Task 1

In this task, you'll import a security log text file and store it as a string to prepare it for analysis.

In Python, a `with` statement is often used in file handling to open a fi
automatically close the file after reading it.

You're given a variable named `import_file` that contains the name
you want to import. Start by writing the first line of the `with` statemen
code cell. Use the `open()` function, setting the second parameter to
running this code will produce an error because it will only contain the first line of the `with` statement; you'll complete this `with` statement in the task after this. Be sure to replace the `### YOUR CODE HERE ###` with your own code.

In [ ]:
```
# Assign `import_file` to the name of the text file that contains the secu

import_file = "data/login.txt"

# First line of the `with` statement
# Use `open()` to import security log file and store it as a string

with open(import_file, "r") as file:
```

Task 2

Now, you'll use the `.read()` method to read the imported file, and you'll store the result in a variable named `text`. Afterwards, display the `text` and explore what it contains by running the cell. Be sure to replace the `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:
```
# Assign `import_file` to the name of the text file that contains the secu

import_file = "data/login.txt"

# The`with` statement
# Use `open()` to import security log file and store it as a string

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store the result in a var

  text = file.read()

# Display the contents of `text`

print(text)
```

Task 3

The output in the previous step is one big string. In this task, you'll explore how you can split the string that contains the entire imported log file into a list of strings, one string per line.

Use the `.split()` method to perform this split and then display the result. Be sure to replace the `### YOUR CODE HERE ###` with your own code before you run the following cell.

Note that displaying `.split()` doesn't change what is stored in the `text` variable. Variable reassignment would be necessary if you want to store the result after splitting.

In [ ]:
```
# Assign `import_file` to the name of the text file that contains the secu

import_file = "data/login.txt"

# The`with` statement
# Use `open()` to import security log file and store it as a string

with open(import_file, "r") as file:

  # Use `.read()` to read the imported file and store the result in a var

  text = file.read()

# Display the contents of `text` split into separate lines

print(text.split())
```

display it.

In [ ]:
```
# Assign `import_file` to the name of the text file that contains the security log file

import_file = "data/login.txt"

# Assign `missing entry` to a log that was not recorded in the log file

missing_entry = "jrafael,192.168.243.140,4:56:27,2022-05-09"

# Use `open()` to import security log file and store it as a string
# Pass in "a" as the second parameter to indicate that the file is being opened for appending purposes

with open(import_file, "a") as file:

    # Use `.write()` to append `missing_entry` to the log file

    file.write(missing_entry)

# Use `open()` with the parameter "r" to open the security log file for reading purposes

with open(import_file, "r") as file:

    # Use `.read()` to read in the contents of the log file and store in a variable named `text`

    text = file.read()

# Display the contents of `text`

print(text)
```

The next task you're responsible for is creating a text file. This text file should include a list of IP addresses that are allowed to access restricted information. Documenting this in a text file will help you communicate your findings to your security team.

Start by creating a variable named `import_file` that stores the name of the file, which should be `"allow_list.txt"`.

You're also given a variable named `ip_addresses` that stores a string containing the IP addresses that are allowed.

Run the code to display the two variables and explore what they contain. Be sure to replace the `### YOUR CODE HERE ###` with your own code before you run the following cell.

In [ ]:
```
# Assign `import_file` to the name of the text file that you want to creat

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed to acce

ip_addresses = "192.168.218.160 192.168.97.225 192.168.145.158 192.168.108

# Display `import_file`

print(import_file)

# Display `ip_addresses`

print(ip_addresses)
```

Task 6

Your next goal is to create a `with` statement in order to write the IP addresses to the text file you created in the previous step.

You'll first open the file using the `"w"` parameter. Then, you'll write the IP addresses to the file. Be sure to replace each `### YOUR CODE HERE ###` with your own code before you run the following cell. Note that the code cell will contain a `with` statement that writes to a file but does not display information to the screen, so running it will not produce an output.

In [ ]:
```
# Assign `import_file` to the name of the text file that you want to creat

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed to acce

ip_addresses = "192.168.218.160 192.168.97.225 192.168.145.158 192.168.108

# Create a `with` statement to write to the text file

with open(import_file, "w") as file:

  # Write `ip_addresses` to the text file

  file.write(ip_addresses)
```

In [ ]:

```
# Assign `import_file` to the name of the text file that you want to creat

import_file = "data/allow_list.txt"

# Assign `ip_addresses` to a list of IP addresses that are allowed to acce

ip_addresses = "192.168.218.160 192.168.97.225 192.168.145.158 192.168.108

# Create a `with` statement to write to the text file

with open(import_file, "w") as file:

    # Write `ip_addresses` to the text file

    file.write(ip_addresses)

# Create a `with` statement to read in the text file

with open(import_file, "r") as file:

    # Read the file and store the result in a variable named `text`

    text = file.read()

# Display the contents of `text`

print(text)
```