1. Create a zoo.py file first. Define the hours() function, which prints the string 'Open 9-5 daily'.
Then, use the interactive interpreter to import the zoo module and call its hours() function.
2. In the interactive interpreter, import the zoo module as menagerie and call its hours() function.
3. Using the interpreter, explicitly import and call the hours() function from zoo.
4. Import the hours() function as info and call it.
5. Create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and print it out.
6.Make an OrderedDict called fancy from the same pairs listed in 5 and print it. Did it print in the
same order as plain?
7. Make a default dictionary called dict_of_lists and pass it the argument list. Make the list dict_of_lists['a'] and append the value 'something for a' to it in one assignment. Print
dict_of_lists['a'].

- To create the zoo.py file and define the hours() function, follow the steps below:
    - Open a text editor and create a new file named zoo.py
    - Inside the file, write the following code:
      scss

```
def hours():
     print("Open 9-5 daily")
```

    -
    - Save and close the file.
- To import the zoo module as menagerie and call its hours() function in the interactive interpreter, follow the steps below:
    - Open the Python interpreter by running the command `python` in your terminal or command prompt.
    - Type the following command to import the zoo module as menagerie:
      javascript

```
>>> import zoo as menagerie
```

-
- Call the hours() function using the menagerie alias:
  scss

```
>>> menagerie.hours()
 Open 9-5 daily
```

    -
- To explicitly import and call the hours() function from zoo in the interpreter, follow the steps below:
    - Type the following command to import the hours() function from the zoo module:
      javascript

```
>>> from zoo import hours
```

-
- Call the hours() function directly:
  scss

```
>>> hours()
 Open 9-5 daily
```

    -
- To import the hours() function as info and call it, follow the steps below:
    - Type the following command to import the hours() function from the zoo module and assign it to the variable info:
      javascript

```
>>> from zoo import hours as info
```

-
- Call the hours() function using the info alias:
  scss

```
>>> info()
 Open 9-5 daily
```

    -
- To create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and print it out, follow the steps below:
    - Type the following command to create the dictionary:
      css

```
>>> my_dict = {'a': 1, 'b': 2, 'c': 3}
```

-
- Print the dictionary:
  javascript

```
>>> print(my_dict)
 {'a': 1, 'b': 2, 'c': 3}
```

    -
- To make an OrderedDict called fancy from the same pairs listed in step 5 and print it, follow the steps below:
    - Type the following command to import the OrderedDict class from the collections module:
      javascript

```
>>> from collections import OrderedDict
```

-
- Type the following command to create the OrderedDict:
  css

```
>>> fancy = OrderedDict([('a', 1), ('b', 2), ('c', 3)])
```

-
- Print the OrderedDict:
  scss

```
>>> print(fancy)
 OrderedDict([('a', 1), ('b', 2), ('c', 3)])
```

    -
    - Yes, it should have printed in the same order as the plain dictionary.
- To make a default dictionary called dict_of_lists and pass it the argument list, make the list dict_of_lists['a'] and append the value 'something for a' to it in one assignment, and print dict_of_lists['a'], follow the steps below:
    - Type the following command to import the defaultdict class from the collections module:
      javascript

```
>>> from collections import defaultdict
```

-
- Type the following command to create the default dictionary and assign the list ['list'] as the default value:
  scss

```
>>> dict_of_lists = defaultdict(list)
```

-
- Type the following command to append the value 'something for a' to the list associated with the key 'a':
  css

```
>>> dict_of_lists['a'].append('something for a')
```

-
- Print the list associated with the key 'a':
  css

```
>>> print(dict_of_lists['a'])
 ['something for a']
```