Girls' Programming Network

2017

Sassy Security Chatbots!
Extensions!

Extension Time!

Now that we have our sassy security chatbot running, it's time to add more features!

You can complete the extensions in this workbook in any order you like!

Make it Funnier!

So we've given our trick information to people with the decoy password, we made them do a maths test, even though we were never going to give them the secret info.

But is that enough? There's more funny things we can do to snoops!

Goal: Make it even funnier!

Task 7.1: More security, more!

Think about jokes and tricks to add to your chatbot. Some ideas are below!
Put on those thinking hats!!

A Magic 8 Ball!

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

Extension: Magic 8 ball!

Goal: Make a magic 8 ball for the imposters

Magic 8 Ball

Step 1) We need a list of responses to choose from
Store it in a variable, like this:
```
answers = ["Yes", "No", "Maybe", "Ask again later"]
```
Think of a few more to add!

Step 2) Use `helper` to ask a question!
Don't forget the question mark!

Step 3) Don't forget to sleep!

Step 4) Write a `while True` loop to never stop asking for questions
Hint: *Have a look at Part 4 from Book 1!*

Step 5) Pick a random answer
Remember how we used `random.choice([1, 2, 3])` to get a random number? Use that to pick a random answer from `answers` and print it out!

Step 6) Ask another question!

Extension: ASCII Art!

Goal: Show some ASCII art to our secret agents

ASCII art

Step 1) Pick some ASCII art
Pick something you like from http://www.ascii-art.de/ascii/ (or make your own!)

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

Step 2) Copy and paste it into IDLE
We want to `print` out our ASCII art. We can use `"""triple quotes"""` to make long strings. For example:
```
answer = """This is
          A really really
          Long
          answer"""
```

Make it More Secure!

So we now have a password and a random maths question to protect our secrets. But is that enough security?

Goal: Make it even more secure!

Task 7.1: More security, more!

This is where you can get creative and come up with your own security questions. **Put on those thinking hats!!**

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

A question only you know!

Just like Dumbledore asked Harry when Lord Voldemort returned.

What's my favourite flavour of jam?
Strawberry

A joke question

If you want to add in some fun and humour, why not add a joke?!

What do you call a bear with no teeth?
A gummy bear!

A knowledge question

If you're super smart, maybe you can ask a smart question. Just like the Ravenclaw dormitory password is the answer to a difficult question!

What's the fastest bird in the world?
Peregrine falcon at 389km/h

Complete the song lyric!

If you like Taylor Swift, you could have something like:

Cause the players gonna play, play, play, play, play
And the haters gonna:
Hate, hate, hate, hate

Let's add that extra layer of security!

Have an idea now? Great! Here are the steps you need to take to add in your extra layer of security. These questions are tailored for a security joke question, but if you're using a different idea, make sure you change the variable name!

Step 1) Let's use helper to ask our joke question and store it in a variable called `joke_response`.

Step 2) We also need to store the real answer. Use the variable `joke_answer`.

Step 3) Just like in Part 6.2, we need to put in a checking mechanism to see if they got the correct answer!
Don't forget to print them a message saying if they got it right or wrong

Step 4) Now we need to update the part of the code where we check that the person Has all the answers correct.
Hint: *Have a look at Part 6.3!*

Bonus: *Give them user more chances to get the joke right. Maybe give them a maximum number of tries! Check out Part 4 from Book 1*

★BONUS 7.3: Pick another security layer!★

There's no limit to how much security you can have. Feel free to be creative and add in as many as you can think of. Just follow the steps in Task 7.2.

Hint: *Make sure you use different variables names!*

Extensions: Make Maths Harder!!

Let's make the maths questions harder, so we can really make sure that we're the only ones who can access it!

Extension: Adding to Addition!

Only having addition is too easy especially as your snoopy younger siblings get older and better at maths!

Goal: Randomly choose from plus, minus and times for our maths question

Asking the question

Step 1) We want to randomly choose between plus, minus and times
Remember `num1 = random.choice([1, 2, 3])` helped us choose a number

Instead of choosing a number chose a mathematical operator from:
`"plus"`, `"minus"` and `"times"`
Store it in a variable

Step 2) Update how you ask your question
Before it might have said:
`What is 5 plus 7:`
But now we need to make it include our randomly chosen operator.

Hint: Add some strings together to include your mathematical operator

Calculating the answer

Calculating the answer just got a bit harder!
Before we could just say: `real_answer = num1 + num2`

But we have to do a different thing based on whether we have chosen
`"plus"`, `"minus"` and `"times"`, These match up to + - *

- **If we have chosen** `"plus"` **we want to do:**
  `real_answer = num1 + num2`
- **If we have chosen** `"times"` **we want to do:**
  `real_answer = num1 * num2`
- **And you can guess** what `"minus"` needs.

Use an if statement to calculate the `real_answer`

Hint: `if` statement and a few `elif` statements

Extension: The More Maths the Merrier!

