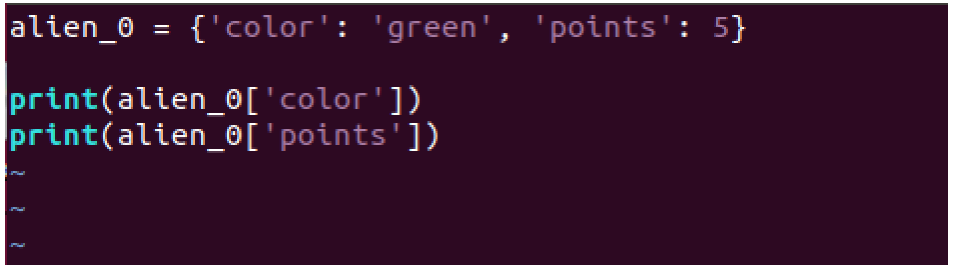
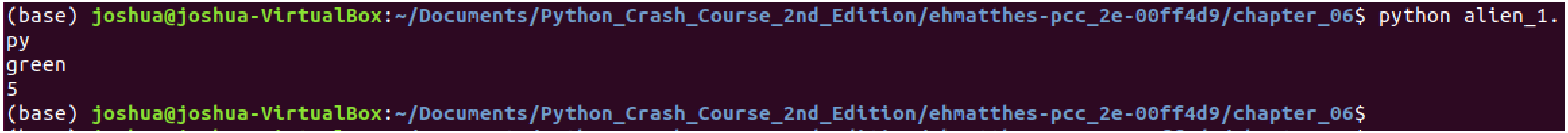
Dictionaries

A Simple Dictionary



```
alien_0 = {'color': 'green', 'points': 5}

print(alien_0['color'])
print(alien_0['points'])
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_1.
py
green
5
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Working with Dictionaries

- A *dictionary* in Python is a collection of *key-value* pairs.
- Each *key* is connected to a value, and you can use a key to access the value associated with that key.
- A key's value can be a number, a string, a list, or even another dictionary.
- In fact, you can use any object that you can create in Python as a value in a dictionary.

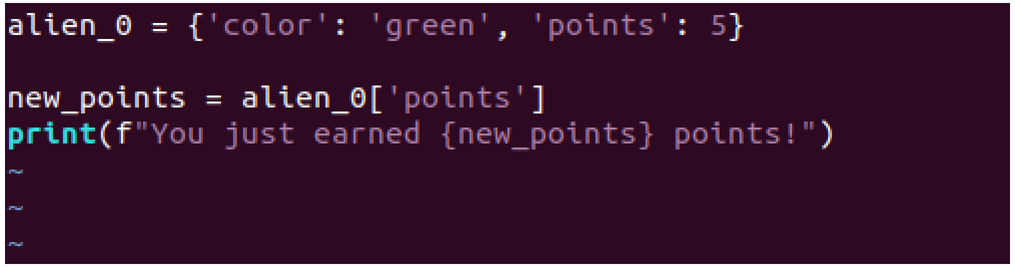
- In Python, a dictionary is wrapped in braces, {}, with a series of key-value pairs inside the braces, as shown in the earlier example:

  ```
  alien_0 = {'color': 'green', 'points': 5}
  ```

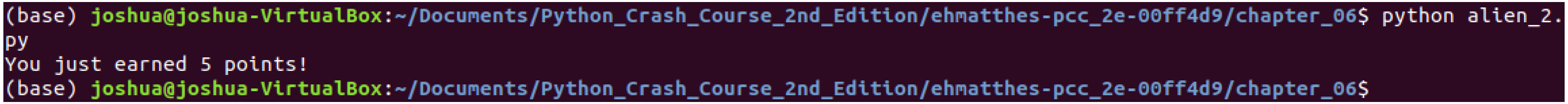
- You can store as many key-value pairs as you want in a dictionary.

Accessing Values in a Dictionary

- To get the value associated with a key, give the name of the dictionary and then place the key inside a set of square brackets.



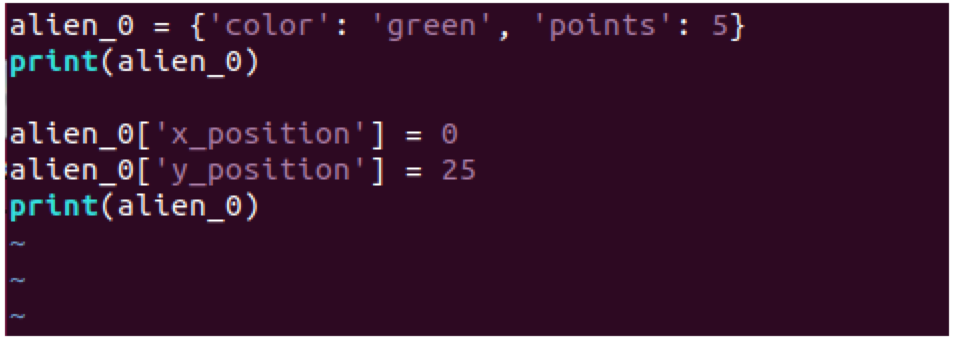
```
alien_0 = {'color': 'green', 'points': 5}

new_points = alien_0['points']
print(f"You just earned {new_points} points!")
~
~
~
```



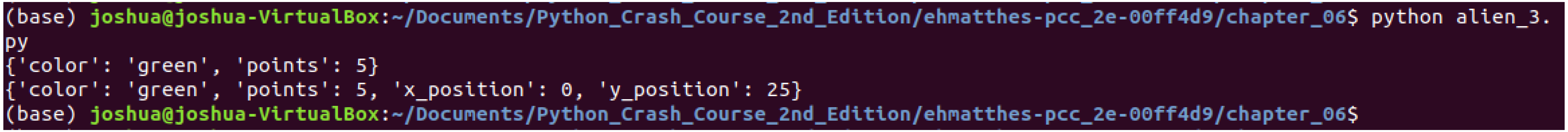
```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_2.
py
You just earned 5 points!
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Adding New Key-Value Pairs

- Dictionaries are dynamic structures, and you can add new key-value pairs to a dictionary at any time.
- For example, to add a new key-value pair, you would give the name of the dictionary followed by the new key in square brackets along with the new value.

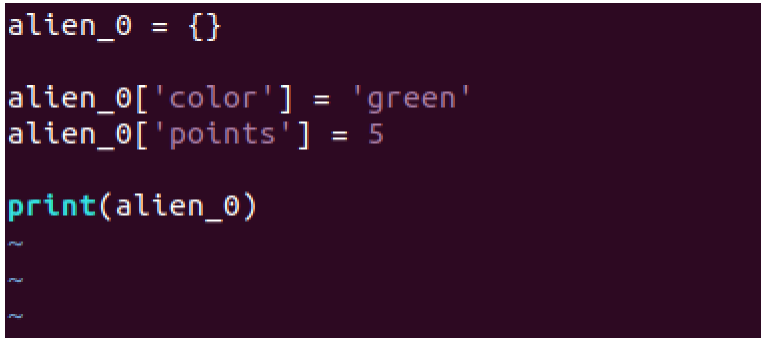


```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)

alien_0['x_position'] = 0
alien_0['y_position'] = 25
print(alien_0)
~
~
~
```

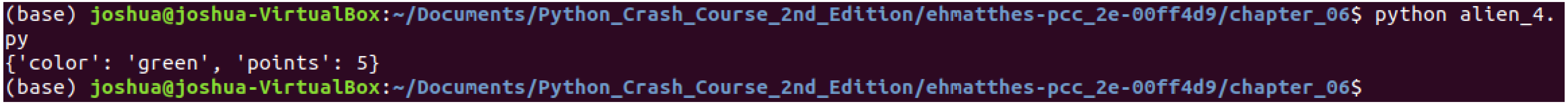


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_3.
py
{'color': 'green', 'points': 5}
{'color': 'green', 'points': 5, 'x_position': 0, 'y_position': 25}
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Starting with an Empty Dictionary



```
alien_0 = {}

alien_0['color'] = 'green'
alien_0['points'] = 5

print(alien_0)
~
~
~
```

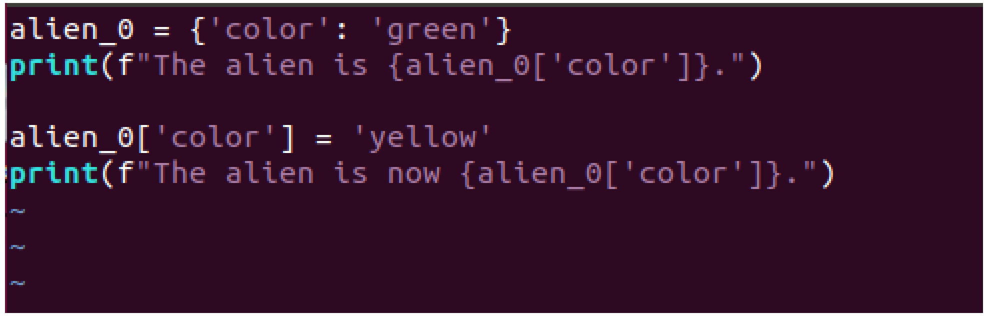


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_4.
py
{'color': 'green', 'points': 5}
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

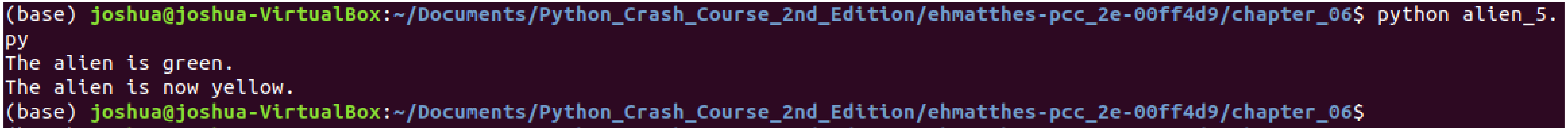
- Typically, you’ll use empty dictionaries when storing user-supplied data in a dictionary or when you write code that generates a large number of key-value pairs automatically.

