

Dictionaries Challenge 21: Thesaurus App

Description:

You are responsible for writing a program that simulates a thesaurus. Your program will present a user with a list of words that your thesaurus contains. Based on the users choice, you will randomly present them with a synonym for their chosen word. Lastly, your program will display all of the potential synonyms for each word in the thesaurus.

Step By Step Guide:

- Create a dictionary called "thesaurus".
 - You must have a minimum of four keys in the dictionary.
 - Each key should be a **string** of a word of your choice.
 - The associated values for each key should be a **list** containing five synonyms for the key.
- For example, my dictionary includes:

```
"hot":['balmy', 'summery', 'tropical', 'boiling', 'scorching'],
"cold":['chilly', 'cool', 'freezing', 'frigid', 'polar'],
"happy":['content', 'cheery', 'merry', 'jovial', 'jocular'],
"sad":['unhappy', 'downcast', 'miserable', 'glum', 'melancholy'],
```
- Print a welcome message.
- Print what words are in your thesaurus.
 - This is represented by the keys of your dictionary
- Ask the user what word they would like to get a synonym for.
- If the users choice is in the thesaurus:
 - Choose a random synonym from the list containing the synonyms for the word.
 - Type import random as the first line of your program.
 - Display the information.
- Else, the word is not in the dictionary:
 - Inform the user.
- Ask the user if they would like to see the whole thesaurus.
 - This is represented by both the keys and values.
- If yes:
 - Display the whole dictionary following the format below.
- Else:
 - Print a goodbye message
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output 1:

Welcome to the Thesaurus App!

Choose a word from the thesaurus and I will give you a synonym.

Here are the words in the thesaurus:

- hot
- cold
- happy
- sad

What word would you like a synonym for: Hot

A synonym for hot is boiling.

Would you like to see the whole thesaurus (yes/no): yes

Hot synonyms are:

- balmy
- summery
- tropical
- boiling
- scorching

Cold synonyms are:

- chilly
- cool
- freezing
- frigid
- polar

Happy synonyms are:

- content
- cheery
- merry
- jovial
- jocular

Sad synonyms are:

- unhappy
- downcast
- miserable
- glum
- melancholy

Example Output 2:

Welcome to the Thesaurus App!

Choose a word from the thesaurus and I will give you a synonym.

Here are the words in the thesaurus:

- hot
- cold
- happy
- sad

What word would you like a synonym for: mad

I'm sorry, that word is not currently in the thesaurus.

Would you like to see the whole thesaurus (yes/no): no

I hope you enjoyed the program. Thank you!

Dictionaries Challenge 22: Database Admin Program

Description:

You are responsible for writing a program that will simulate logging into a database and prompting a user to change their password. All usernames and passwords to the database will be stored in a dictionary. Upon entering the correct credentials, your program will prompt the user to enter a new password that is a minimum of eight characters long. If the new password meets the criteria, it will be accepted, otherwise the new password will be rejected. If the user who logged in is the admin, a list of all usernames and passwords will be displayed.

Step By Step Guide:

- Print a welcome message.
- Create a dictionary called "log_on_information".
 - The key-value pairs in this dictionary will be username:password.
 - Store 5 keys.
 - Each key should be a string representing a username.
 - One username must be "admin00".
 - Store 5 values.
 - Each value should be a string representing a password at least 8 characters long.
- Get user input for their username.
- Follow the following conditional logic to control user interaction.
- If the username is in the database, get user input for their password.
 - If they enter the password correct, greet them with a message.
 - If the user that logged in is the admin00, display the whole dictionary.
 - Else, if the user who logged in is not the admin, ask the user if they would like to change their password.
 - If the user wants to change their password, ask them for their new password and inform them that the password must be 8 characters long.
 - If the password is not 8 characters long, do not accept.
 - Else, If the password is 8 characters long or more, accept and display the username and new password.
 - Else, if the user does not want to change their password, print a goodbye message.
 - Else, if the user enters their password incorrectly, inform the user.
 - Else, If the username is not in the database, inform the user.
 - Use at least 2 comments to describe sections of your code.
 - “Chunk” your code so that is readable.
 - Use appropriate and informative variable names.
 - Format your output as below.

