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.

Creating zoo.py file with the `hours()` function:

```python

# zoo.py

def hours():

print('Open 9-5 daily')

```

Using the interactive interpreter to import the zoo module and call its `hours()` function:

```python

>>> import zoo

>>> zoo.hours()

Open 9-5 daily

```

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

Importing the `zoo` module as `menagerie` and calling its `hours()` function:

```python

>>> import zoo as menagerie

>>> menagerie.hours()

Open 9-5 daily

```

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

Explicitly importing and calling the `hours()` function from `zoo`:

```python

>>> from zoo import hours

>>> hours()

Open 9-5 daily

```

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

Importing the `hours()` function as `info` and calling it:

```python

>>> 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.

Creating a plain dictionary with the key-value pairs 'a': 1, 'b': 2, and 'c': 3, and printing it out:

```python

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

>>> print(plain)

{'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?

Making an `OrderedDict` called `fancy` from the same pairs listed in 5 and printing it. It should print in the same order as plain.

```python

from collections import OrderedDict

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

>>> print(fancy)

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

```

Yes, it printed 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'].

Making a default dictionary called `dict\_of\_lists` and passing it the argument `list`. Making the list `dict\_of\_lists['a']` and appending the value 'something for a' to it in one assignment. Printing `dict\_of\_lists['a']`.

```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']

```