Modifying Values in a Dictionary

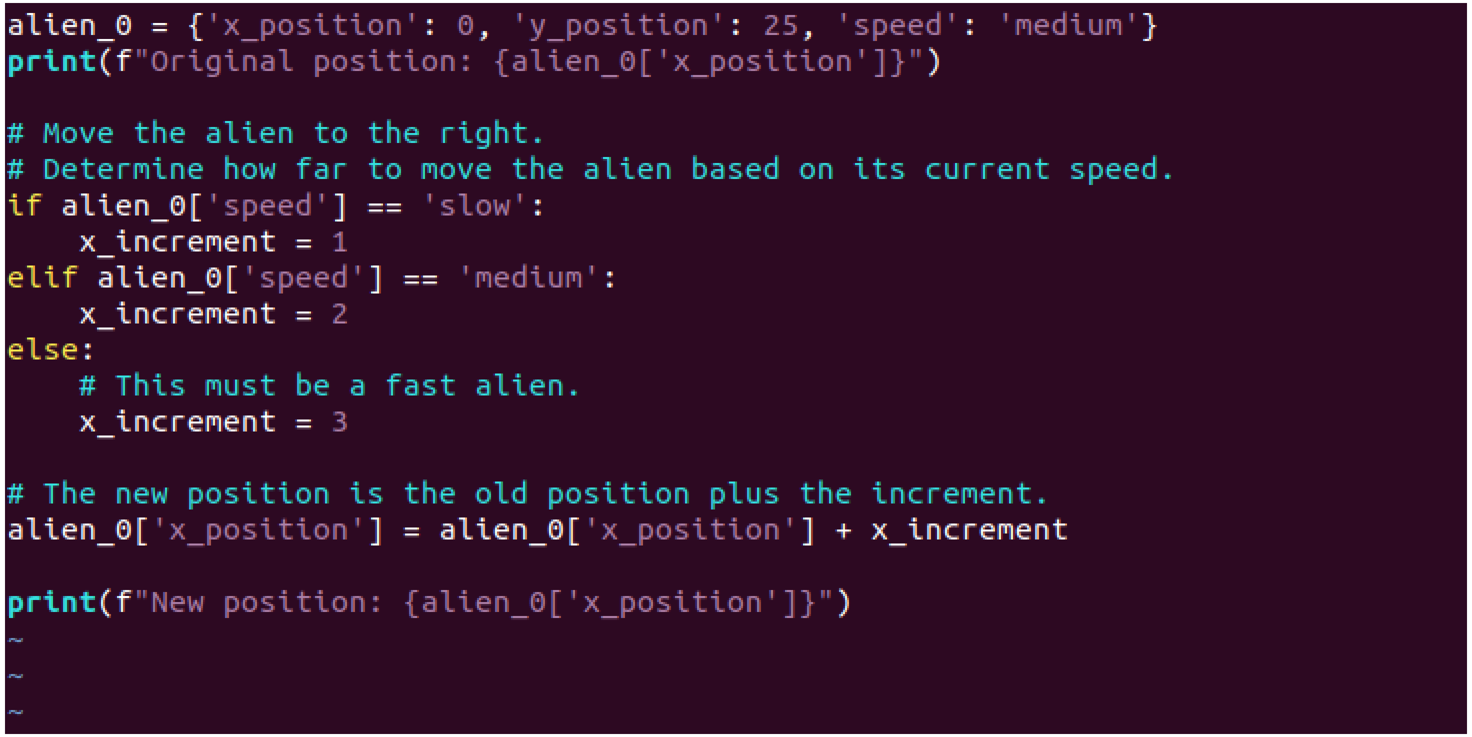
- To modify a value in a dictionary, give the name of the dictionary with the key in square brackets and then the new value you want associated with that key.



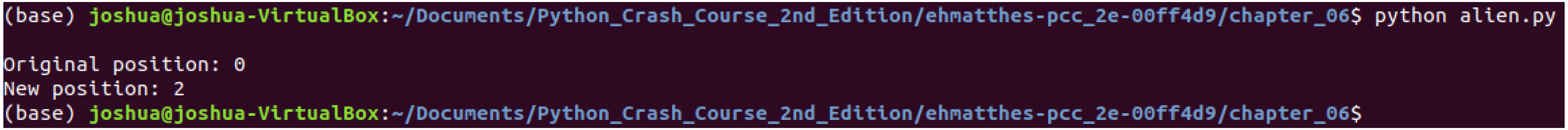
```
alien_0 = {'color': 'green'}
print(f"The alien is {alien_0['color']}.")

alien_0['color'] = 'yellow'
print(f"The alien is now {alien_0['color']}.")
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_5.
py
The alien is green.
The alien is now yellow.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```



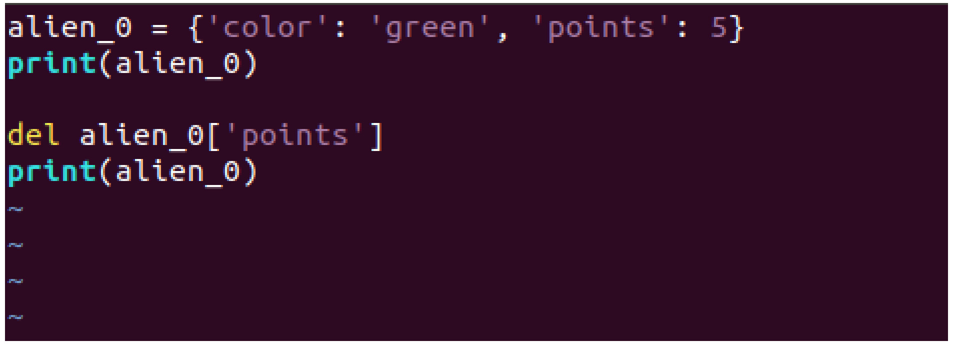
```
alien_0 = {'x_position': 0, 'y_position': 25, 'speed': 'medium'}
print(f"Original position: {alien_0['x_position']}")

# Move the alien to the right.
# Determine how far to move the alien based on its current speed.
if alien_0['speed'] == 'slow':
    x_increment = 1
elif alien_0['speed'] == 'medium':
    x_increment = 2
else:
    # This must be a fast alien.
    x_increment = 3

# The new position is the old position plus the increment.
alien_0['x_position'] = alien_0['x_position'] + x_increment

print(f"New position: {alien_0['x_position']}")
~
~
~
```



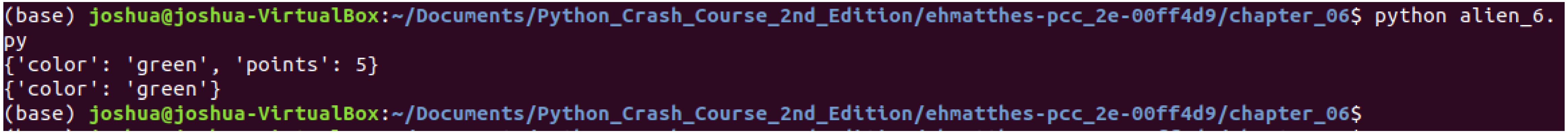
```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien.py

Original position: 0
New position: 2
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Removing Key-Value Pairs

- When you no longer need a piece of information that’s stored in a dictionary, you can use the `del` statement to completely remove a key-value pair.

