COMP10001 Foundations of Computing
Semester 2, 2022

Tutorial Questions: Week 9

— VERSION: 1479, DATE: SEPTEMBER 19, 2022 —

Discussion

1. What is a list comprehension? How do we write one and how do they make our code simpler?
2. What happens if we use curly brackets instead of square brackets around a "list" comprehension? What happens if we use parentheses?

   Now try Exercise 1
3. Why do we use files? Could we use computers without them?
4. What are the steps to reading and writing files?
5. What is a csv file and why is it useful for storing and manipulating data?

   Now try Exercises 2 & 3

Exercises

1. Evaluate the following list comprehensions. For each one, also write some python code to generate the same list without using a comprehension.

   (a) `[(name, 0) for name in ("evelyn", "alex", "sam")]`

   (b) `[i**2 for i in range(5) if i % 2 == 1]`

   (c) `"".join([letter.upper() for letter in "python"])`

2. Fill in the blanks in the program below which reads from `in.txt` and writes to `out.txt`.

```
outfile = ____1____("out.txt", "w")
with open("in.txt", ____2____) as infile:
    line_no = 1
    for line in ____3____:
        outfile.____4____(f"line: {line_no}, length: {len(line)}\n")
        line_no += 1
outfile.write("The End")
____5____
```

3. "travel.csv" is a csv file containing data on how people get to work in different cities in Australia. "process.py" is a python program which processes this data. What information does the "process.py" attempt to find and print? How could we edit it to find different statistics?

   `travel.csv`

```
City,Train,Tram,Bus,Ferry,Car,Total
Melbourne,242969,55169,31937,783,1282997,1613855
Sydney,368572,3210,138340,9007,1206350,1725482
Adelaide,13715,4137,33673,211,390360,442102
Brisbane,62069,229,58228,3761,663353,787650
Perth,56417,223,37899,373,594571,689489
```

   Data Source: Census of Population and Housing, 2016, TableBuilder

   `process.py`

```
import csv

fp = open("travel.csv")
city = ''
curr_max = 0.0
for row in csv.DictReader(fp):
    ferry = int(row["Ferry"])
    total = int(row["Total"])
    if ferry / total > curr_max:
        city = row["City"]
        curr_max = ferry / total
print(city)
```

Problems

1. Let's head back to worksheet 8 and solve "For a While ..." again! **Using a list comprehension**, rewrite the function `allnum` that takes a list of strings, and returns a list of those that exclusively contain digits. `allnum(['3', '-4', '5', '3.1416', '0xfff', 'blerg!'])` should return `['3', '5']`.

2. Remember the "Gamertag Problem" from worksheet 8? Rewrite the `make_gamertag` function that takes a `name` string and returns a string with a hyphen after each letter, but this time, **using a list comprehension**. `make_gamertag('Alex')` should return `'A-l-e-x-'`.

3. You've found a secret message:

   `secret_message.txt`

   ```
   pf wizveu,
   nv riv gcreezex r jligizjv grikp wfi ave'j szikyurp.
   zk svxzej rk 5gd fe jleurp.
   gcvrjv nvri r k-ivo tfjkldv - zk'j r uivjj lg grikp nzky r uzefjrli
   kyvdv.
   ufe'k cvk lj ufne!
   ```

   All that you know about the message is that it it encrypted by a basic shift cipher (also known as a Caesar cipher, where each letter is shifted by some constant number of places in the alphabet), any alphabetic character in the message is lowercase, and that it contains the string segment `'plan'`.

   Write a function to decrypt the message that takes an `infilename`, `outfilename` and `segment` (all strings), You can use a brute-force approach (try all possible values) to guess the number to shift by. You might find the functions, `ord(character)` and `chr(number)`, useful!