Name :- Deepawali . B. Mhaisagar
Assignment no 18

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.

ANS

To create the `zoo.py` file, you can follow these steps:

Step 1: Create a new file named `zoo.py` and open it in a text editor

Step 2: Define the `hours()` function in the `zoo.py` file with the desired code. In this case, the function will print the string "Open 9-5 daily". Here's the code for `zoo.py`:

```
def hours():

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

Step 3: Save the `zoo.py` file

Now, to use the interactive interpreter to import the `zoo` module and call its `hours()` function, follow these steps:

Step 1: Open the Python interactive interpreter by running the `python` command in your terminal or command prompt.

Step 2: Import the `zoo` module by typing the following command:

```
import zoo
```

Step 3: Call the `hours()` function from the `zoo` module using the following command:

```
zoo.hours()
```

The output will be:```

```
Open 9-5 daily
```

By following these steps, you have created the `zoo.py` file and successfully imported and called the `hours()` function from the `zoo` module using the Python interactive interpreter.

2. In the interactive interpreter, import the zoo module as menagerie and call its hours() function.

ANS

To import the `zoo` module as `menagerie` in the interactive interpreter and call its `hours()` function, follow these steps:

Step 1: Open the Python interactive interpreter by running the `python` command in your terminal or command prompt

Step 2: Import the `zoo` module as `menagerie` using the `import` statement and assigning it the alias `menagerie`

```
import zoo as menagerie
```

Step 3: Call the `hours()` function from the `menagerie` module using the following command:

```
menagerie.hours()
```

The output will be:

```
Open 9-5 daily
```

By following these steps, you have successfully imported the `zoo` module as `menagerie` and called its `hours()` function in the Python interactive interpreter.

3. Using the interpreter, explicitly import and call the hours() function from zoo.

ANS

To print the values of the list `[3, 2, 1, 0]` using a for loop, you can iterate over the elements of the list and print each element. Here's an example:

```
my_list = [3, 2, 1, 0]



for num in my_list:

    print(num)
```

After executing this code, the output will be:

```
3

2

1

0
```

The for loop iterates over each element in the list `my_list` and assigns it to the variable `num`. The `print(num)` statement then prints each element on a separate line.

4. Import the hours() function as info and call it.

ANS

To import the `hours()` function as `info` and call it, you can follow these steps:

Step 1: Create a `zoo.py` file if it doesn't exist already.

Step 2: Define the `hours()` function inside `zoo.py`:

```
def hours():

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

Step 3: Save the `zoo.py` file.

Step 4: Open the Python interactive interpreter by running the `python` command in your terminal or command prompt.

Step 5: Import the `hours()` function as `info` from the `zoo` module using the `import...as` statement:

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

Step 6: Call the `info()` function:

```
info()
```

The output will be:

```
Open 9-5 daily
```

By following these steps, you have imported the `hours()` function from the `zoo` module as `info` and called it in the Python interactive interpreter.

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.

ANS

To create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and print it out, you can follow these steps:

Step 1: Define the dictionary with the desired key-value pairs:

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

Step 2: Print out the dictionary using the `print()` function:

```
print(my_dict)
```

The output will be:

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

By following these steps, you have created a plain dictionary with the specified key-value pairs and printed 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?

ANS

To create an OrderedDict called `fancy` with the same key-value pairs as in question 5 and print it, you can follow these steps:

Step 1: Import the `OrderedDict` class from the `collections` module:

```
from collections import OrderedDict
```

Step 2: Create an OrderedDict called `fancy` with the desired key-value pairs:

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

Step 3: Print out the `fancy` OrderedDict using the `print()` function:

```
print(fancy)
```

The output will be:

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

In an OrderedDict, the order of insertion is preserved, so the key-value pairs are printed in the same order as they were inserted.

By following these steps, you have created an OrderedDict called `fancy` with the same key-value pairs as in question 5 and printed it. The order of the key-value pairs is maintained in the printed output.

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 ‘ to it in one assignment.
Print
dict_of_lists[‘a’].

ANS

To create a defaultdict called `dict_of_lists` and pass it the argument `list`, then append the value 'something for a' to `dict_of_lists['a']` in one assignment, and finally print `dict_of_lists['a']`, you can follow these steps:

Step 1: Import the `defaultdict` class from the `collections` module:

```python
from collections import defaultdict
```

Step 2: Create a defaultdict called `dict_of_lists` with the argument `list`:

```python
dict_of_lists = defaultdict(list)
```

Step 3: Append the value 'something for a' to `dict_of_lists['a']` in one assignment:

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

Step 4: Print `dict_of_lists['a']`:

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

The output will be:

```
['something for a']
```
By following these steps, you have created a defaultdict called `dict_of_lists`, assigned the value 'something for a' to `dict_of_lists['a']` in one assignment, and printed `dict_of_lists['a']`.