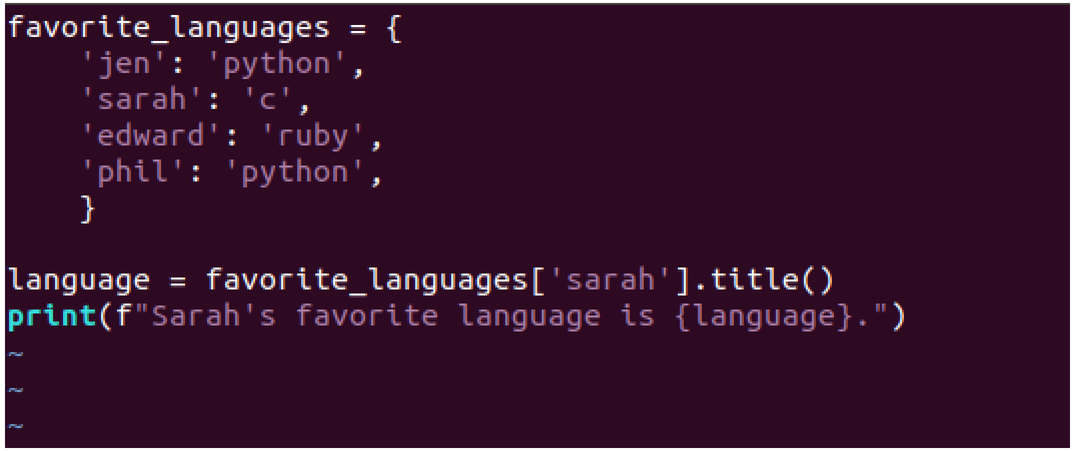


```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)

del alien_0['points']
print(alien_0)
~
~
~
~
```

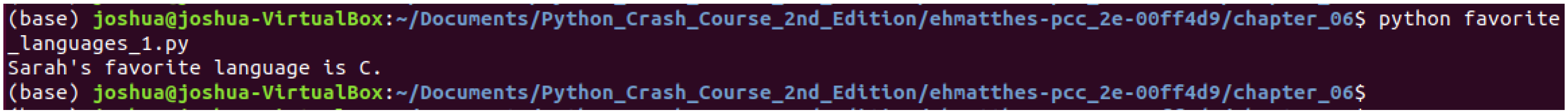


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python alien_6.
py
{'color': 'green', 'points': 5}
{'color': 'green'}
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

A Dictionary of Similar Objects



```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

language = favorite_languages['sarah'].title()
print(f"Sarah's favorite language is {language}.")
~
~
~
```

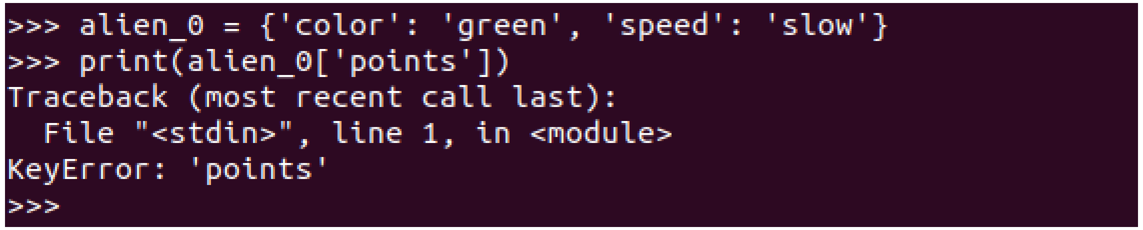


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_1.py
Sarah's favorite language is C.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

- It’s good practice to include a comma after the last key-value pair as well, so you’re ready to add a new key-value pair on the next line.

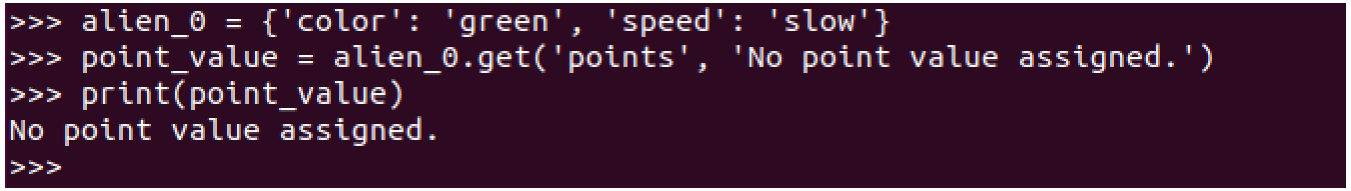
Using `get()` to Access Values

- Using keys in square brackets to retrieve the value you’re interested in from a dictionary might cause one potential problem: if the key you ask for doesn’t exist, you’ll get an error.



```
>>> alien_0 = {'color': 'green', 'speed': 'slow'}
>>> print(alien_0['points'])
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'points'
>>>
```

- The `get()` method requires a key as a first argument.
- As a second optional argument, you can pass the value to be returned if the key doesn’t exist:

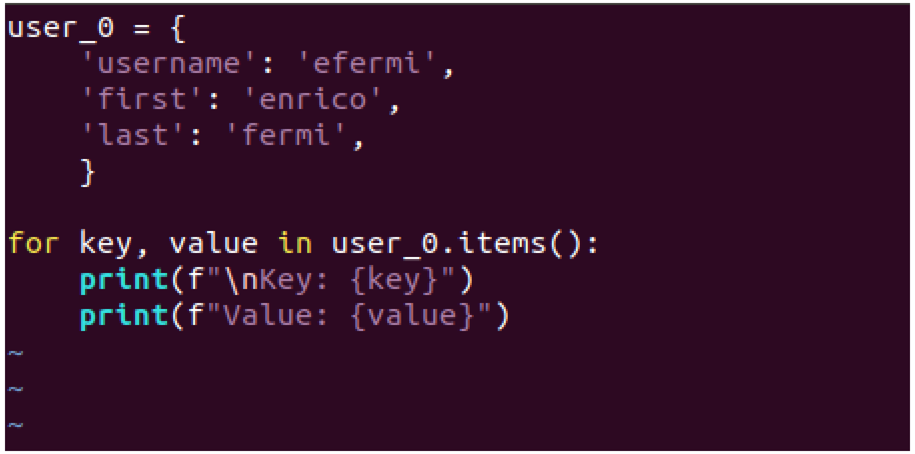


```
>>> alien_0 = {'color': 'green', 'speed': 'slow'}
>>> point_value = alien_0.get('points', 'No point value assigned.')
>>> print(point_value)
No point value assigned.
>>>
```

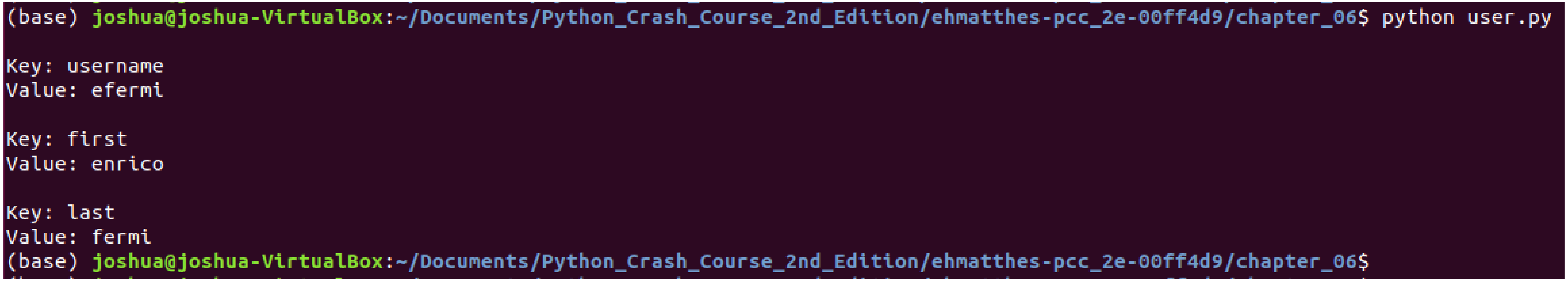
- If you leave out the second argument in the call to `get()` and the key doesn’t exist, Python will return the value `None`.
- The special value `None` means “no value exists.”

