Solutions - Chapter 6

6-1: Person

Use a dictionary to store information about a person you know. Store their first name, last name, age, and the city in which they live. You should have keys such as first\_name, last\_name, age, andcity. Print each piece of information stored in your dictionary.

person **=** {

'first\_name': 'eric',

'last\_name': 'matthes',

'age': 43,

'city': 'sitka',

}

**print**(person['first\_name'])

**print**(person['last\_name'])

**print**(person['age'])

**print**(person['city'])

Output:

eric

matthes

43

sitka

6-2: Favorite Numbers

Use a dictionary to store people’s favorite numbers. Think of five names, and use them as keys in your dictionary. Think of a favorite number for each person, and store each as a value in your dictionary. Print each person’s name and their favorite number. For even more fun, poll a few friends and get some actual data for your program.

favorite\_numbers **=** {

'mandy': 42,

'micah': 23,

'gus': 7,

'hank': 1000000,

'maggie': 0,

}

num **=** favorite\_numbers['mandy']

**print**("Mandy's favorite number is " **+** str(num) **+** ".")

num **=** favorite\_numbers['micah']

**print**("Micah's favorite number is " **+** str(num) **+** ".")

num **=** favorite\_numbers['gus']

**print**("Gus's favorite number is " **+** str(num) **+** ".")

num **=** favorite\_numbers['hank']

**print**("Hank's favorite number is " **+** str(num) **+** ".")

num **=** favorite\_numbers['maggie']

**print**("Maggie's favorite number is " **+** str(num) **+** ".")

Output:

Mandy's favorite number is 42.

Micah's favorite number is 23.

Gus's favorite number is 7.

Hank's favorite number is 1000000.

Maggie's favorite number is 0.

6-3: Glossary

A Python dictionary can be used to model an actual dictionary. However, to avoid confusion, let’s call it a glossary.

* Think of five programming words you’ve learned about in the previous chapters. Use these words as the keys in your glossary, and store their meanings as values.
* Print each word and its meaning as neatly formatted output. You might print the word followed by a colon and then its meaning, or print the word on one line and then print its meaning indented on a second line. Use the newline character ('\n') to insert a blank line between each word-meaning pair in your output.

glossary **=** {

'string': 'A series of characters.',

'comment': 'A note in a program that the Python interpreter ignores.',

'list': 'A collection of items in a particular order.',

'loop': 'Work through a collection of items, one at a time.',

'dictionary': "A collection of key-value pairs.",

}

word **=** 'string'

**print**("\n" **+** word**.**title() **+** ": " **+** glossary[word])

word **=** 'comment'

**print**("\n" **+** word**.**title() **+** ": " **+** glossary[word])

word **=** 'list'

**print**("\n" **+** word**.**title() **+** ": " **+** glossary[word])

word **=** 'loop'

**print**("\n" **+** word**.**title() **+** ": " **+** glossary[word])

word **=** 'dictionary'

**print**("\n" **+** word**.**title() **+** ": " **+** glossary[word])

Output:

String: A series of characters.

Comment: A note in a program that the Python interpreter ignores.

List: A collection of items in a particular order.

Loop: Work through a collection of items, one at a time.

Dictionary: A collection of key-value pairs.

6-4: Glossary 2

Now that you know how to loop through a dictionary, clean up the code from Exercise 6-3 (page 102) by replacing your series of print statements with a loop that runs through the dictionary’s keys and values. When you’re sure that your loop works, add five more Python terms to your glossary. When you run your program again, these new words and meanings should automatically be included in the output.

glossary **=** {

'string': 'A series of characters.',

'comment': 'A note in a program that the Python interpreter ignores.',

'list': 'A collection of items in a particular order.',

'loop': 'Work through a collection of items, one at a time.',

'dictionary': "A collection of key-value pairs.",

'key': 'The first item in a key-value pair in a dictionary.',

'value': 'An item associated with a key in a dictionary.',

'conditional test': 'A comparison between two values.',

'float': 'A numerical value with a decimal component.',

'boolean expression': 'An expression that evaluates to True or False.',

}

**for** word, definition **in** glossary**.**items():

**print**("\n" **+** word**.**title() **+** ": " **+** definition)

Output:

Dictionary: A collection of key-value pairs.

String: A series of characters.

Boolean Expression: An expression that evaluates to True or False.

Comment: A note in a program that the Python interpreter ignores.

