PYTHON ASSIGNMENT 18

1. Create a zoo.py file first. Define the hours() function, which prints the string &#39;Open 9-5 daily&#39;.

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 &#39;a&#39;: 1, &#39;b&#39;: 2, and &#39;c&#39;: 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[&#39;a&#39;] and append the value &#39;something for a&#39; to it in one assignment. Print

dict_of_lists[&#39;a&#39;].

SOLUTIONS

```
# 1. Create zoo.py and define the hours() function

# zoo.py

def hours():

    print('Open 9-5 daily')
```

```
# 2. Import zoo module and call its hours() function
import zoo
zoo.hours()

# 3. Explicitly import and call the hours() function from zoo
from zoo import hours
hours()

# 4. Import the hours() function as info and call it
from zoo import hours as info
info()

# 5. Create a plain dictionary and print it
plain_dict = {'a': 1, 'b': 2, 'c': 3}
print(plain_dict)

# 6. Make an OrderedDict called fancy and print it
from collections import OrderedDict
fancy_dict = OrderedDict([('a', 1), ('b', 2), ('c', 3)])
print(fancy_dict)

# 7. Make a default dictionary called dict_of_lists
from collections import defaultdict
dict_of_lists = defaultdict(list)

# Append the value 'something for a' to dict_of_lists['a'] in one assignment
dict_of_lists['a'].append('something for a')
```

```
print(dict_of_lists['a'])
```

Note: Save the **zoo.py** file with the **hours()** function definition before running the code. The use of **OrderedDict** in 6 ensures that the order of items is maintained, unlike a regular dictionary.