Example Output 1:

Welcome to the Database Admin Program

Enter your username: admin00

Enter your password: admin1234

Hello admin00! You are logged in!

Here is the current user database:

Username: mooman74	Password: alskes145
Username: meramo1986	Password: kehns010101
Username: nickyD	Password: world1star
Username: george2	Password: booo3oha
Username: admin00	Password: admin1234

Example Output 2:

Welcome to the Database Admin Program

Enter your username: mooman74

Enter your password: alskes145

Hello mooman74! You are logged in!

Would you like to change your password: yes

What would you like your new password to be: 123456789

mooman74 your password is 123456789

Example Output 3:

Welcome to the Database Admin Program

Enter your username: nickyD

Enter your password: world1star

Hello nickyD! You are logged in!

Would you like to change your password: yes

What would you like your new password to be: helpme

helpme not the minimum eight characters.

nickyD your password is world1star

Example Output 4:

Welcome to the Database Admin Program

Enter your username: george2

Enter your password: happy

Password incorrect!

Example Output 5:

Welcome to the Database Admin Program

Enter your username: billbobagins

Username not in database, goodbye.

Dictionaries Challenge 23: Yes or No Polling App

Description:

You are responsible for writing a program that will conduct a poll on a yes or no issue. Upon starting the program a user will be prompted for an issue to vote on, the number of voters, and a password to view the poll results. Your program will then conduct the poll. Each time a user votes, your program will ask for the voter's full name to verify that they have not yet voted. If the voter has not yet voted, they will be presented with the issue and can vote yes or no. The vote will be recorded. Once the number of voters specified by the user has been reached, the poll will close and a summary will be displayed. If the user enters the correct password a result of each voter's name and how they voted will be displayed.

Step By Step Guide:

- Print a welcome message.
- Get user input for the issue that the voters will be voting on.
- Get user input for the number of voters that will be allowed to vote.
- Get user input for a password to be used to view the polling results.

- Define a variable to count the number of yes votes and initialize it to zero.
- Define a variable to count the number of no votes and initialize it to zero.
- Define a blank dictionary that will hold the results of the poll.
 - Each key in the dictionary will be a person's full name
 - The associated value will be their vote.

- Simulate the polling process by using a for loop the appropriate number of times.
- Each iteration of the loop:
 - Ask the user to enter their full name.
 - If the user's name has already been registered:
 - Print a message that informs them they cannot vote again.
 - Else, they haven't yet voted:
 - Print a message describing the issue that is to be voted on.
 - Get user input for their choice, yes or no.
 - You should allow users to type 'yes', 'YES', 'Yes', 'y' or 'Y' for yes and similar for no.
 - Depending on the user input, increment the number of yes or no votes by one.
 - If the user entered in another option; record their vote but inform them that their vote was not yes or no and will not influence the poll.
 - Add a new element to your dictionary.
 - The key of this element should be the user's name.
 - The value of this element should be their choice; which regardless of how they entered it should be 'yes', 'no', or their invalid vote.
 - Print a message thanking the user and confirming their vote.

- Once the voting is finished, print a summary that shows the total number of voters.
- List the full names of all the voters.
- Give a final voting result.
- Reprint the issue that was being voted on.
- Use an if/elif/else chain to print an appropriate message if yes won, if no won, and if it was a tie.
- Get user input for a password to see the voting results, vote by vote.
- If the user enters the password correct:
 - Print all full names and their corresponding votes.
- Else:
 - Let the user know the password was not correct.
- Print a message thanking the user for using the program.
- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output 1:

Welcome to the Yes or No Issue Polling App

What is the yes or no issue you will be voting on today: Should soda be banned for young children?

What is the number of voters you will allow on the issue: 4

Enter a password for polling results: admin1234

Enter your full name: john smith

Here is our issue: Should soda be banned for young children?

What do you think...yes or no: yes

Thank you John Smith! Your vote of yes has been recorded.

Enter your full name: madison jones

Here is our issue: Should soda be banned for young children?