Value: An item associated with a key in a dictionary.

Loop: Work through a collection of items, one at a time.

List: A collection of items in a particular order.

Conditional Test: A comparison between two values.

Key: The first item in a key-value pair in a dictionary.

Float: A numerical value with a decimal component.

6-5: Rivers

Make a dictionary containing three major rivers and the country each river runs through. One key-value pair might be 'nile': 'egypt'.

* Use a loop to print a sentence about each river, such as *The Nile runs through Egypt.*
* Use a loop to print the name of each river included in the dictionary.
* Use a loop to print the name of each country included in the dictionary.

rivers **=** {

'nile': 'egypt',

'mississippi': 'united states',

'fraser': 'canada',

'kuskokwim': 'alaska',

'yangtze': 'china',

}

**for** river, country **in** rivers**.**items():

**print**("The " **+** river**.**title() **+** " flows through " **+** country**.**title() **+** ".")

**print**("\nThe following rivers are included in this data set:")

**for** river **in** rivers**.**keys():

**print**("- " **+** river**.**title())

**print**("\nThe following countries are included in this data set:")

**for** country **in** rivers**.**values():

**print**("- " **+** country**.**title())

Output\*:

The Mississippi flows through United States.

The Yangtze flows through China.

The Fraser flows through Canada.

The Nile flows through Egypt.

The Kuskokwim flows through Alaska.

The following rivers are included in this data set:

- Mississippi

- Yangtze

- Fraser

- Nile

- Kuskokwim

The following countries are included in this data set:

- United States

- China

- Canada

- Egypt

- Alaska

\*Sometimes we like to think of Alaska as our own separate country.

6-6: Polling

Use the code in *favorite\_languages.py* (page 104).

* Make a list of people who should take the favorite languages poll. Include some names that are already in the dictionary and some that are not.
* Loop through the list of people who should take the poll. If they have already taken the poll, print a message thanking them for responding. If they have not yet taken the poll, print a message inviting them to take the poll.

favorite\_languages **=** {

'jen': 'python',

'sarah': 'c',

'edward': 'ruby',

'phil': 'python',

}

**for** name, language **in** favorite\_languages**.**items():

**print**(name**.**title() **+** "'s favorite language is " **+**

language**.**title() **+** ".")

**print**("\n")

coders **=** ['phil', 'josh', 'david', 'becca', 'sarah', 'matt', 'danielle']

**for** coder **in** coders:

**if** coder **in** favorite\_languages**.**keys():

**print**("Thank you for taking the poll, " **+** coder**.**title() **+** "!")

**else**:

**print**(coder**.**title() **+** ", what's your favorite programming language?")

Output:

Jen's favorite language is Python.

Sarah's favorite language is C.

Phil's favorite language is Python.

Edward's favorite language is Ruby.

Thank you for taking the poll, Phil!

Josh, what's your favorite programming language?

David, what's your favorite programming language?

Becca, what's your favorite programming language?

Thank you for taking the poll, Sarah!

Matt, what's your favorite programming language?

Danielle, what's your favorite programming language?

6-7: People

Start with the program you wrote for Exercise 6-1 (page 102). Make two new dictionaries representing different people, and store all three dictionaries in a list called people. Loop through your list of people. As you loop through the list, print everything you know about each person.

*# Make an empty list to store people in.*

people **=** []

*# Define some people, and add them to the list.*

person **=** {

'first\_name': 'eric',

'last\_name': 'matthes',

'age': 43,

'city': 'sitka',

}

people**.**append(person)

person **=** {

'first\_name': 'ever',

'last\_name': 'matthes',

'age': 5,

'city': 'sitka',

}

people**.**append(person)

person **=** {

'first\_name': 'willie',

'last\_name': 'matthes',

'age': 8,

'city': 'sitka',

}

people**.**append(person)

*# Display all of the information in the dictionary.*

**for** person **in** people:

name **=** person['first\_name']**.**title() **+** " " **+** person['last\_name']**.**title()

age **=** str(person['age'])

city **=** person['city']**.**title()

**print**(name **+** ", of " **+** city **+** ", is " **+** age **+** " years old.")

Output:

Eric Matthes, of Sitka, is 43 years old.

Ever Matthes, of Sitka, is 5 years old.

Willie Matthes, of Sitka, is 8 years old.

