Girls' Programming Network

Secret Diary Chatbot Extension!

Add extra layers of security to your secret diary to keep out nosy siblings or friends!

This project was created by GPN Australia for GPN sites all around Australia!

This workbook and related materials were created by tutors at:

Sydney, Canberra and Perth

Girls' Programming Network

If you see any of the following tutors don't forget to thank them!!

Writers

Courtney Ross
Alex McCulloch
Renee Noble

Testers

Ash Liu
Aadeeba Mou
Taylah Griffiths
Sheree Pudney

Extension 1: It's a trap!

Sometimes siblings like to get in your stuff, so we're going to set a trap for anyone trying to sneak into our secret diary. Let's make a decoy password that shows them a fake diary if they enter it

Task 6.1: Decoy Password

1. In the same place that you set **`password`**, make a new variable called **`decoy_password`** and set it to something else that you think your friends or siblings might guess

Task 6.2: Let them through!

If they guess the **`decoy_password`** while they are trying to get into the secret diary, we want to let them past the while loop.

1. Add an `and` condition to the while loop so that it keeps looping if the **`guessed_password`** isn't the real password and the **`guessed_password`** isn't the decoy password.

Task 6.3: Fake news!

We need to make sure that if they guessed the **`decoy_password`**, they don't get into our real diary and get a fake diary instead!

1. Create an `if` statement after the `while` loop that checks if the **`guessed_password`** is the same as the **`decoy_password.`**

2. Inside the `if` statement, print out whatever you want the fake diary to be! For example:

```
Here's the secret diary (I hope you can read gibberish!)
aghioahgiagipsjgij sagagieja iahgoaihg aoireioaghoiag io
wiohga aowihgaiog aohireg hai ogaa oiarghag iroagoi aoihgri
oaehr iohaihagh oiaegroha oaihgehaeo arheioargh oihrg haeo
```

Task 6.4: Could the real Secret Diary please step forward?

We still want whoever puts in the real password to have access to the diary

1. Add an `else` after the `if` that we just wrote and put all of the code below inside of it so that it only runs when the password isn't the decoy.

Hint: Indenting multiple lines

It can be a little annoying when you have to add a tab to the front of a lot of lines. In IDLE you can select all of the lines that you want to indent (that means, add a space in front of) and press the TAB key once and it will move all of the lines you had selected! Aren't computers cool?

☑ CHECKPOINT ☑

If you can tick all of these off you can go to the Extensions:

- ☐ You have a decoy password
- ☐ When the user puts in the decoy password they see the fake diary
- ☐ When the user puts in the real password they can still get the real diary

Extension 2: What's your style?

Task 7.1: Give it a fun theme!

In the instructions, we've based it on keeping a secret diary safe from our little siblings! **But you can add your own theme!!**

Think about a theme you want to add to your chatbot! Some ideas are below!!

A spy theme!!

The chatbot protects the secret identities and speaks like a secret agent.

A secret club for a school project!!

Keep copycats at bay and teach them a lesson for trying to steal your ideas! Teach cheaters a lesson!

Pokemon Go!!

Use your chatbot to store the secret locations of rare pokemon!

A fight between Good and Evil!

Every team has to store their secret info somewhere whether they are in Dumbledore's Army, The Rebel Alliance or the Powerpuff Girls!

Task 7.2: S-S-S-Style!

Use your theme!

1. Change, or add in more print statements to reflect your chosen theme!
2. Change your questions and answers so they match the theme as well!

Extension 3: Intelligent Checks!

Task 8.1: You're so smart!

You know lots of things and you can prove it! Let's add another level of security that checks if you are smart enough to read the diary.

1. Find the part of the code before we ask the user to enter the password.
2. Use `input` to ask the user the answer to an intelligence question. For example: `"What is 2 + 2? "` and store it in a variable called **`answer`**.

Task 8.2: Or are you?

Now we need to check whether or not the answer is right!

1. Write an `if` statement that checks if the **`answer`** that they gave does NOT equal the right answer.
2. If the answer is wrong, `print` something to let them know that they aren't smart enough to get into your diary and use **`exit()`** to make the program stop right there.

Task 8.3: I know lots of things!

You can add as many of these questions as you want by copying your code from 8.1 and 8.2 and changing the question and answer! See if the person next to you can get past your intelligence checks!

Extension 4: Make it Funnier!

Task 9.1: More security, more!

