PHOT 110: Introduction to programming

Lecture 12: exercises on IO and error handling

Michaël Barbier, Spring semester (2023-2024)

Exercises on file Input/Output

We will exercise reading data from and writing data to files of different formats.

There are several possible cases, for example, we might want:

- to read from a file
- to write to a file and either:
    - first erase the file if it already exists,
    - create the file if it doesn't exist yet,
    - or append the data to the already existing file

This is especially applicable to text files where we can imagine that sometimes we want just to append some text, but at other times we want to erase (and start over) the old file before writing new text.

With the following example code we open a file as a text-file and for only reading. Then we load the text as a string into variable `my_text`, and then close the file.

```python
# Open the file for only reading (file must exist)
f = open("sample_text_file.txt", "r", encoding="utf-8")
# Read the text-data
my_text = f.read()
# Close the file afterwards, otherwise data might not be saved or corrupt
f.close()
```

We can choose in which **mode** (read/write/append/…) we open a file by adding a mode argument:

- "r" stands for only reading,
- "w" stands for only writing,

- “a” stands for appending to the end,
- “r+” stands for creating the file if it doesn’t exist and both reading and writing.

Because it is very important to close the file afterwards, it is advised to open a file using the `with` keyword, which automatically closes the file after executing the code inside the `with` statement:

```
# Open the file for reading and writing
with open("sample_text_file.txt", "r+", encoding="utf-8") as f:
  # Read the text-data
  my_text = f.read()
  # Adapt the text
  my_text = my_text.replace("banana", "orange")
  # Append the adapted text to the old file content
  f.write("This and the next part will be appended to the original:\n")
  f.write(my_text)
```

Exercise 1: Reading a text file

Read a text file on your computer and print it. Use the “r” mode argument to read. You can use the `sample_text_file.txt` provided previously (or any other text file).

Exercise 2: Create and write to a new text file

Write the text: “Hello, this is some text.” to a new text-file. Use the mode argument “w+” to create a new file and open it for writing.

Exercise 3: Appending a sentence to a text file

Use the “a+” mode argument to append the line: “this is even more text.” to the previously generated text-file of exercise 2.

Exercises on error handling

Exercise 4: User console input

When we ask a user to fill in a form or type something at the prompt, we assume a certain format. Let’s take the example where we ask how tall a person is. Hereby we expect to get the answer as an integer number in centimeters. However, when prompted with this question, a user can fill in something invalid (e.g. “1m72”). When that happens we want to allow the user to try again and maybe give an extra hint.

A typical manner to arrange this is:

```
while True:
  try:
    height = int(input("\tYour height (in cm) = "))
    print(f"You are {height} cm")
  except ValueError:
    print(f"Invalid input: please try again")
```

- Adapt the above code so that it accepts decimal input such as “174.7”

Exercise 5: Read from a text file

- Adapt the code to read a text file on your computer of exercise one so that it doesn’t crash when the file path doesn’t exist.
- Prompt the user to choose the file name.