6-8: Pets

Make several dictionaries, where the name of each dictionary is the name of a pet. In each dictionary, include the kind of animal and the owner’s name. Store these dictionaries in a list called pets. Next, loop through your list and as you do print everything you know about each pet.

*# Make an empty list to store the pets in.*

pets **=** []

*# Make individual pets, and store each one in the list.*

pet **=** {

'animal type': 'python',

'name': 'john',

'owner': 'guido',

'weight': 43,

'eats': 'bugs',

}

pets**.**append(pet)

pet **=** {

'animal type': 'chicken',

'name': 'clarence',

'owner': 'tiffany',

'weight': 2,

'eats': 'seeds',

}

pets**.**append(pet)

pet **=** {

'animal type': 'dog',

'name': 'peso',

'owner': 'eric',

'weight': 37,

'eats': 'shoes',

}

pets**.**append(pet)

*# Display information about each pet.*

**for** pet **in** pets:

**print**("\nHere's what I know about " **+** pet['name']**.**title() **+** ":")

**for** key, value **in** pet**.**items():

**print**("\t" **+** key **+** ": " **+** str(value))

Output:

Here's what I know about John:

weight: 43

animal type: python

name: john

owner: guido

eats: bugs

Here's what I know about Clarence:

weight: 2

animal type: chicken

name: clarence

owner: tiffany

eats: seeds

Here's what I know about Peso:

weight: 37

animal type: dog

name: peso

owner: eric

eats: shoes

6-9: Favorite Places

Make a dictionary called favorite\_places. Think of three names to use as keys in the dictionary, and store one to three favorite places for each person. To make this exericse a bit more interesting, ask some friends to name a few of their favorite places. Loop through the dictionary, and print each person’s name and their favorite places.

favorite\_places **=** {

'eric': ['bear mountain', 'death valley', 'tierra del fuego'],

'erin': ['hawaii', 'iceland'],

'ever': ['mt. verstovia', 'the playground', 'south carolina']

}

**for** name, places **in** favorite\_places**.**items():

**print**("\n" **+** name**.**title() **+** " likes the following places:")

**for** place **in** places:

**print**("- " **+** place**.**title())

Output:

Ever likes the following places:

- Mt. Verstovia

- The Playground

- South Carolina

Erin likes the following places:

- Hawaii

- Iceland

Eric likes the following places:

- Bear Mountain

- Death Valley

- Tierra Del Fuego

6-10: Favorite Numbers

Modify your program from Exercise 6-2 (page 102) so each person can have more than one favorite number. Then print each person’s name along with their favorite numbers.

favorite\_numbers **=** {

'mandy': [42, 17],

'micah': [42, 39, 56],

'gus': [7, 12],

}

**for** name, numbers **in** favorite\_numbers**.**items():

**print**("\n" **+** name**.**title() **+** " likes the following numbers:")

**for** number **in** numbers:

**print**(" " **+** str(number))

Output:

Micah likes the following numbers:

42

39

56

Mandy likes the following numbers:

42

17

Gus likes the following numbers:

7

12

## 6-11: Cities

Make a dictionary called cities. Use the names of three cities as keys in your dictionary. Create a dictionary of information about each city and include the country that the city is in, its approximate population, and one fact about that city. The keys for each city’s dictionary should be something like country, population, and fact. Print the name of each city and all of the information you have stored about it.

cities **=** {

'santiago': {

'country': 'chile',

'population': 6158080,

'nearby mountains': 'andes',

},

'talkeetna': {

'country': 'alaska',

'population': 876,

'nearby mountains': 'alaska range',

},

'kathmandu': {

'country': 'nepal',

'population': 1003285,

'nearby mountains': 'himilaya',

}

}

**for** city, city\_info **in** cities**.**items():

country **=** city\_info['country']**.**title()

population **=** city\_info['population']

mountains **=** city\_info['nearby mountains']**.**title()

**print**("\n" **+** city**.**title() **+** " is in " **+** country **+** ".")

**print**(" It has a population of about " **+** str(population) **+** ".")

**print**(" The " **+** mountains **+** " mountains are nearby.")

Output:

Santiago is in Chile.

It has a population of about 6158080.

The Andes mountains are nearby.

Kathmandu is in Nepal.

It has a population of about 1003285.

The Himilaya mountains are nearby.

Talkeetna is in Alaska.

It has a population of about 876.

The Alaska Range mountains are nearby.