 ebooks = ['War_of_the_worlds.txt', 'Free_as_Freedom.txt']
2 word = 'the '
3 for ebook in ebooks:
4     try:
5         with open(ebook) as eb:
6             content = eb.read()
7     except FileNotFoundError:
8         pass
9     else:
10        print(f"Book: {ebook}")
11        occurrences = content.lower().count(word)
12        print(f"The word, '{word}' appears {occurrences} times")

```

9 Exercise 10-11 Favorite Number

Write a program that prompts for the user's favorite number. Use `json.dump()` to store this number in a file. Write a separate program that reads in this value and prints the message, "I know your favorite number! It's _____."

```

1 import json
2
3 # Write part
4 print("Write your favorite number.\nType 'q' to exit")
5 fav_num = input("Number: ")
6 out_file = 'Favorite_number.json'
7 with open(out_file, 'w') as fh:
8     json.dump(fav_num, fh)
9     print('Input recorded')
10

```

```

11 # read part
12 try:
13     with open(out_file) as fh:
14         content = json.load(fh)
15 except FileNotFoundError:
16     pass
17 else:
18     print(f"Your favorite number is: {content}")

```

10 Exercise 10-12 Favorite Number Remembered

Combine the two programs from Exercise 10-11 into one file. If the number is already stored, report the favorite number to the user. If not, prompt for the user's favorite number and store it in a file. Run the program twice to see that it works.

```

1 import json
2
3 def get_stored_username(user):
4     """Get stored username if available."""
5     filename = f'{user}.json'
6     try:
7         with open(filename) as f:
8             username = json.load(f)
9     except FileNotFoundError:
10         return None
11     else:
12         return username
13
14 def get_new_username(user):
15     # """Prompt for a new username."""
16     # username = input("What is your name? ")
17     username = user
18     filename = f'{username}.json'
19     with open(filename, 'w') as f:
20         json.dump(username, f)
21     return username
22
23 def greet_user(user):
24     """Greet the user by name."""
25     username = get_stored_username(user)
26     if username:
27         print(f>Welcome back, {username}!")
28     else:
29         username = get_new_username(user)
30         print(
31             f>We'll remember you when you come back, {username}!"
32         )
33
34 greet_user('tim')

```