Strings - Tutorial 5 - Parsing Text

Overview :

You will practice using strings in a Python program. The sequence operations will be used with strings, and several string methods are available for more advanced behaviors.

System Requirements:

- **Computer**: A computer is required to complete this tutorial. Any OS should work.
- **Python:** You can use the online Python compiler (Online Python Compiler) or a Python system of your choice.

Background:

String Parsing :

You are going to write a program to extract, or *parse*, the required information data from a given string. You can create your own string with your list from the previous tutorial or use the example provided.

Problem Statement:

- Given: A string or file containing the list of items. The string should be comma, newline, or space delimited.
- Complete: Generate a program to extract, or parse, the required information from the string. The extracted information should be printed to the screen.

Example Data to be Parsed:

```
Name, ID, Price, Quantity
Motor, 103, $357.45, 2
Wheel, 78, $48.23, 4
LiDAR, 1089, $2500.00, 1
Computer, 99, $725.78, 1
Battery, 401, $501.01, 2
Battery Controller, 700, $58.99, 1
```

(see tutorial5_raw_data.txt for data file)

Program Minimum Requirements:

The program should accomplish the following tasks.

1. Part 1

   (a) Initialize a single string named `raw_data` containing all the data with comma, space, or newline deliniation.

   (b) Extract the names of each of the items in `raw_data`, and store all the names in a list called `items`.

   (c) Extract the price of each item in `raw_data`, and store the prices in a list called `prices`.

2. Part 2

   (a) Determine the most expensive item, and print the name and price to the screen.

   (b) Determine the least expensive item, and print the name and price to the screen.

   (c) Determine the total cost of all the items combined.

3. Optional Advanced Features:

   - Show the items with prices in order from least to greatest price.

   - Show the items with prices in alphabetical order by name.

   - Modify the program to parse the string from a data file instead of a hardcoded literal.

   - Modify the program to store the parsed data in a Python dictionary.

Example Code:

```python
# Lists - GSET - Summer 2023
```

Part 3 - Testing:

1. Complete the Python code to the solve the problem described.

2. Test your code with different inputs. Is the answer correct? How do you know? Are there certain inputs that do not work?

3. Save your code with the download button or use copy and paste. You can view and edit the code in any text editor. Also, save a copy of the program output for your tutorial summary.

Solution Code:

```
COMING SOON
```

Tutorial Summary:

Write a brief summary of what you accomplished and what you struggled with the most.

Include the following items in the summary:

- a copy of the output of your program
- a description of what the program does and how to use it

Submission:

Use the appropriate assignment folder on ilearn to submit your program and summary. Submit the following items with your TNTech username in the filenames as shown below.

Files for Tutorial 5 (TNTech Username : twhill21)

- Tutorial Summary: **twhill21_summary5.txt**
- Python Source Code: **twhill21_tutorial5.py**

Tutorial Complete:

Congratulations, after completing *Tutorial 5 - The Shopping List*, you have learned the basics of Python! You are now ready to start learning about boolean logic and program flow.