Looping Through a Dictionary

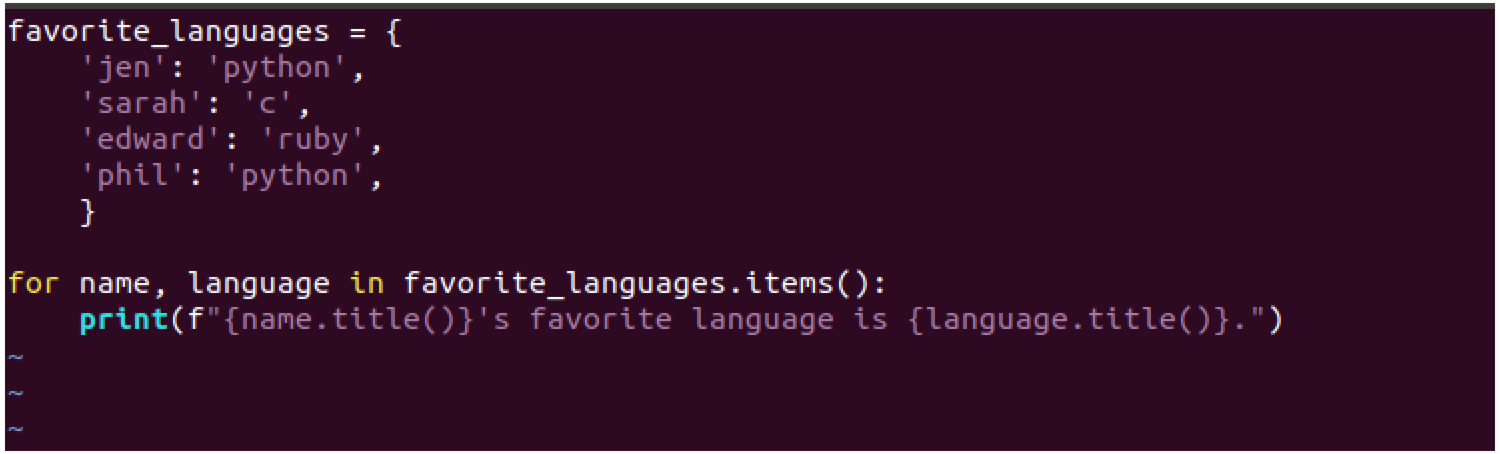
Looping Through All Key-Value Pairs



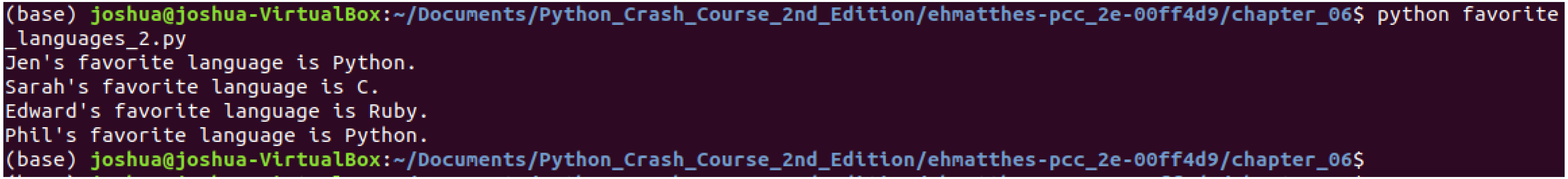
```
user_0 = {
    'username': 'efermi',
    'first': 'enrico',
    'last': 'fermi',
    }

for key, value in user_0.items():
    print(f"\nKey: {key}")
    print(f"Value: {value}")
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python user.py

Key: username
Value: efermi

Key: first
Value: enrico

Key: last
Value: fermi
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

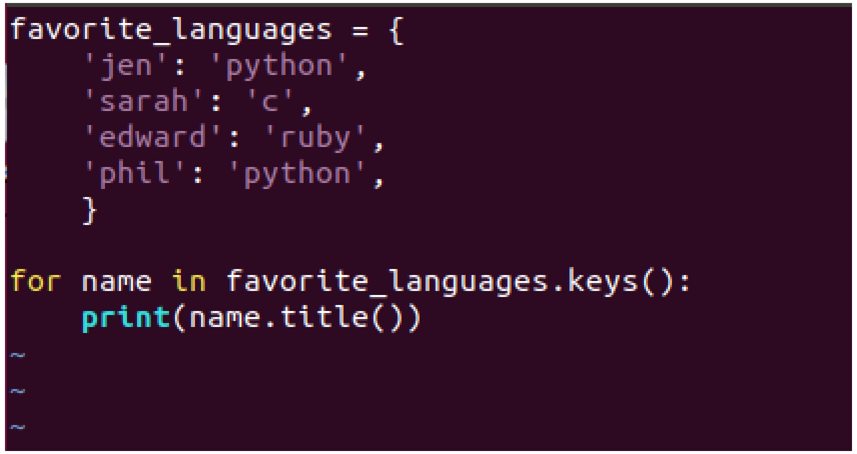


```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

for name, language in favorite_languages.items():
    print(f"{name.title()}'s favorite language is {language.title()}.")
~
~
~
```

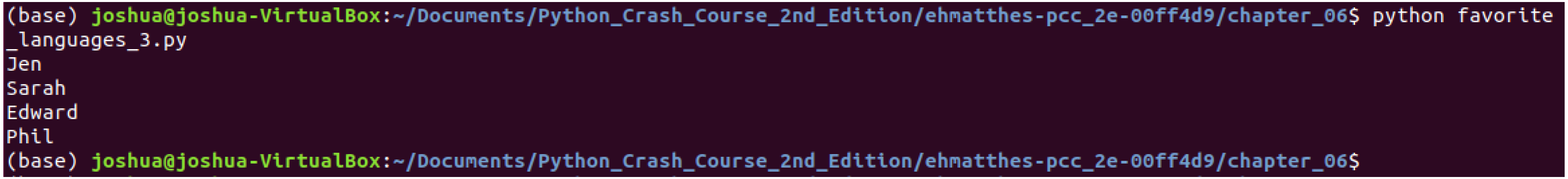


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_2.py
Jen's favorite language is Python.
Sarah's favorite language is C.
Edward's favorite language is Ruby.
Phil's favorite language is Python.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Looping Through All the Keys in a Dictionary

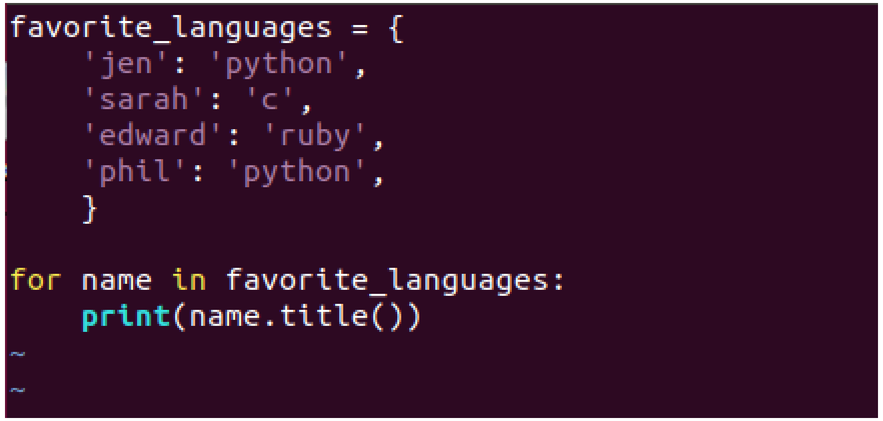


