Error Handling, File Input & Output

Ceren Kocaoğullar
ckocaogullar15@ku.edu.tr
Hasan Can Aslan
haslan16@ku.edu.tr

Date: 11th of March

1 Overview

This handout is prepared for KOLT Python Certificate Program. It contains a brief review of this week's topics and exercise questions.
You can download the starter code from here.
You can find the solutions at here after all the sections are conducted.

2 Review

2.1 Data Structures

2.1.1 Tuples

- **Immutable** sequence of elements.

  Cannot make changes on a tuple using functions like add(), append(), remove() etc.

- Elements are **ordered.**

  Therefore you **can** use indexing, slicing.

  You can iterate over them using for loops.

- To create an empty one: `tuple()` or just `()`.

2.1.2 Sets

- **Unordered** sequence of elements.
- Elements are **unique**.

  Therefore you **cannot** use indexing/slicing.

  But you **can** iterate over them with for loops.

- Mutable, you can use add(element), remove(element) methods.
- You can use **union, intersection, difference, symmetric difference** operations on them.
- To create an empty one: `set()`.

2.1.3 Dictionaries

- Collection of **key−value** pairs.
- Keys of a dictionary are **unique**.
- In general, they are not **ordered**.

  However, in Python 3.7 pairs are guaranteed to be in insertion order.

  This means we will get pairs in insertion order if we loop over one.

  However, you **cannot** use **indexing/slicing**.

  But you **can** iterate over them with `for` loops.
- To create an empty one: {} or `dict()`: empty dictionary
- To access values: `print(d['key'])` # ⇒ 1
- To add/update a key-value pair: `print(d['key'] = value)` # ⇒ 1

2.2 Error/Exception Handling

2.2.1 Syntax Errors

Errors that you get when you have a **syntactically incorrect** piece of code. i.e. Code that doesn't follow the rules of coding in Phython.

```
for i in range(100)
print(i)
# SyntaxError: invalid syntax

while True:
print('Hello')
# IndentationError: expected an indented block
```

2.2.2 Runtime Errors

When a statement is **syntactically correct**, this doesn't mean we are safe. Python interpreter will run the code in that case, but can still raise errors.

```
print(3 / 0)
# ZeroDivisionError: division by zero

int('hello')
# ValueError: invalid literal for int() with base 10: 'hello'

'hello'[2] = 'a'
# TypeError: 'str' object does not support item assignment
```

2.2.3 Try Except Blocks

To be safe from these errors, we could put `if` checks everywhere. But it would be too much effort, and probably we cannot list every condition. The solution is `try-except-finally` blocks.

```
try:
    <risky-statement>
    <risky-statement>
    <risky-statement>
    ...
except ValueError as valError:
    print('value error', valError)
except (RuntimeError, TypeError, NameError):
    print('One of the above errors, but not ValueError')
else:
    print('Here, do what you want to do if there are no errors.
')
finally:
    print('This part runs no matter what.')
```

It is true that Python throws these errors anyway. However, if you handle them like this:

- You hold the power over deciding what to do if an error occurs.
- The program doesn't stop when it raises an exception. It just does what you wrote under the relevant except block and keeps running.

Pro Tip: You don't have to memorize the names of a whole list of errors! You can imagine what could go wrong, do the wrong thing and what the name of the error is from the Python itself!

2.3 File Input/Output

Access to a `file object` using `open(filename, mode='r')` function

- `filename`: File name including the **file extension**. Ex: 'data.txt'

  If you want to access/create a file outside of current **working directory**, you also need to include its path. Ex: './FolderName/data.txt', 'C:/Users/AUYSAL16/Desktop/data.txt'

- `mode` denotes how the file will be used. It is optional to declare mode, it has a default value of `w`:
  - 'r': read mode, default
  - 'w': write mode, overrides the file contents if it already exists
  - 'x': create & write mode, similar to write mode gives error if file already exists
  - 'a': append mode, adds content to the end of file

2.3.1 Reading file content

- Open the file with read mode (which is already the default mode).

  Ex: `f = open('my_file.txt')`

- `f.read()`: returns content of entire file as a string
- `f.readline()`: returns a single line from file
- `for line in f:` ⇒ Iterate over all lines

- `list(f)` or `f.readlines()`: read file lines to a list
- **Always** close the file when you are done: `f.close()`

2.3.2 Creating/modifying files

- Open the file with a write enabled mode, e.g, `w, x, a`

  Ex: `f = open('my_file','w')`
- Use `f.write(string)` to write to file.

  Warning! `file.write()` method **only** takes `str` values!
- Close the file when you are done.
- `f.close()`

3 Exercises

3.1 Registrar's Office

You are a work and study student for the Registrar's Office, help manage Kusis.

1. Oops, our database is not updated since last Fall, So correct the instructor for Engr 200 by changing it to Lerzan Ormeci.
2. Python Section 2 should contain all students in the other four courses. However, it is missing 2 students, find and add them to the class set.
3. Chem 103 and Econ 201 students will attend an interdisciplinary seminar, find the number of students who will be attending this seminar.
4. Math 205 class needs to be rescheduled, determine which time slot it can be placed without a time conflict.

```
Output:

Chem 103  Sarp Kaya  {'Gamze', 'Ata', 'Ayse', 'Mahsa', 'Furkan'}
Engr 200  Lerzan Ormeci  {'Ahmet', 'Canan', 'Furkan', 'Gonca'}
Econ 201  Seda Ertac   {'Gokce', 'Emirhan', 'Ayse', 'Abdullah', '
    Meva'}
Math 205  Nadim Rustom   {'Ahmet', 'Zeynep', 'Mahsa', 'Ilayda'}
Python Section 2  Hasan Can Aslan  {'Gokce', 'Gamze', 'Emirhan', '
    Ahmet', 'Meva', 'Ayse', 'Abdullah', 'Ilayda', 'Mahsa', 'Canan',
     'Furkan', 'Gonca'}

{'Ata', 'Zeynep'}

{'Chem 103': ('Sarp Kaya', {'Gamze', 'Ata', 'Ayse', 'Mahsa', '
    Furkan'}), 'Engr 200': ('Lerzan Ormeci', {'Ahmet', 'Canan', '
    Furkan', 'Gonca'}), 'Econ 201': ('Seda Ertac', {'Gokce', '
    Emirhan', 'Ayse', 'Abdullah', 'Meva'}), 'Math 205': ('Nadim
    Rustom', {'Ahmet', 'Zeynep', 'Mahsa', 'Ilayda'}), 'Python
    Section 2': ('Hasan Can Aslan', {'Gokce', 'Emirhan', 'Ata', '
    Ayse', 'Abdullah', 'Mahsa', 'Canan', 'Furkan', 'Gamze', 'Ahmet
    ', 'Ilayda', 'Meva', 'Gonca', 'Zeynep'})}

9

['Econ 201']
```

3.2 World Happiness Report

You are given the dataset of World Happiness Report 2019, in a separate file named data.csv.

1. Try to understand how data is stored in data.csv file, and think about how to parse data.
   * CSV is a flat file format describing values in a table. Each record consists of M values, separated by commas. The last value is followed by a new line instead of a comma.

2. Try to understand what below code does:

```
with open('data.csv', 'r') as data_file:
    lines = data_file.readlines()
    # why did we use strip and split?
    columns = tuple(lines[0].strip().split(',')[1:])
    for row in lines[1:]:
        row_data = row.strip().split(',')
        data[row_data[0]] = tuple(row_data[1:])
    data_file.close()
```

3. Create a new file named corruption.csv that contains only country names and corruption statistics.