One maths question is not enough! We want to keep asking them questions, even if they'd never get in anyway.

Asking lots of questions!

Step 1) Add a `for` loop so we can ask them 5 maths questions!

If we want to do something 100 times we can do
```
for counter in range(100):
    Do a thing in here!!
```

Step 2) Move our code into the `for` loop
We need to have all of our math code inside the for loop so it'll happen for every question. That includes finding new random numbers to calculate!

A better scoring system!

Instead of storing "passed" or "failed" for our test result, let's keep count of how many they get right by storing a number instead! **Change how we find `test_result`.**

Step 1) When you create `test_result` set it to 0.

Step 2) Every time they get the answer correct add 1 to their `test_result`!

Step 3) Change where you check if the test_result is "passed" or "failed" to check whether they got enough right. **Let's say they have to get 3 out of 5!**

Hint: Check that test_result is greater or equal to 3 to pass or less than 3 to fail.

Extension: Secret Message Board

Remember, you should only be able to read and write to the diary if you passed all our security!

Goal: Let's read and write to our secret diary!

Create the File

Step 1) In the same directory as your program, create a text file. You can name it what you like, but make sure it has the file extension "`.txt`"

Read the file!

Step 1) First we need to open the file, and store it in a variable.
Hint: Use `file = open(`"`myfakediary.txt`", "`r`"`)`, *where **myfakediary** is the name of the text file we created in Task 1, and* "`r`" *means open in read mode.*

Step 2) Read the contents of the file, and store it in the variable
Hint: use `file.read()` *to read the contents of the file.*

Step 3) Send the contents of the file to the user!

Step 4) We need to close the file when we've finished with.
Hint: Use `file.close()` *to close the file.*

Read or write?

Now that we can read our diary, maybe we want to write to it instead?

Step 1) Above the code you wrote in Task 2, ask the user if they'd like to read the diary or write to it instead. Don't forget to store their answer in a variable!

Step 2) Create an `if` statement where if the user said 'read', it opens the file in read mode. Move your code from step 2 into the if statement.

Step 3) Create an `elif` at the end of your `if` statement so if the user said 'write', it opens the file in write mode.
Hint: Use "`w`" *instead of* "`r`" *when opening the file. This will overwrite the contents of the file. If you'd like to add to the end of the file instead, use* "`a`"

And again!

Let's create a loop, so we can read our diary multiple times, or write to it multiple times!

Step 1) Create a variable called `keep_going`, and set its value to "yes"
Hint: Make sure you set this variable above all your code you've written for this task.

Step 2) Below where you created the variable `keep_going`, create a `while` loop that continues while `keep_going` is set to "yes"

Step 3) Move the code you wrote for Tasks 1 - 3 inside the `while` loop

Step 4) Below the code you wrote for Tasks 1 - 3, but still inside the `while` loop, ask if the user would like to keep going, and update the variable `keep_going`.

★Bonus Task! Multiple diaries! ★

We might have multiple diaries that we want to read and write to!

Step 1) In the same place where you created your first file, create some more text files! Make sure they have the file extension ".txt"

Step 2) Create a list of the file names! Call the list `files`.
Hint: Remember, lists are created using square brackets and a comma to separate the filenames. An example is `myList = ["I", "love", "chocolate"]`

Step 3) Create a *for* loop that loops over the `files` list and sends the names of the files to the user. Number each diary so that the user can select a number.

Step 4) Ask the user what diary they'd like to use. Convert their answer back into an `integer`!

Step 5) Get the name of chosen diary from the list, and store it in the variable `chosenDiary`

Step 6) Update the code inside the *if-elif* statement to use the `chosenDiary` variable to open the text file!

Extension: Talk to your friend

We're all on the same network (all the computers in the school are connected!) So we can talk to each other over the network using the "intranet".

Goal: Talk to a friend over the network!

Buddy Up!

Step 1) Find a partner!

Step 2) Decide who will be the server and who will be talking to the server

Get those digits!

We need to find out the IP Address of the server computer!

Step 1) in the Window menu open a program call *CMD*

Step 2) Type `ipconfig` **and press enter!**

Step 3) Take a note of your IPv4 Address.
Yours will be different to the one in the picture! But it will be in the same place!

```
C:\Python34>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : internal.usyd.edu.au
   Link-local IPv6 Address . . . . . : fe80::182:4107:e4a7:d3d%3
   IPv4 Address. . . . . . . . . . . : 10.70.12.93
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.70.12.254

Tunnel adapter isatap.internal.usyd.edu.au:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : internal.usyd.edu.au
```

Step 4) Tell your IP address to your partner!
It's like telling them your phone number!

Step 5) Run server.py!

Connecting up!

Step 1) Open your file portal.py

Step 2) Change what is stored in the ip_address variable, put in the number your partner gave you instead!

Step 3) Run server.py! Are you talking to your partner's computer now?