```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

for name in favorite_languages.keys():
    print(name.title())
~
~
~
```

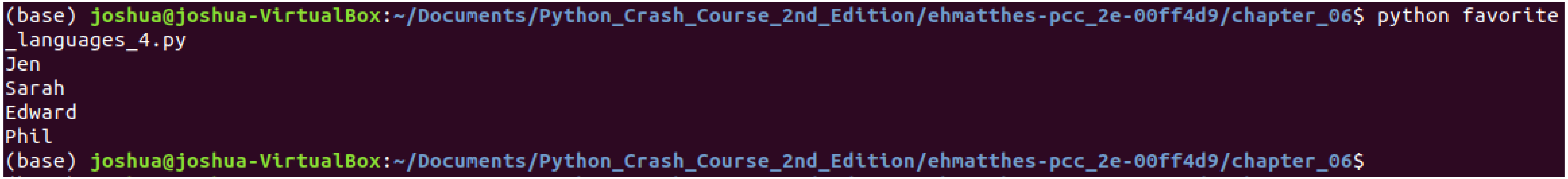


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_3.py
Jen
Sarah
Edward
Phil
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

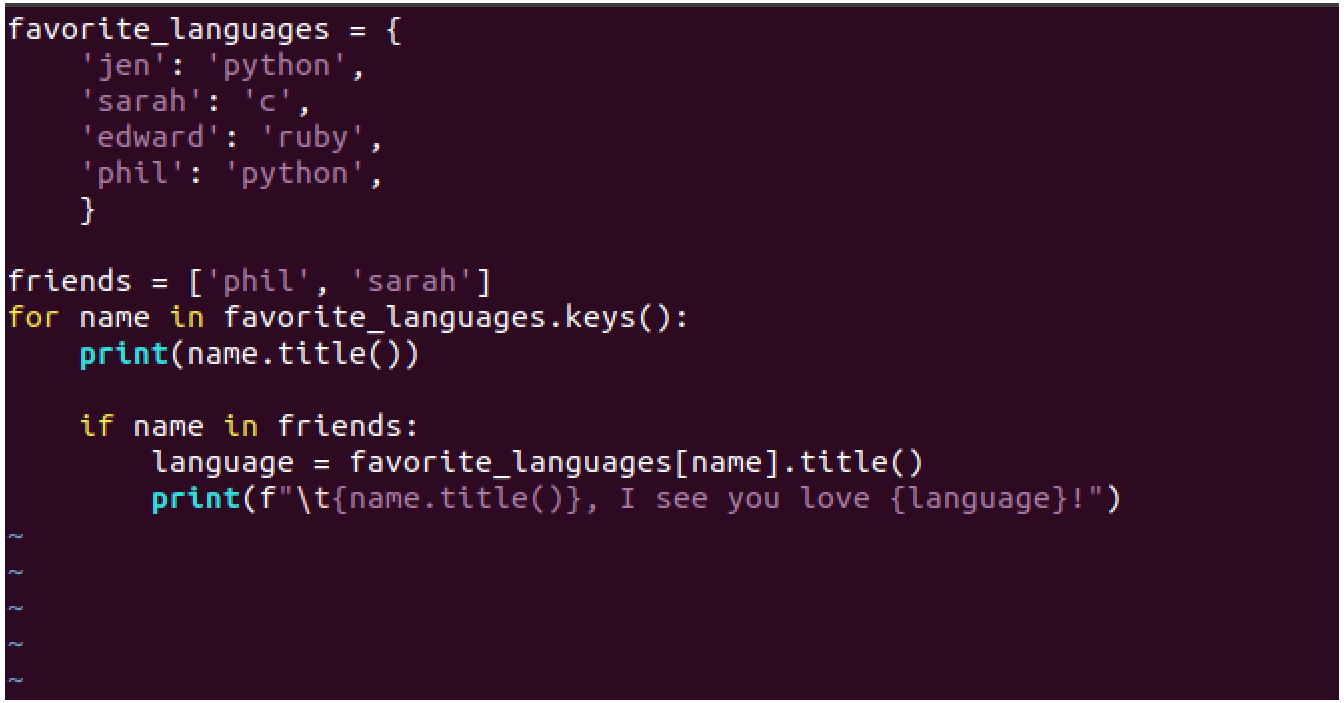
- Looping through the keys is actually the default behavior when looping through a dictionary.



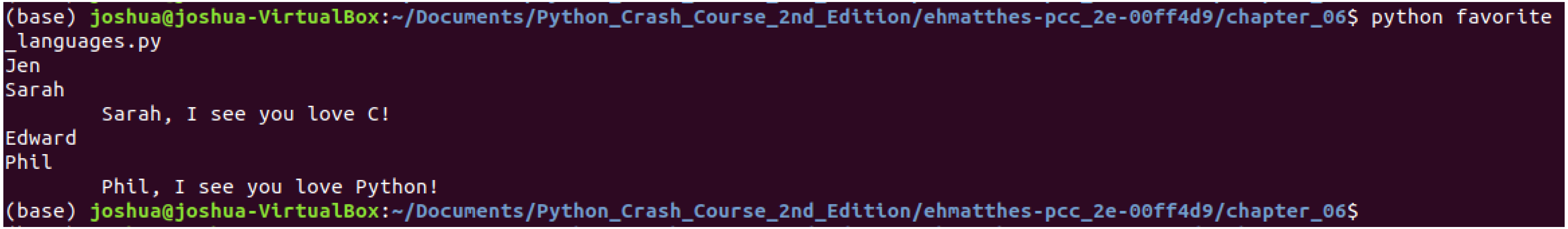
```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

for name in favorite_languages:
    print(name.title())
~
~
```



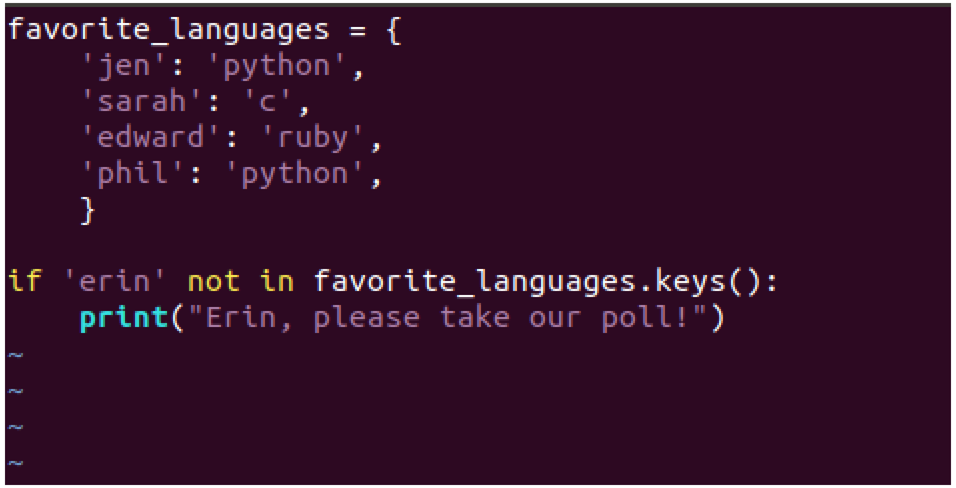
```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_4.py
Jen
Sarah
Edward
Phil
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```



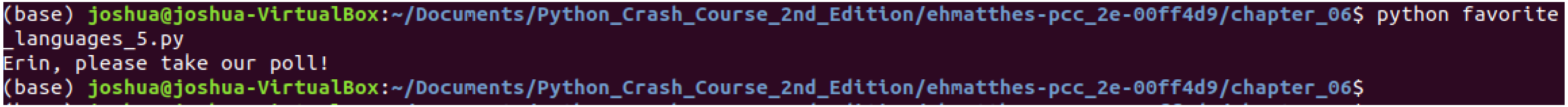
```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

friends = ['phil', 'sarah']
for name in favorite_languages.keys():
    print(name.title())

    if name in friends:
        language = favorite_languages[name].title()
        print(f"\t{name.title()}, I see you love {language}!")
~
~
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages.py
Jen
Sarah
        Sarah, I see you love C!
Edward
Phil
        Phil, I see you love Python!
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```



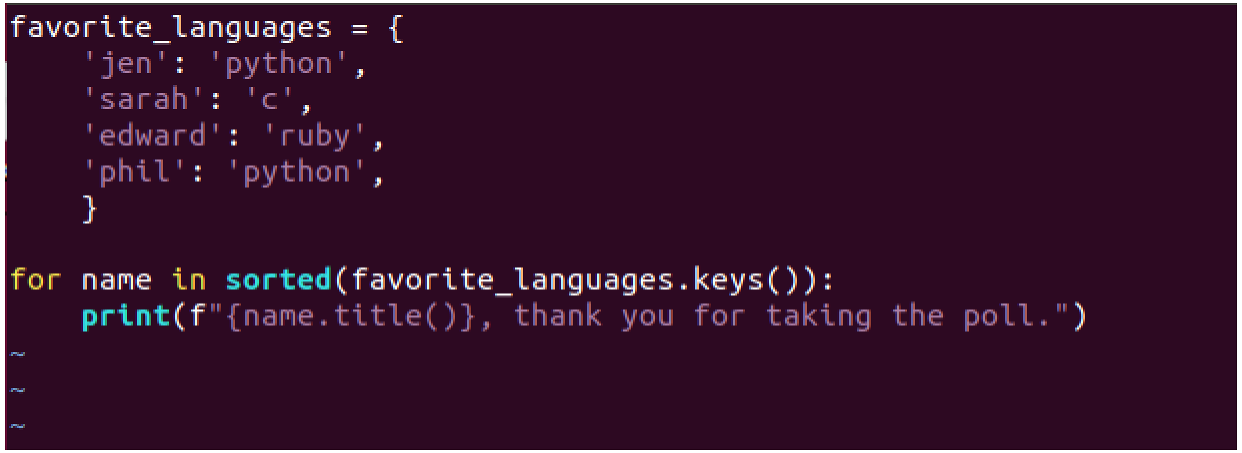
```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

if 'erin' not in favorite_languages.keys():
    print("Erin, please take our poll!")
~
~
~
~
```



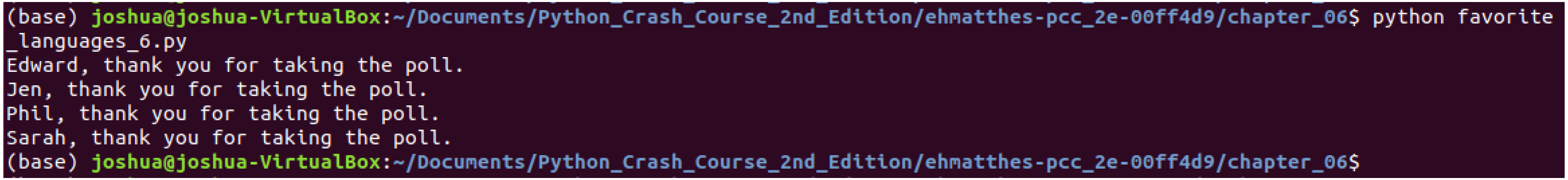
```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_5.py
Erin, please take our poll!
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Looping Through a Dictionary’s Keys in a Particular Order

- Starting in Python 3.7, looping through a dictionary returns the items in the same order they were inserted.
- Sometimes, though, you’ll want to loop through a dictionary in a different order.