Think about jokes and tricks to add to your chatbot when the user guesses the decoy password. Some ideas are below! If you haven't done extension 1 yet, do that first and come back.

A Magic 8 Ball!

Trick imposters into thinking that your secret is a magic 8 ball. Distract the imposter with your magic 8 ball!

```
Ask the magic 8 ball for
advice: Will it rain today?
>>> Yes
Ask another question: Should I
go shopping?
>>> Maybe
Ask another question: Will it
rain today?
>>> Ask again later
```

ASCII Art

Add some ASCII art for fun!!!

```
Here is the super secret info
...
                _=,_
             o_/6 /#\\
             \__ |##/
              ='|--\\
                /   #'-.
                \#|_   _'-. /
                 |/ \\_( # |"
                C/ ,--___/
Woof...
```

Task 9.2: Magic 8 Ball

Let's create a Magic 8 Ball!

1. At the top of your code, `import random`. This will let us choose things randomly!
2. Create a `list` called `answers`. Add in some Magic 8 answers to this list, like `"Yes"`, `"No"`, or `"Maybe"`. Add as many answers to this list as you like!
3. Use `input` to ask the user what question they'd like the answer to.
4. Use `random.choice` to select an answer from the answers list. Tell the user!
5. Add a `while` loop, so your code keeps asking questions and giving random answers.

Hint: Lists

You can create a list of your favourite foods like this:

```
myFave = ["pizza", "chocolate", "nutella", "lemon"]
```

Hint: Random choice

You can use `random.choice` to select a random number between 1 and 3 like this:

```
mynum = random.choice([1, 2, 3])
```

Task 9.3: ASCII art

Let's show some ASCII art to our secret agents!

1. Pick some ASCII art! You can choose from http://www.ascii-art.de/ascii/, or make your own!

2. Copy and paste it into your program. You will need to `print` your ASCII art.

Note: If your picture doesn't print out exactly as you expect you might have quotes inside it that are breaking your print. Edit the picture, or choose another!

Hint:

You can use 2 special tricks to print your art:

- triple quotes to `print` things on multiple lines using one `print` statement.
- Put the letter r just before your multiline string. This makes it a **raw** string. *Normal strings do special things with characters like `"\"`. We don't want that.*

```
print(r"""This is
A really really
Long answer""")
```

Extension 5: Enc🍊ding in Em🍊ji

Right now our diary is protected from snoops when they use the chatbot, because they need the password. **But what if someone just goes and opens the text file and reads it!**

We'll make our words unreadable by encoding them with emoji!

Task 10.1: Apple-ify your message!

We'll start by replacing any letter "a" with an apple emoji.

1. Go to the place in your code after you ask the user for their diary entry. (before you write it to the diary file.)

2. To write or read our emojis in the file, we need to specify the encoding. Go to the two lines in your code, where you open the file and add `encoding="utf-8"` to the command. It should look like this:

   ```
   f = open("secret.txt", "r" , encoding="utf-8")
   ```

   We'll start by replacing all of the letter "a" with the 🍎 emoji using this line of code:
   `diary_entry = diary_entry.replace("a", "🍎")`

   To get the emoji you can go to https://emojityper.com/, search for apple and copy it.

3. Emoji are different to normal letters so we need to tell Python to do things a little differently when we write to the file. Add encoding = "utf-8" at the end of your file open command so it looks like this: open("secret_diary.txt", "a", encoding="utf-8")

4. Test your code!
   - Run your code, and add a new message (make sure it includes the letter "a")
   - Look in your secret diary file by hand, does it have an apple?
   - Try and read your diary with your program.
     …. Whoops there's still apples there. We haven't written the decoder yet!

Task 10.2: De-apple-ify your diary!

When we want to read the diary again we'll need to put the letters back in, and remove the emoji!

1. Go to the place in your code after you read the content of the diary.

2. We'll do the opposite of what we did before. Use replace again. But this time replace `"🍎"` with `"a"`. *(switch the order from before)*

3. Add the encoding = "utf-8" code when you open the file for reading here too! It should look like open("secret_diary.txt", "r", encoding="utf-8")

4. Test your code!
   a. Run your code, and add a new message (enter a message with an "a" in it)
   b. Try and read your diary with your program. Are the apples gone?

Task 10.3: More emoji is more secret!

The more letters you replace the harder it will be to understand any words.

1. Repeat the encoding and decoding process above, but for different letters and emoji. You can use *emojityper* to find more emoji to copy.