What do you think...yes or no: YES

Thank you Madison Jones! Your vote of yes has been recorded.

Enter your full name: bryce leaf

Here is our issue: Should soda be banned for young children?

What do you think...yes or no: N

Thank you Bryce Leaf! Your vote of no has been recorded.

Enter your full name: madison jones

Sorry, it seems that someone with that name has already voted.

The following 3 people voted:

John Smith

Madison Jones

Bryce Leaf

On the following issue: Should soda be banned for young children?

Yes wins! 2 votes to 1.

To see the voting results enter the admin password: admin1234

Voter: John Smith Vote: yes

Voter: Madison Jones Vote: yes

Voter: Bryce Leaf Vote: no

Thank you for using the Yes or No Issue Polling App.

Example Output 2:

Welcome to the Yes or No Issue Polling App

What is the yes or no issue you will be voting on today: Should college tuition be free?

What is the number of voters you will allow on the issue: 3

Enter a password for polling results: freecollege

Enter your full name: rob alman

Here is our issue: Should college tuition be free?

What do you think...yes or no: y

Thank you Rob Alman! Your vote of yes has been recorded.

Enter your full name: tim rean

Here is our issue: Should college tuition be free?

What do you think...yes or no: Who cares...I dont.

That is not a yes or no answer, but okay...

Thank you Tim Rean! Your vote of who cares...i dont. has been recorded.

Enter your full name: mary smith

Here is our issue: Should college tuition be free?

What do you think...yes or no: n

Thank you Mary Smith! Your vote of no has been recorded.

The following 3 people voted:

Rob Alman

Tim Rean

Mary Smith

On the following issue: Should college institution be free?

It was a tie! 1 votes to 1.

To see the voting results enter the admin password: payme

Sorry, that is not the correct password. Goodbye...

Thank you for using the Yes or No Issue Polling App.

Dictionaries Challenge 24: Frequency Analysis App

Description:

You are responsible for writing a program that will analyse the letter distribution of a given text. Your program will take any text, remove all non-alpha characters, count the frequency of each letter within the text, calculate the percentage of occurrence for each letter, and create a list of letters ordered from highest occurrence to lowest occurrence. Your program will perform these operations for two different bodies of text.

Step By Step Guide:

- Print a welcome message.
- Create a list called non_letters.
 - This will hold all non letter characters that may appear in a phrase the user enters.
 - Make sure to include all punctuation marks, numbers, a blank space, the newline character, and the tab character.
- Get user input for a phrase to be analyzed.
 - Store this phrase in a variable called key_phrase_1.
 - Take proper precautions to standardize the user input such that it will always be lower case.
- Remove all non letters from the phrase entered by the user.
- To do this use a for loop to loop through your list of non letters.
 - Each iteration, use the .replace() method for strings to replace the current non_letter with "; or nothing.
 - This will remove any non letters from your phrase and replace them with an empty character.
 - I would suggest looking up how to use this new method.
 - After this step is done a previously entered string such as "Hello! How are you doing today? 32 years old I'm today." would appear as, "hellohowareyoudoingtodayyearsoldimtoday"
- Store the total length of the new "cleaned up" phrase entered by the user in a variable called total_occurrences.
- Create a Counter object called letter_count.
 - A Counter is a collection where elements are stored as dictionary keys and their counts are stored as dictionary values. Counts are allowed to be any integer value including zero or negative counts.
 - For our purpose, each letter in our phrase will be a key to this dictionary and the number of occurrences will be the value.
 - Counters are outside the scope of basic Python so we will need to import an extra library of code.

- Type “from collections import Counter” as your first line of code in your program.
 - This will import the Counter dictionary subclass.

- To create the Counter type the following:
 - `letter_count = Counter(key_phrase_1)`
- `letter_count` will be a dictionary that has every letter as a key and the associated number of occurrences of that letter as the value.

- Print a frequency analysis of the letters used in the phrase entered by the user.
 - This should show the letter, the total number of occurrences, and the percentage that it occurred in the phrase.
 - Sort the results such that they are in alphabetical order.
 - Round the percentage to two decimal places.