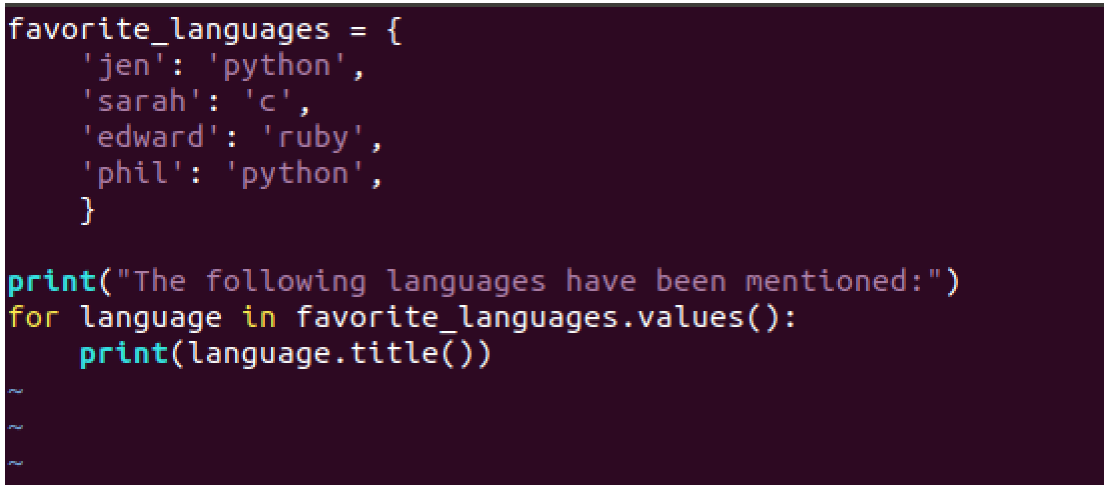


```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

for name in sorted(favorite_languages.keys()):
    print(f"{name.title()}, thank you for taking the poll.")
~
~
~
```

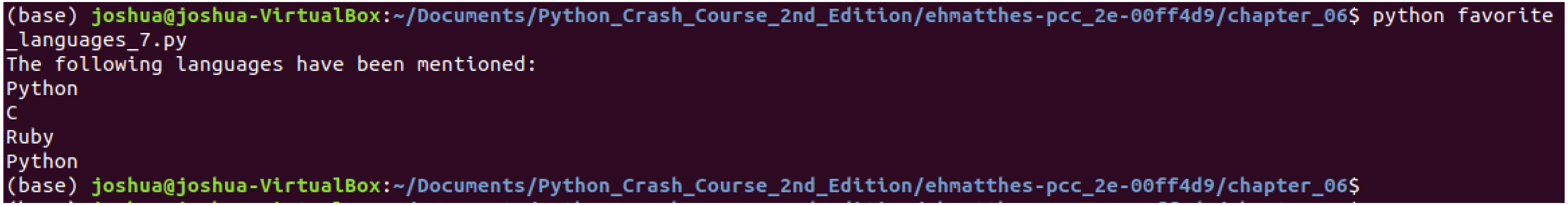


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_6.py
Edward, thank you for taking the poll.
Jen, thank you for taking the poll.
Phil, thank you for taking the poll.
Sarah, thank you for taking the poll.
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Looping Through All Values in a Dictionary

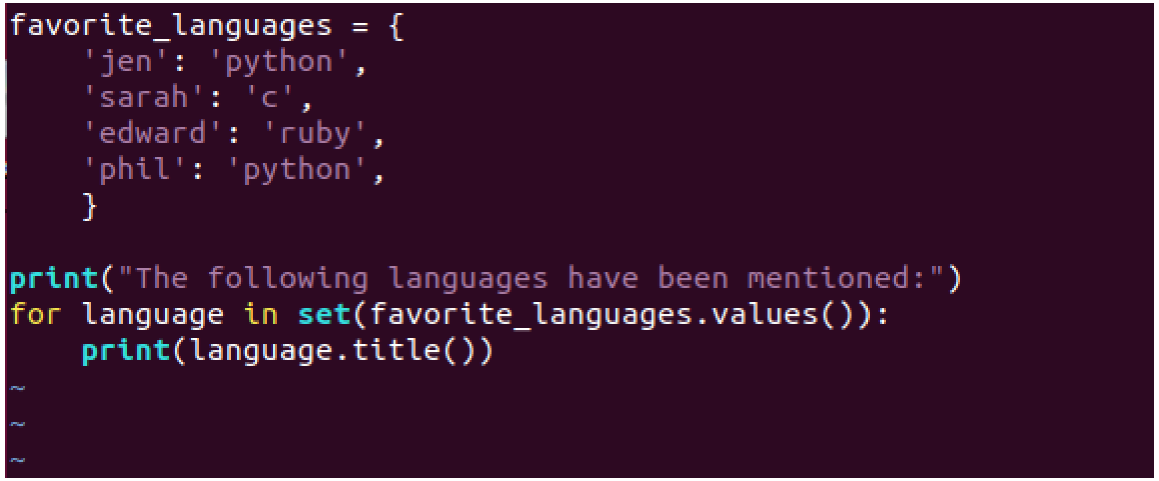


```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

print("The following languages have been mentioned:")
for language in favorite_languages.values():
    print(language.title())
~
~
~
```

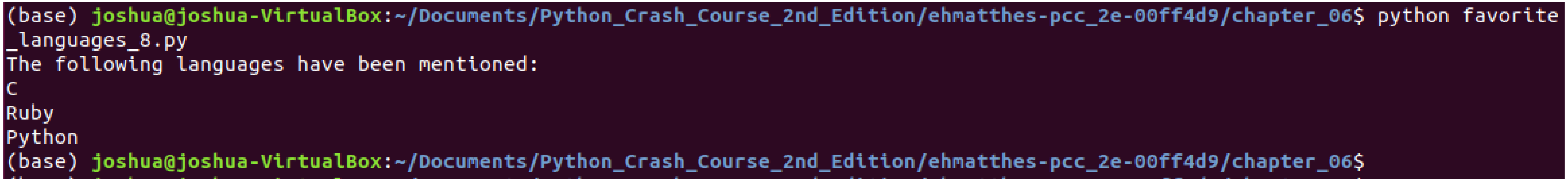


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_7.py
The following languages have been mentioned:
Python
C
Ruby
Python
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

- To see each language chosen without repetition, we can use a set.
- A `set` is a collection in which each item must be unique.



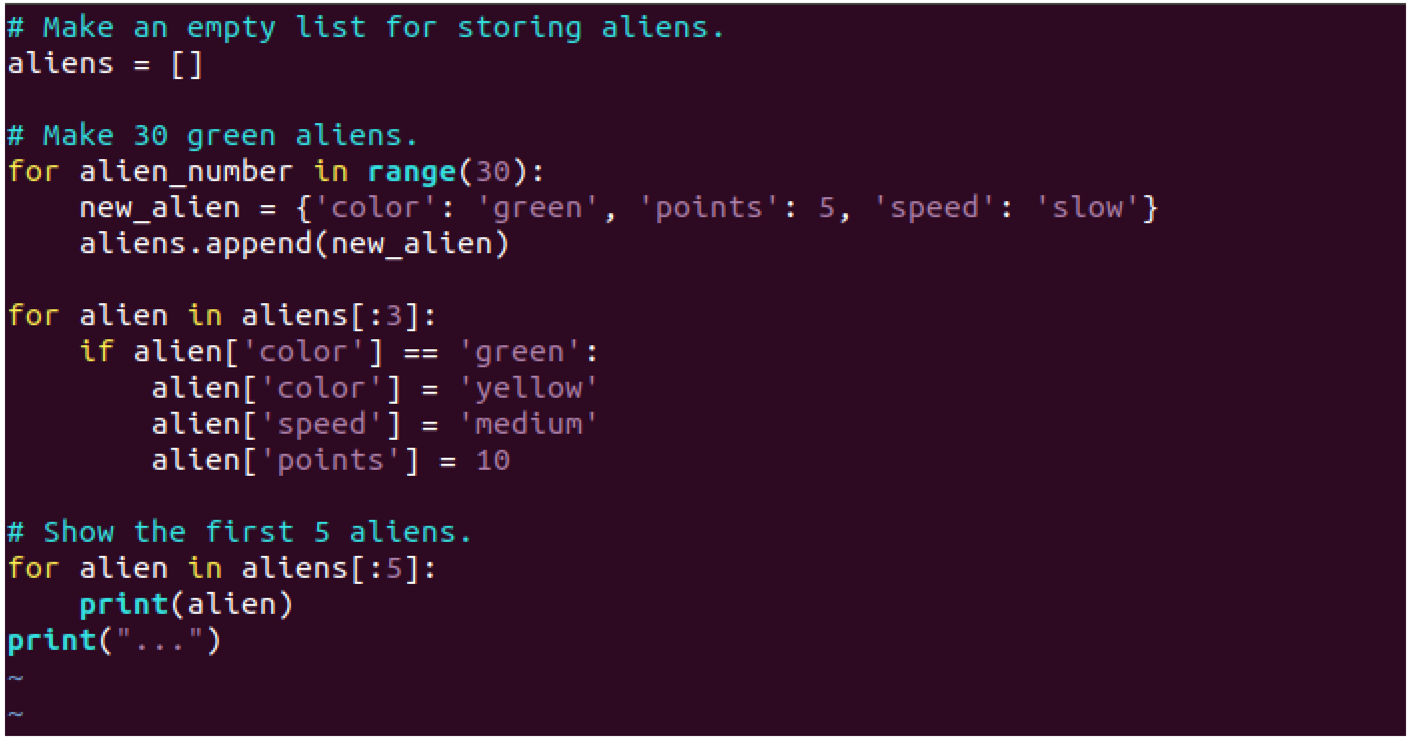
```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
    }

print("The following languages have been mentioned:")
for language in set(favorite_languages.values()):
    print(language.title())
~
~
~
```



