**Assignment 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.**

**Sure! Here's the code for the zoo.py file:**

**python**

**Copy code**

**# zoo.py**

**def hours():**

**print("Open 9-5 daily")**

**And here's how you can import the zoo module and call its hours() function in the interactive interpreter:**

**python**

**Copy code**

**>>> import zoo**

**>>> zoo.hours()**

**Open 9-5 daily**

**This will import the zoo module and call its hours() function, which will print the string "Open 9-5 daily" to the console.**

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

**Sure! Here's how you can import the zoo module as menagerie and call its hours() function in the interactive interpreter:**

**python**

**Copy code**

**>>> import zoo as menagerie**

**>>> menagerie.hours()**

**Open 9-5 daily**

**This will import the zoo module as menagerie and call its hours() function, which will print the string "Open 9-5 daily" to the console.**

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

**Sure! Here's how you can explicitly import and call the hours() function from the zoo module using the interactive interpreter:**

**python**

**Copy code**

**>>> from zoo import hours**

**>>> hours()**

**Open 9-5 daily**

**This will import the hours() function directly from the zoo module and call it, which will print the string "Open 9-5 daily" to the console.**

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

**Sure! Here's how you can import the hours() function from the zoo module as info and call it using the interactive interpreter:**

**python**

**Copy code**

**>>> from zoo import hours as info**

**>>> info()**

**Open 9-5 daily**

**This will import the hours() function from the zoo module as info and call it, which will print the string "Open 9-5 daily" to the console.**

**5. Create a plain dictionary with the key-value pairs ‘a’: 1, ‘b’: 2, and ‘c’: 3, and print it out.**

**Sure! Here's the code to create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and print it out:**

**python**

**Copy code**

**my\_dict = {'a': 1, 'b': 2, 'c': 3}**

**print(my\_dict)**

**This will create the dictionary with the specified key-value pairs and print it out to the console:**

**css**

**Copy code**

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

**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?**

**Sure! Here's the code to create an OrderedDict called fancy from the same key-value pairs listed in 5 and print it:**

**python**

**Copy code**

**from collections import OrderedDict**

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

**print(fancy)**

**This will create an OrderedDict with the specified key-value pairs and print it out to the console:**

**css**

**Copy code**

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

**Yes, it printed in the same order as plain because the OrderedDict remembers the order in which its keys were first inserted. Since the key-value pairs were inserted in the same order as they appear in the plain dictionary, they are printed in the same order as well.**

**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’].**

**Sure! Here's the code to create a defaultdict called dict\_of\_lists and append the value 'something for a' to its list associated with the key 'a' in one assignment:**

**python**

**Copy code**

**from collections import defaultdict**

**dict\_of\_lists = defaultdict(list)**

**dict\_of\_lists['a'].append('something for a')**

**print(dict\_of\_lists['a'])**

**This will create a defaultdict with the default value type of a list, and then append the string 'something for a' to the list associated with the key 'a' in one assignment. Finally, it will print the list associated with the key 'a':**

**css**

**Copy code**

**['something for a']**

**Since the key 'a' did not previously exist in the defaultdict, it creates a new empty list associated with the key 'a' and then appends the value 'something for a' to it.**