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.

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 'a': 1, 'b': 2, and 'c': 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['a'] and append the value 'something for a' to it in one assignment. Print dict\_of\_lists['a'].

Ans:

1. Here's the code for the `zoo.py` file:

```python

# zoo.py

def hours():

print('Open 9-5 daily')

```

To use the interactive interpreter, follow these steps:

```python

>>> import zoo

>>> zoo.hours()

Open 9-5 daily

```

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

```python

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

```

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

```python

>>> from zoo import hours

>>> hours()

Open 9-5 daily

```

4. You can import the `hours()` function as `info` and call it:

```python

>>> from zoo import hours as info

>>> info()

Open 9-5 daily

```

5. To create a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and print it, you can use the following code:

```python

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

>>> print(plain\_dict)

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

```

6. To create an `OrderedDict` called `fancy` from the same key-value pairs and print it, you can use the following code:

```python

>>> from collections import OrderedDict

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

>>> print(fancy)

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

```

Yes, the `OrderedDict` will print the key-value pairs in the same order as they were inserted.

7. To create a `defaultdict` called `dict\_of\_lists` and append the value 'something for a' to `dict\_of\_lists['a']` in one assignment, you can use the following code:

```python

>>> 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` automatically creates an empty list as the default value for any non-existing key, so we can directly append the value to it without explicitly creating a list first.