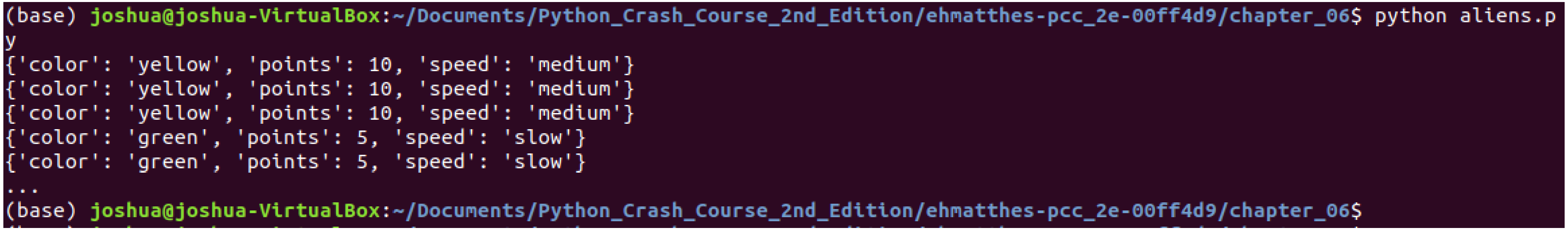
```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_8.py
The following languages have been mentioned:
C
Ruby
Python
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

Nesting

A List of Dictionaries

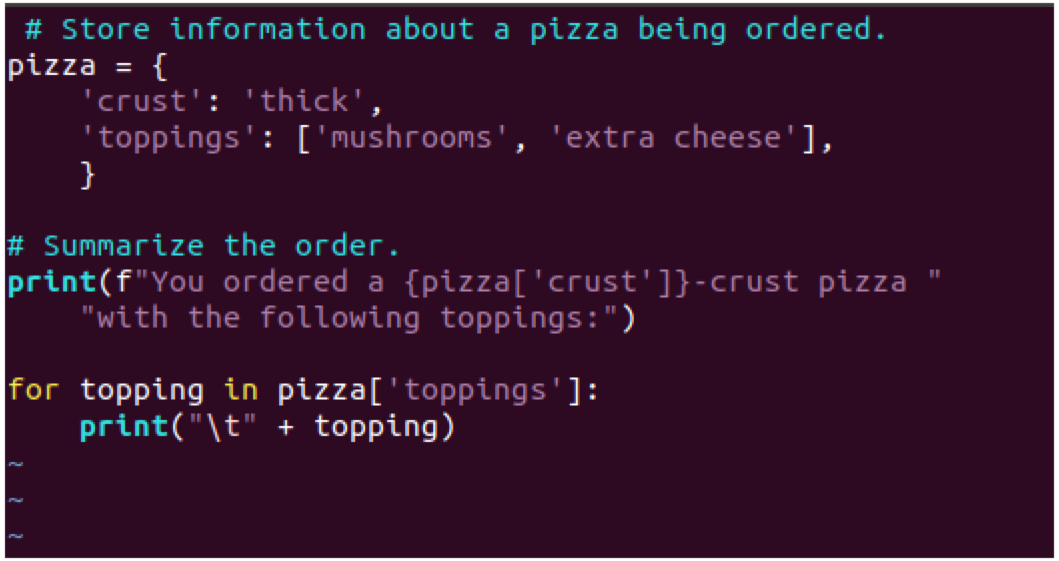


```
# Make an empty list for storing aliens.
aliens = []

# Make 30 green aliens.
for alien_number in range(30):
    new_alien = {'color': 'green', 'points': 5, 'speed': 'slow'}
    aliens.append(new_alien)

for alien in aliens[:3]:
    if alien['color'] == 'green':
        alien['color'] = 'yellow'
        alien['speed'] = 'medium'
        alien['points'] = 10

# Show the first 5 aliens.
for alien in aliens[:5]:
    print(alien)
print("...")
~
~
```

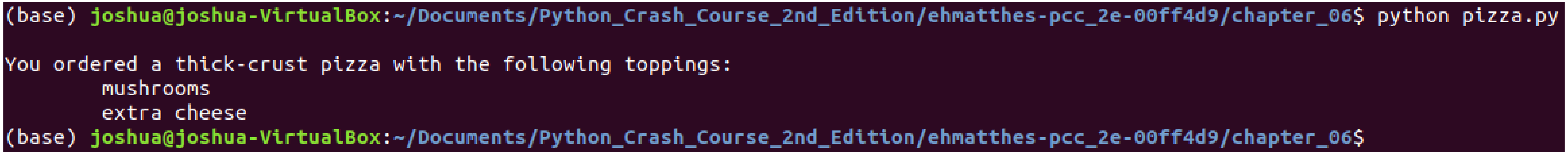


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python aliens.p
y
{'color': 'yellow', 'points': 10, 'speed': 'medium'}
{'color': 'yellow', 'points': 10, 'speed': 'medium'}
{'color': 'yellow', 'points': 10, 'speed': 'medium'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
...
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

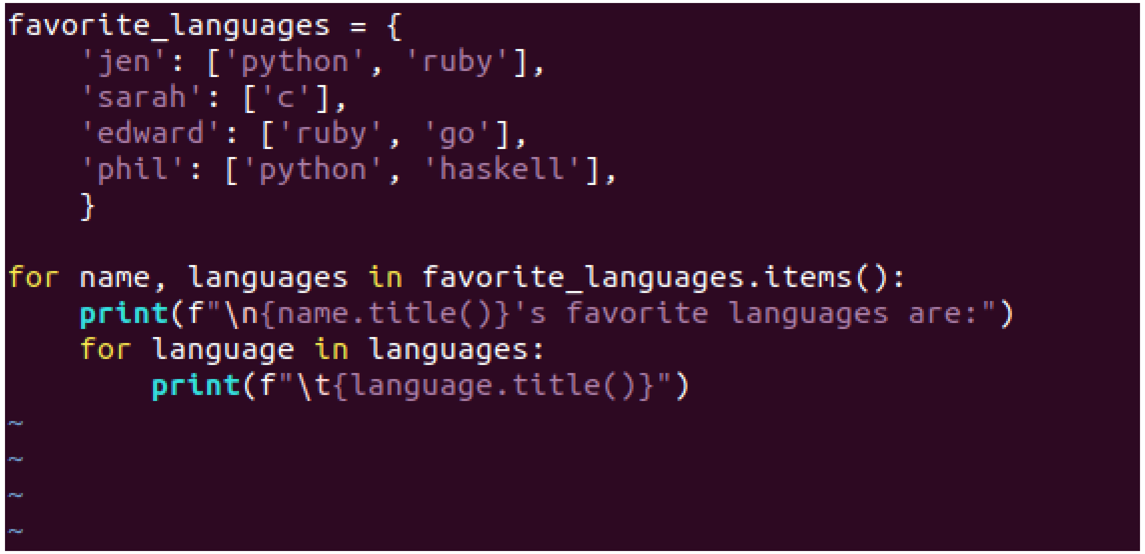
A List in a Dictionary



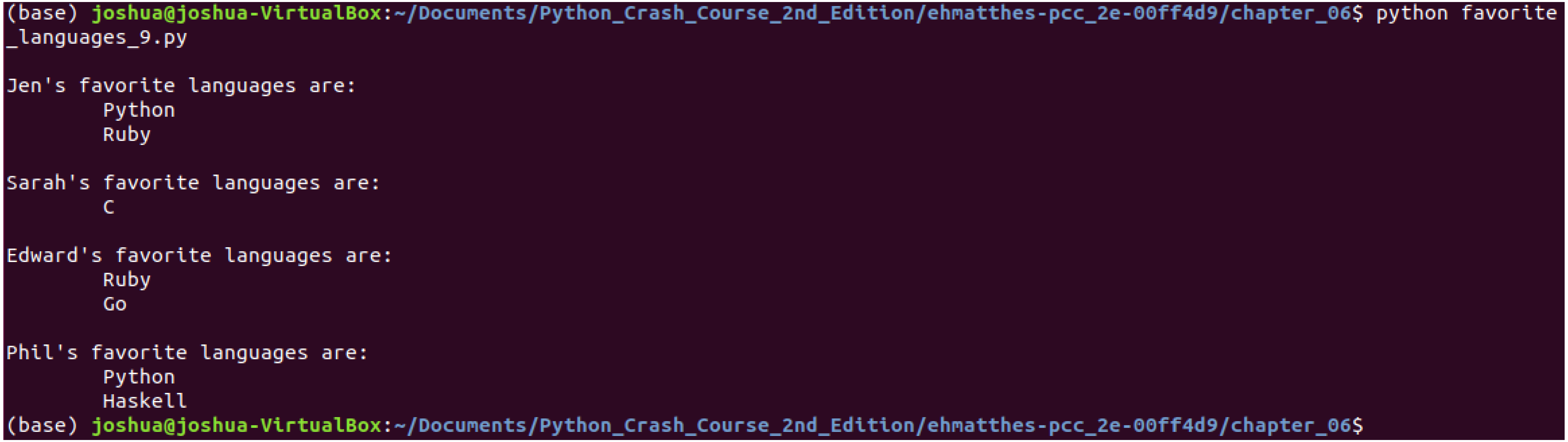
```
 # Store information about a pizza being ordered.
pizza = {
    'crust': 'thick',
    'toppings': ['mushrooms', 'extra cheese'],
    }