- After you display the frequency analysis for `key_phrase_1`, order the letters from highest to lowest occurrence.
 - In order to do this create a variable called `ordered_letter_count`.
 - Set the value of `ordered_letter_count` equal to your Counter you created from part 1, `letter_count` and use the `.most_common()` method.
 - I would highly suggest looking up "python counter most common" and see how this method works.
 - I would also suggest experimenting with this method and see what kind of data type it returns and what each piece of information represents.

- Create a blank list called `key_phrase_1_ordered_letters`.
- Your goal is to append all of the letters from `key_phrase_1` to this list in order from most occurrences to least occurrences.
- Once you have done this, print all the letters in `key_phrase_1_ordered_letters` on one line.
 - If you can't remember how to do this look into the `end=` argument of the `print` function.

- Expand your code to allow the user to enter in a second message after the first.
- Display all the same statistics for the second message.
- Choose your variable names wisely!
- Don't overwrite any information from the first message when you run the code a second time.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output

Welcome to the Frequency Analysis App

Enter a word or phrase to count the occurrence of each letter: Hello! How are you doing today?
32 years old I'm today.

Here is the frequency analysis from key phrase 1:

Letter	Occurrence	Percentage
a	4	10.26%
d	4	10.26%
e	3	7.69%
g	1	2.56%
h	2	5.13%
i	2	5.13%
l	3	7.69%
m	1	2.56%
n	1	2.56%
o	7	17.95%
r	2	5.13%
s	1	2.56%
t	2	5.13%
u	1	2.56%
w	1	2.56%
y	4	10.26%

Letters ordered from highest occurrence to lowest:
oaydelhritwungsm

Enter a word or phrase to count the occurrence of each letter: This is pretty interesting. I feel like I'm learning something new!

Here is the frequency analysis from key phrase 2:

Letter	Occurrence	Percentage
a	1	1.89%
e	9	16.98%
f	1	1.89%
g	3	5.66%
h	2	3.77%
i	9	16.98%
k	1	1.89%
l	3	5.66%
m	2	3.77%
n	6	11.32%
o	1	1.89%
p	1	1.89%

r	3	5.66%
s	4	7.55%
t	5	9.43%
w	1	1.89%
y	1	1.89%

Letters ordered from highest occurrence to lowest:

ientsrglhmpyfkaow

Dictionaries Challenge 25: Code Breaker App

Description:

You are responsible for writing a program that will encode or decode a message based off the letter distribution of a predetermined key text. Your program will determine a frequency analysis for two texts and use these letter distributions to create a cipher to either encode or decode a message based off user input. This program is an extension of the Frequency Analysis App.

Step by Step Guide:

- Begin by copying your code from the Frequency Analysis App
- Print a new welcome message.
- Comment out the user input for getting key_phrase_1
- Hard code a predetermined phrase between you and the person you are communicating with for key_phase_1. I am using excerpts from Sherlock Holmes, but anything will do.

key_phrase_1 = " "

To Sherlock Holmes she is always the woman. I have seldom heard him mention her under any other name.

In his eyes she eclipses and predominates the whole of her sex. It was not that he felt any emotion akin to love for Irene Adler.

All emotions, and that one particularly, were abhorrent to his cold, precise but admirably balanced mind.

He was, I take it, the most perfect reasoning and observing machine that the world has seen, but as a lover he would have placed himself in a false position.

He never spoke of the softer passions, save with a gibe and a sneer.

They were admirable things for the observer excellent for drawing the veil from men's motives and actions.

But for the trained reasoner to admit such intrusions into his own delicate and finely adjusted temperament was to introduce

a distracting factor which might throw a doubt upon all his mental results.

Grit in a sensitive instrument, or a crack in one of his own highpower lenses, would not be more disturbing than a strong emotion in a nature such as his.

And yet there was but one woman to him, and that woman was the late Irene Adler, of dubious and questionable memory.

I had seen little of Holmes lately. My marriage had drifted us away from each other.

