Read questionnaire from spreadsheet into a dictionary

Questionnaire data structure:

Questions are a class containing each of the following. Some could be a list or dictionary

- Question
- Notes
- Possible answers
  - Scores to go with answers

The whole questionnaire should be an ordered list so remain in order containing all the instances of question

Function: Read in questionnaire

In questionnaire.py

Dependencies:

- Google sheets access

Libraries:

- import Gspread

pip3 install gspread google-auth

```
class Question:
    """Creates question instance"""
    Def __init__(self, question_info, max_poss_score, options)
        Self.question_info = question_info
        Self.max_poss_score = max_poss_score
        Self.options = options



#List to hold questions so ordered
questions = []
summary = ""

def get_questionnaire()
    first_step = True
    loop through all rows
        if row data in column A contains "Step"
            if first_step == True
                first_step = False
            else
                question_instance = Question(question_info, max_poss_score, options)
            #held in a list so ordered
            options = []
            question_info = data in column B
            max_poss_score = data in column C
        elif row data in column A contains "Option"
            option = {}
            option["option_detail"] = data in column B
            option["score"] = data in column C
            options += option
        elif row data in column A  =  "Summary"
```

summary = data in column B

Function: String_wrap
This function takes long text data from the spreadsheet and adds in a whitespace \n before the 70th character which is the max desired line length.

Convert string to list so can be modified. Start at 0 index and add 70. If the character in index 70 is a whitespace (" ") the insert "\n" else count back through the list to the last whitespace (" ") and insert there. Add 70 to that index and repeat.

Main software process:

Loop through the questionnaire in a function called from main displays each question and obtains an option selection from the user. The question function loops through the questions and stores the users responses in the user variable it created. After all the questions have been answered, all the data is stored in the correct place in the spreadsheet with user id and score.

User class:

The outputs to be put into the spreadsheet will be an instance of a class called user that contains all the user's responses to each of the questions. User contains a value questions with a dictionary in it with as many answers as necessary.

Class user:

"""Creates a user instance"""

Def __init__(self, responses, total_score)

Self.responses = responses

Self.total_score = total_score

Add a user_previous class that uses 'user' as a parent and adds the previous scores held in the spreadsheet to it. This makes the code extensible and adds features that can be removed and modified without effecting the original features making the code modular. It also means a single variable can be passed to, for example,e the questionnaire function and if previous data is present it can be compared to current data

Function: Random user id generator:

dependencies

Random library

String library

Function: Choice function to pick from a range of characters and numbers

def user_id ()

"""

Generates a random 5 character alphanumeric user id from a pool of upper and lower case letters and the numbers 0 -9

"""

num_char_pool = list(string.ascii_letters) + list(range(10))

```
user_id = “”
For x in range(5):
        user_id += random.choice(num_char_pool)
return user_id
```

In calling function, check if user id already in list of users (retrieve from spreadsheet if necessary) if it is then re-call user_id

If not then give the user id to the user and record it in the spreadsheet user table with their data.