# Summarize the order.
print(f"You ordered a {pizza['crust']}-crust pizza "
    "with the following toppings:")

for topping in pizza['toppings']:
    print("\t" + topping)
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python pizza.py

You ordered a thick-crust pizza with the following toppings:
        mushrooms
        extra cheese
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

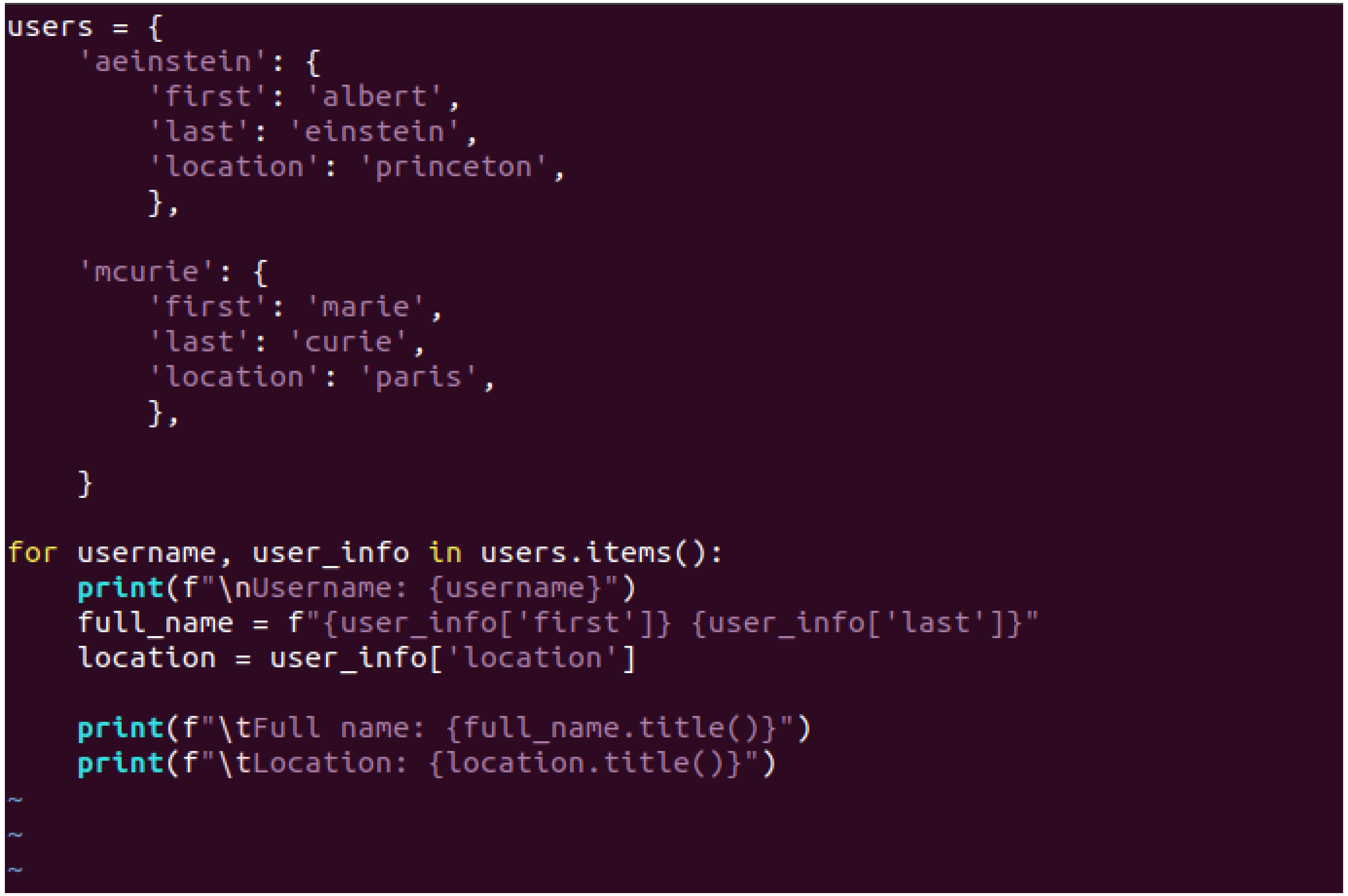


```
favorite_languages = {
    'jen': ['python', 'ruby'],
    'sarah': ['c'],
    'edward': ['ruby', 'go'],
    'phil': ['python', 'haskell'],
    }

for name, languages in favorite_languages.items():
    print(f"\n{name.title()}'s favorite languages are:")
    for language in languages:
        print(f"\t{language.title()}")
~
~
~
~
```

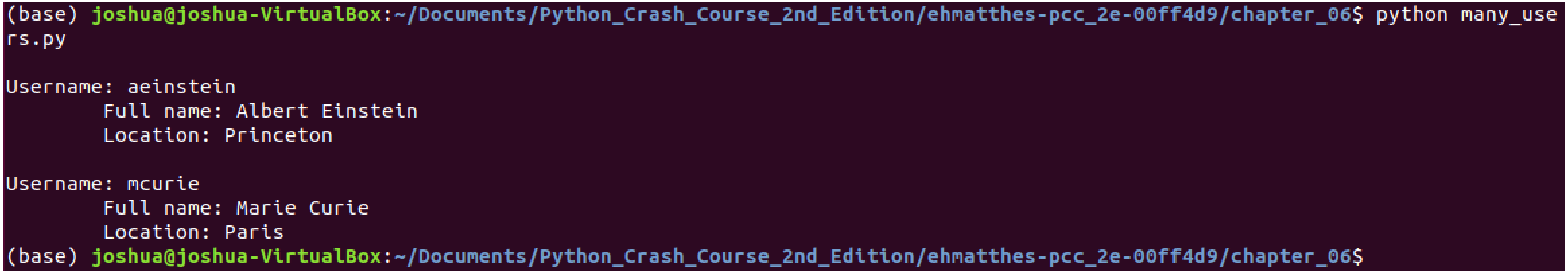


```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python favorite
_languages_9.py

Jen's favorite languages are:
	Python
	Ruby

Sarah's favorite languages are:
	C

Edward's favorite languages are:
	Ruby
	Go

Phil's favorite languages are:
	Python
	Haskell
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```

A Dictionary in a Dictionary



```
users = {
    'aeinstein': {
        'first': 'albert',
        'last': 'einstein',
        'location': 'princeton',
        },

    'mcurie': {
        'first': 'marie',
        'last': 'curie',
        'location': 'paris',
        },

    }

for username, user_info in users.items():
    print(f"\nUsername: {username}")
    full_name = f"{user_info['first']} {user_info['last']}"
    location = user_info['location']

    print(f"\tFull name: {full_name.title()}")
    print(f"\tLocation: {location.title()}")
~
~
~
```



```
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$ python many_use
rs.py

Username: aeinstein
        Full name: Albert Einstein
        Location: Princeton

Username: mcurie
        Full name: Marie Curie
        Location: Paris
(base) joshua@joshua-VirtualBox:~/Documents/Python_Crash_Course_2nd_Edition/ehmatthes-pcc_2e-00ff4d9/chapter_06$
```