Name: Chen Wang

Date: 2022/03/17

Assignment Title: Individual Programming \_\_3 (Part2)\_\_\_

Summary of how the program should run:

*# 1~3 read a relative link, storage vocabulary, and give its definition, then save it  
 1: Open a relative link from a complete URL then return how many vocabularies in the relative link  
 1a: Give a word and return its relative link  
 1b: Use Beautifulsoup, parser and decode defined URL  
 1c: Get vocabularies from this relative link  
 1d : 1d: Feedback to the user how many vocabularies was found in these relative links  
 2: check if want to store a definition or not, storage a vocabulary (from relative links) and its definition (user-defined)  
 2a: Request to key-in a word and store its definition when the user feedback 'Y'  
 2a.1: Give a warning when vocabulary's definitions are not empty  
 2a.2: Request a definition  
 2a.3: Store it in a defined dictionary  
 3: After completing to storage vocabulary and its definitions, save file and print vocabulary and its definition to a file  
 3a: check the longest characters in the storage dictionary  
 3b: Request a filename from the user  
 3c: follow the order to store the term(vocabularies/definitions to the file)*

List of changes made to the original assignment:

*# 1e: Search for the word meaning from the dictionary website, and store the first definition.*

*# 1e.1: Open dictionary website from dictionary.com ex: 'https://www.dictionary.com/browse/bazaar'  
 # 1e.2: Store all the information into the dictionary*

*# 1f: Create a sheet to store an example (sentence) from the article which is relative to the new word  
 # 1f.1: Store all sentences from the article  
 # 1f.2: Compared to the vocabulary list, store them in a new create dictionary*

*# 2b: Ask the user if want to replace the old one  
 # 2b.1: Give a warning when vocabulary's definitions are not empty  
 # 2b.2: Request a definition  
 # 2b.3: Store it in a defined dictionary*

Examples of the program (please also upload any attachments necessary):

*# 1e: Search for the word meaning from the dictionary website, and store the first definition.  
# 1e.1: Open dictionary website from dictionary.com ex: 'https://www.dictionary.com/browse/bazaar'***for** i **in** list(dictionary.keys()):  
 result = urlopen(CheckDict(i))  
 soup\_Dict = BeautifulSoup(result, **'html.parser'**)  
 a\_tags = soup\_Dict.findAll(**'span'**, attrs={**'class'**: **'one-click-content css-nnyc96 e1q3nk1v1'**})  
 Word\_Temp = []  
 *# 1e.2: Store all the information into the dictionary* **for** tag **in** a\_tags:  
 dreamstring = **"definition"** Word\_Temp.append(tag.text)  
 dictionary[i] = Word\_Temp[0]

Errors that will crash the program or cause illogical problems:

It’s my second time to use try/except to pass through all the coding. Until this extension assignment, I did not think it will be a problem for a programmer. When you just use try/except, any error will be considered to follow the except path. As a result, it will be difficult to debug. As a result, it unavoidably wastes me more time to handle the whole project

I consider two ways to keep them from happening:

1. Clearly use except: Use ‘except ValueError:‘ instead of just using ‘except:’
2. Use except function only at the position you want to use: narrow down its scope. (I need to learn how to clearly use any function in python coding because each useful coding will usually be>10000 lines).

Additional examples or other comments/notes:

*# 1f: Create a sheet to store example (sentence) from the article which is relative to the new word  
# 1f.1: Store all sentence from the article.*new\_string =[]  
story = soup.findAll(**'div'**, attrs={**'class'**: **'StoryPara'**})  
**for** para **in** story:  
 detail = re.split(**r'[.;!?\t|]'**, para.text)  
 **for** additem **in** detail:  
 **if** len(additem) > 0:  
 **if** additem[0] == **' '**:  
 new\_string.append(additem[1:])  
 **else**:  
 new\_string.append(additem)  
  
*# 1f.2: Compared to the vocabulary list, store them in a new create dictionary***for** key **in** list(example.keys()):  
 **for** sentence **in** new\_string:  
 **if** key **in** sentence:  
 example[key] = sentence  
 **break**

*# 2b: Ask the user if want to replace the old one*Warningcheck = **True  
while** Warningcheck:  
 Warningask = input(  
 **'Would you \033[94m\033[1mSTILL\033[0m like to update a definition (Y/N)? '**)  
 *# 2b.1: Replace when user say 'Yes'* **if** Warningask == **'Y' or** Warningask == **'Yes' or** Warningask == **'yes' or** Warningask == **'y'**:  
 definition = input(**'Definition: '**)  
 dictionary[term] = definition  
 Warningcheck = **False** *# 2b.2: No store when user say 'No'* **elif** Warningask == **'N' or** Warningask == **'No' or** Warningask == **'no' or** Warningask == **'n'**:  
 Warningcheck = **False** *# 2b.3: Keep asking when use neither say 'Yes' or 'No* **else**:  
 print(**'Please enter Y/N again.'**)