My own complete happiness, and the homecentred interests which rise up around the man who first finds himself master of his own establishment,

were sufficient to absorb all my attention, while Holmes, who loathed every form of society with his whole Bohemian soul,

remained in our lodgings in Baker Street, buried among his old books, and alternating from week to week between cocaine and ambition,

the drowsiness of the drug, and the fierce energy of his own keen nature.

He was still, as ever, deeply attracted by the study of crime,

and occupied his immense faculties and extraordinary powers of observation in following out those clues,
and clearing up those mysteries which had been abandoned as hopeless by the official police.
From time to time I heard some vague account of his doings: of his summons to Odessa in the case of the Trepoff murder,
of his clearing up of the singular tragedy of the Atkinson brothers at Trincomalee,
and finally of the mission which he had accomplished so delicately and successfully for the reigning family of Holland.
Beyond these signs of his activity, however, which I merely shared with all the readers of the daily press, I knew little of my former friend and companion.

:::::

- Comment out the user input for getting key_phrase_2
- Hard code a predetermined phrase between you and the person you are communicating with for key_phase_2. I am using excerpts from Sherlock Holmes, but anything will do.

key_phrase_2 = " "
Quite so! You have not observed. And yet you have seen.

That is just my point. Now, I know that there are seventeen steps, because I have both seen and observed.

By the way, since you are interested in these little problems, and since you are good enough to chronicle one or two of my trifling experiences, you may be interested in this.

He threw over a sheet of thick, pink tinted notepaper which had been lying open upon the table. It came by the last post, said he. Read it aloud.

The note was undated, and without either signature or address.

There will call upon you tonight, at a quarter to eight o'clock, it said, "a gentleman who desires to consult you upon a matter of the very deepest moment. Your recent services to one of the royal houses of Europe have shown that you are one who may safely be trusted

with matters which are of an importance which can hardly be exaggerated.

This account of you we have from all quarters received.

Be in your chamber then at that hour, and do not take it amiss if your visitor wear a mask.

This is indeed a mystery, I remarked. What do you imagine that it means?

I have no data yet. It is a capital mistake to theorise before one has data.

Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.

But the note itself. What do you deduce from it?

I carefully examined the writing, and the paper upon which it was written.

The man who wrote it was presumably well to do, I remarked, endeavouring to imitate my companion's processes.

Such paper could not be bought under half a crown a packet.

It is peculiarly strong and stiff.

:::::

- You should be able to encode or decode a message regardless of the key phrases chosen.
- To accomplish this, you must look at the frequency analysis of the two key phrases.
- Given a character in the secret message that is to be encoded, you must find its index in the frequency analysis of the first message.
- Then find the letter that appears at the same index in the frequency analysis of the second message.
- This is your encoding rule to encode one character as another.
- For example, the letter "o" appears at index 1 in the first frequency analysis. The letter "t" appears at index 1 in the second frequency analysis. Therefore the letter "o" would be encoded to the letter "t".
- The letter "h" appears at index 7 in the first frequency analysis. The letter "r" appears at index 7 in the second frequency analysis. Therefore the letter "h" would be encoded to the letter "r".
- Similarly, the word "oh" would be encoded to "tr" using the given key phrases.

NEW CODE TO ADD

- Ask the user if they would like to encode or decode a message.
- Ask the user for the message.
 - You should take proper precautions to make the message all lower case.
 - You should take proper precautions to remove all non-letters from their message.
- If the user chose to encode the message, run an algorithm to encode and print the message.
 - Create a blank list called `encoded_phrase`.
 - For each letter that is in the phrase:
 - Create a variable called `index` and set it equal to the index of the current letter in `key_phrase_1_ordered_letters`.
 - To accomplish this you can use the `.index()` method.
 - Google or check Python documentation on the `.index()` method.
 - Create a variable called `letter` and set it equal to the letter that appears in `key_phrase_2_ordered_letters` at the specified index.
 - Append `letter` to the list `encoded_phrase`.
 - Print the encoded message.
 - There are multiple ways to do this. You may use the `end=` argument of the `print` function or the `.join()` method for strings.
- Elif the user chose to decode the message, run an algorithm to decode and print the message.
 - Create a blank list called `decoded_phrase`.
 - For each letter that is in the phrase:
 - Create a variable called `index` and set it equal to the index of the current letter in `key_phrase_2_ordered_letters`.
 - To accomplish this you can use the `.index()` method.
 - Google or check Python documentation on the `.index()` method.

