6. Dictionaries

A Simple Dictionary

In [1]:

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

```
green
5
```

- we made a *dictionary* with 2 informations
- the color of the alien and the points of it
- a dictionary has some *key-value pairs*
- a key has a value which can be any object in python (a string, integer, bool, or whatever)
- every key is connected with a colon
- individual key-value pair are connected by commas
- you can store as many key-value pairs in a dictionary
- the simplest dictionary has exactly one pair

Working with Dictionaries

In [2]:

```python
alien_0 = {'color': 'green'}
print(alien_0['color'])
```

```
green
```

- **accessing values in a dictionary**
- here we get the value of the key by targeting the dictionary and indexing the key to get the value out

In [4]:

```python
alien_0 = {'color': 'green', 'points': 5}
new_points = alien_0['points']
print("You just earned " + str(new_points) + " points!")
```

```
You just earned 5 points!
```

- storing a keys value in a variable and printing it out in the print statement (don't forget to convert it to a string)
- note that the value will change, so this is not like a tuple!

In [8]:

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

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

```
{'color': 'green', 'points': 5}
{'color': 'green', 'points': 5, 'x_position': 0, 'y_position': 25}
```

- **adding new key-value pairs**
- dictionaries are dynamic
- you can add new value pairs
- give the name of the dictionary
- followed by the new key-value pair in square brackets

In [9]:

```
alien_0 = {}

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

print(alien_0)
```

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

- **starting with an empty dictionary**
- define an empty dictionary
- and add each key with the value
- convenient for storing user data

In [13]:

```
alien_0 = {'color': 'green', 'points': 5}
print("The alien is " + alien_0['color'] + ".")

alien_0['color'] = 'yellow'
print("The alien is " + alien_0['color'] + ".")
```

```
The alien is green.
The alien is yellow.
```

- **modify values in a dictionary**
- give the name of the dictionary with the key you want to modify
- and simply define a new value to it

In [12]:

```
alien_0 = {'x_position': 0, 'y_position': 25, 'speed': 'medium'}
print("Original x-position: " + str(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("New x-position: " + str(alien_0['x_position']))
```

```
Original x-position: 0
New x-position: 2
```

- we define an initial x position which is 0, an y position which is 25 and a speed which is 'medium*
- we also print the original x position
- first we do a if-elif-else chain which increments the x position of the object
    - if the speed is '*slow*' than it increments it by 1
    - if the speed is '*medium*' than it increments it by 2
    - if the speed is '*more than medium*' than it increments it by 3
- then we modify the key-value of the x_position in our dictionary by simply **adding** the new *x_increment*
- we finish by printing out the new x-position with the modified new kew-value for the x-position
- because we have a medium speed we have a new x-position of 2

In [14]:

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

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

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

- **removing key-value pairs**
- with the ***del*** statement we can delete key-value pairs from a dictionary
- the pair is deleted permanently

In [17]:

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

print("Sarah's favorite language is " +
      favorite_languages['sarah'].title() +
      ".")
```

```
Sarah's favorite language is C.
```

- **a dictionary has similar objects**
- many objects can have the same value
- this dictionary is broken into several lines
- after the last pair put a comma so you can add more pairs
- we can use the value of the key in our print statement
- you can also break up a long print statement into several lines

Tasks

- 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, and city. Print each piece of information stored in your dictionary.
- 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.
- 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.

In [63]:

```
# 6.1
nico = {
    'first_name': 'nico',
    'last_name': 'foschia',
    'age': 27,
    'city': 'paris',
}

for key, value in nico.items():
    print("\n" + key.replace("_", " ").title() + ": " + str(value).title())
```

```
First Name: Nico

Last Name: Foschia

Age: 27

City: Paris
```

- we use here **the replace()** method to replace the underscore with whitespace

In [64]:

```
# 6.2
favorite_numbers = {
    'nico': 16,
    'roman': 31,
    'daphne': 1,
    'simon': 13,
    'laure': 11,
}

for key, value in favorite_numbers.items():
    print("\n" + key.title() + ": " + str(value))
```

```
Nico: 16

Roman: 31

Daphne: 1

Simon: 13

Laure: 11
```

In [78]:

```
# 6.3
glossary = {
    'loop': 'doing things again and again',
    'print': 'printing out things',
    'method': 'does something for your help',
    'function': "i don't know the difference between it and the method",
    'pythion': 'seems to be a pretty nice language',
}

for key, value in glossary.items():
    print("\n" + key.title() + ": " + '"' + value.capitalize()  + '."' )
```

```
Loop: "Doing things again and again."

Print: "Printing out things."

Method: "Does something for your help."

Function: "I don't know the difference between it and the method."

