Friday, 21 August 2020

This lab contains four main topics, namely Expression, String, and Files.

1 Expression (5 mins)

1.1 Objectives

1. Be familiar with expression
2. Can read input from the keyboard and use it correctly

1.2 Description

Read a degree from the keyboard and convert it to radian. Compare your result with function

```
math.radians()
```

In python, if you want to use some predefined libraries, you can use it by using the **import** *library name*. For example, if you want to use the library **math**, you must add the code below at the beginning your file.

```
import math
```

To use the function from library, you can call:

```
import math

d30=math.radian(30)
d45=math.radian(45)
print('30 degrees =',d30,'radians')
print('45 degrees =',d45,'radians')
```

The resulting output should be:

```
30 degrees = 0.5235987755982988 radians
45 degrees = 0.7853981633974483 radians
```

1.3 Procedure

Your program must read the degree from the keyboard, calculate your answer using the equation:

$$radian = \frac{degree}{180} * \pi \tag{1}$$

Then, print your calculated result in the first output line and the result from **math.radians()** in the second output line. *Hint:*

In python, you can use **math.pi** to refer to π.

1.4 Sample Output Screen

```
Enter degree: 35
From equation: 0.6108652381980153
From math.radians: 0.6108652381980153
```

2 String (10 mins)

2.1 Objectives

1. Can use the command **in** correctly
2. Can use loop with string

2.2 Description

Read a string from the keyboard and count the number of characters in that string.

2.3 Procedure

Read a string from the keyboard. Remove all punctuation symbols. Count all characters that are character or digit.

Display the string after all punctuations removed and the number of characters.

Hint: You can use **len(***str***)** to find length of a *str*

2.4 Sample Output Screen

```
Input sentence: My ID is "5932093921".
Remove punctuations: MyIDis5932093921
Total characters: 16
```

```
Input sentence: This is my cat. It’s called neko.
Remove punctuations: ThisismycatItscalledneko
Total characters: 24
```

3 File (15 mins)

3.1 Objectives

1. Be familiar with file reading operation
2. Know the **split()** function
3. Can write a little bit longer program

3.2 Description

Read the `score.csv` file, which stores score of students. The format of the file is shown in the box below. Each line store the ID of students and scores from five quizzes separated by ",". Find the highest total score of the student.

```
5995778521,4,0,4,6,7
5974831121,6,5,0,6,4
5992211621,9,2,0,0,4
5987519321,3,5,1,4,3
5979431521,5,7,1,8,2
5976635721,5,6,5,5,1
5911249921,0,6,1,3,7
5906374521,9,0,7,3,8
5946204221,8,4,4,1,6
5963287221,4,0,5,5,9
```

3.3 Procedure

Read all lines from the file. In each line, you have to split all score and id using *str*.**split(',')** which will split the *str* into a list of strings using the ',' as a delimiter.

For example, this code

```
s='5963287221,4,0,5,5,9'
print(s.split(','))
```

will create the list

`['5963287221', '4', '0', '5', '5', '9'].`

If you want to find summation of all scores, you must iterate through all scores pointed by indices `1, 2, 3, 4,` and `5`. The code should be:

```
s='5963287221,4,0,5,5,9'
tokens=s.split(',')
sum=0
for i in range(1,6):
  sum=sum+int(tokens[i]) # int(str) will convert str
    into an integer
print(sum)
```

3.4 Sample Output Screen

```
Highest total score is 38.
```

4 PM2.5 (10 mins)

4.1 Objectives

1. Be familiar with file reading operation
2. Know the **split()** function
3. Can write a little bit longer program

4.2 Description

Read the `pm25.csv` file, which stores the pm2.5 level as in the format below. Find the highest pm2.5 level and date/time of that level. Please be noticed that in the case of missing value, the value of the pm2.5 will be '-'.

```
2018-12-01 00:00:00,42
2018-12-01 01:00:00,44
2018-12-01 02:00:00,38
2018-12-01 03:00:00,40
2018-12-01 04:00:00,38
2018-12-01 05:00:00,40
2018-12-01 06:00:00,48
2018-12-01 07:00:00,57
2018-12-01 08:00:00,57
2018-12-01 09:00:00,57
```

4.3 Procedure

Read all lines from the file. In each line, you have to split date/time and pm2.5 level using *str*.**split(’,’)** which will split the *str* into a list of strings using the ’,’ as a delimiter.

For example, this code

```python
s='2018-12-01 09:00:00,57'
dt,pm25=s.split(',')
print(dt)
print(pm25)
```

will show the result below.

```
2018-12-01 09:00:00
57
```

4.4 Sample Output Screen

```
The highest pm2.5 is 120.
The highest date/time is 2019-01-11 06:00:00.
```