**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.**

Here is an example of how you could create the zoo.py file:

def hours():

print("Open 9-5 daily")

And then you can use the Python interactive interpreter to import the zoo module and call its hours() function like this:

>>> import zoo

>>> zoo.hours()

Open 9-5 daily

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

You can import the zoo module as menagerie and call its hours() function in the Python interactive interpreter like this:

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

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

You can explicitly import the hours() function from the zoo module in the Python interactive interpreter like this:

>>> from zoo import hours

>>> hours()

Open 9-5 daily

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

You can import the hours() function from the zoo module as info in the Python interactive interpreter like this:

>>> from zoo import hours as info

>>> info()

Open 9-5 daily

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

You can create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3 in Python like this:

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

>>> print(my\_dict)

{'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?**

You can make an OrderedDict called fancy from the same key-value pairs listed in the previous question using the collections module in Python like this:

>>> from collections import OrderedDict

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

>>> print(fancy)

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

Yes, it will print in the same order as the plain dictionary because an OrderedDict is a dictionary that remembers the order in which items were inserted.

**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'].**

You can make a default dictionary called dict\_of\_lists and pass it the argument list using the collections module in Python like this:

>>> from collections import defaultdict

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

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

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

['something for a']

The defaultdict class takes a default factory as an argument (in this case, list), which is used to create a new value when a key that doesn't exist in the dictionary is accessed. So when you try to access dict\_of\_lists['a'], the list factory is called to create a new list, and then you append the value 'something for a' to it.