- Create a variable called letter and set it equal to the letter that appears in key_phrase_1_ordered_letters at the specified index.
- Append letter to the list encoded_phrase.
- Print the decoded message.
 - There are multiple ways to do this. You may use the end= argument of the print function or the .join() method for strings.
- Else, the user chose an invalid option and inform them.

- Use at least 2 comments to describe sections of your code.
- “Chunk” your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Example Output 1:

Welcome to the Code Breakers App

Here is the frequency analysis from key phrase 1:

Letter	Occurrence	Percentage
a	153	7.77%
b	32	1.63%
c	53	2.69%
d	81	4.12%
e	249	12.65%
f	57	2.9%
g	28	1.42%
h	117	5.95%
i	150	7.62%
j	1	0.05%
k	12	0.61%
l	85	4.32%
m	69	3.51%
n	142	7.22%
o	157	7.98%
p	27	1.37%
q	1	0.05%
r	116	5.89%
s	139	7.06%
t	154	7.83%
u	45	2.29%
v	17	0.86%
w	46	2.34%
x	3	0.15%

y 34 1.73%

Letters ordered from highest occurrence to lowest:
eotainshrlmdfcwuybgpvkxjq

Here is the frequency analysis from key phrase 2:

Letter	Occurrence	Percentage
a	103	8.17%
b	21	1.67%
c	36	2.85%
d	45	3.57%
e	169	13.4%
f	21	1.67%
g	16	1.27%
h	67	5.31%
i	88	6.98%
j	1	0.08%
k	10	0.79%
l	33	2.62%
m	29	2.3%
n	78	6.19%
o	103	8.17%
p	26	2.06%
q	3	0.24%
r	72	5.71%
s	77	6.11%
t	135	10.71%
u	46	3.65%
v	15	1.19%
w	29	2.3%
x	3	0.24%
y	35	2.78%

Letters ordered from highest occurrence to lowest:
etoainsrhudcylmwpbfgvkqxj

Would you like to encode or decode a message: encode
What is the phrase: Wow, this is awesome!

The encoded message is:
mtmorisisamestce

Example Output 2:
Welcome to the Code Breakers App

Here is the frequency analysis from key phrase 1:

Letter	Occurrence	Percentage
a	153	7.77%
b	32	1.63%
c	53	2.69%
d	81	4.12%
e	249	12.65%
f	57	2.9%
g	28	1.42%
h	117	5.95%
i	150	7.62%
j	1	0.05%
k	12	0.61%
l	85	4.32%
m	69	3.51%
n	142	7.22%
o	157	7.98%
p	27	1.37%
q	1	0.05%
r	116	5.89%
s	139	7.06%
t	154	7.83%
u	45	2.29%
v	17	0.86%
w	46	2.34%
x	3	0.15%
y	34	1.73%

Letters ordered from highest occurrence to lowest:

eotainshrldmfcwuybgpvkxjq

Here is the frequency analysis from key phrase 2:

Letter	Occurrence	Percentage
a	103	8.17%
b	21	1.67%
c	36	2.85%
d	45	3.57%
e	169	13.4%
f	21	1.67%
g	16	1.27%
h	67	5.31%
i	88	6.98%

j	1	0.08%
k	10	0.79%
l	33	2.62%
m	29	2.3%
n	78	6.19%
o	103	8.17%
p	26	2.06%
q	3	0.24%
r	72	5.71%
s	77	6.11%
t	135	10.71%
u	46	3.65%
v	15	1.19%
w	29	2.3%
x	3	0.24%
y	35	2.78%

Letters ordered from highest occurrence to lowest:

etоainsrhudcylmwpbfгvкqxj

Would you like to encode or decode a message: decode

What is the phrase: mtmorisisamestce

The decoded message is:

wowthisisawesome