**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.**

def hours():

print('Open 9-5 daily')

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

import zoo

zoo.hours()

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

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

from mymodule import myfunc as my\_alias

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

Here's how you can create and print a plain dictionary with the given key-value pairs:

```python

# create the dictionary

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

# print the dictionary

print(my\_dict)

```

Output:

```

{'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?**

Yes, an OrderedDict maintains the order of the keys as they were inserted. Here's an example code to create an OrderedDict called fancy and print it:

```python

from collections import OrderedDict

# create the plain dictionary

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

# create the ordered dictionary

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

print(plain)

print(fancy)

```

This will output:

```

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

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

```

As you can see, the order of the items in the `fancy` OrderedDict is the same as the order in which they were added.

**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'].**

Here's how you can create a `defaultdict` and append a value to its list associated with a key:

```python

from collections import defaultdict

dict\_of\_lists = defaultdict(list) # Creating the defaultdict with default values as empty list

dict\_of\_lists['a'].append('something for a') # Appending a value to the list associated with the key 'a'

print(dict\_of\_lists['a']) # Printing the value(s) associated with key 'a'

```

Output:

```

['something for a']

```