Pythion: "Seems to be a pretty nice language."
```

- the **capitalize() method** converts the first character of a string to capital (uppercase) letter

Looping Through a Dictionary

In [24]:

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

for key, value in user_0.items():
    print("\nKey: " + key)
    print("Value: " + value)
```

```
Key: username
Value: efermi

Key: first
Value: enrico

Key: last
Value: fermi
```

- **looping through all key-value pairs**
- we want to see everything stored in this user's dictionary
- we create to variables **key** and **value** to loop from user_0 dictionary
    - we want to print out each key
    - we want to print out each value
- The *items()* method returns a view object that displays a list of a given dictionary's (key, value) tuple pair.

In [27]:

```
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() + ".")
```

```
Jen's favorite language is Python.
Sarah's favorite language is C.
Edward's favorite language is Ruby.
Phil's favorite language is Python.
```

- same works also for dictionaries with keys having same values

In [26]:

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

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

```
Jen
Sarah
Edward
Phil
```

- **looping through all the keys in a dictionary**
- here we are looping a dictionaries keys out
- The *keys()* method returns a view object. The view object contains the keys of the dictionary, as a list.
- looping through the keys of a dictionionary is actually the default
- but saying *key()* makes the code more readable

In [30]:

```
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:
        print(" Hi " + name.title() +
         ", I see your favorite language is " +
         favorite_languages[name].title() + "!")
```

```
Jen
Sarah
 Hi Sarah, I see your favorite language is C!
Edward
Phil
 Hi Phil, I see your favorite language is Python!
```

- we have a given dictionary
- we have a list of friedns
- we loop our dictionaries keys
- we print all our keys from our dictionary
- and
    - we have a if statement which is true for all friends from the list
    - which has a print method as action for all our friends in our list

In [31]:

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

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

```
Erin, please take our poll!
```

- we use the *keys()* method to check if a key is in our dictionary
- note the *not* keyword

In [32]:

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

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

```
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.
```

- this time we sort the output of the dictionaries keys in alphabetical order
- we have just wrapped the *sorted() function* around

In [33]:

```
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())
```

```
The following languages have been mentioned:
Python
C
Ruby
Python
```

- **looping through all values in a dictionary**
- we are interested just in the values
- we use the ***values()*** method to return the values of the dictionary

In [34]:

```
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())
```

```
The following languages have been mentioned:
Ruby
Python
C
```

- to see each value just once we can use the ***set()***
- it identifies any duplicate from a list and gives the unique items

Tasks

- 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.
- 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.
- 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.

In [81]:

```python
# 6.4
glossary = {
    'loop': 'doing things again and again',
    'print': 'printing out things',
    'method': 'does something for your help',
    'function': "i don't know the difference between it and the method",
    'python': 'seems to be a pretty nice language',
    'while loop': 'i will do that in the next chapter',
    'if-elif-else chain': 'is everything you need babe',
}

for key, value in glossary.items():
    print("\n" + key.title() + ": " + '"' + value.capitalize()  + '."' )
```

```
Loop: "Doing things again and again."

Print: "Printing out things."

Method: "Does something for your help."

Function: "I don't know the difference between it and the method."

Python: "Seems to be a pretty nice language."

While Loop: "I will do that in the next chapter."

If-Elif-Else Chain: "Is everything you need babe."
```

In [91]:

```
#6.5
rivers = {
    'rhein': 'germany',
    'nile': 'egypt',
    'mississippi': 'u.s.a.',
}

for key, value in rivers.items():
    print("The " + key.title() + " runs through " + value.title() + ".")


print("\n")

for key in rivers.keys():
    print("The " + key.title() + " is in the list!")

print("\n")

for value in rivers.values():
    print(value.title() + " is in the list!")
```

```
The Rhein runs through Germany.
The Nile runs through Egypt.
The Mississippi runs through U.S.A..


The Rhein is in the list!
The Nile is in the list!
The Mississippi is in the list!


Germany is in the list!
Egypt is in the list!
U.S.A. is in the list!
```

In [109]:

```
# 6.6
poll_done = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}

names = ['jen', 'sarah', 'edward', 'phil', 'jacob']

for name in names:
    if name in poll_done.keys():
        print(name.title() + " did the Poll, thanks!")
    else:
        print(name.title() + ", please do the Poll!")
```

```
Jen did the Poll, thanks!
Sarah did the Poll, thanks!
Edward did the Poll, thanks!
Phil did the Poll, thanks!
Jacob, please do the Poll!
```

Nesting

In [35]:

```
alien_0 = {'color': 'green', 'points': 5}
alien_1 = {'color': 'yellow', 'points': 10}
alien_2 = {'color': 'red', 'points': 15}

aliens = [alien_0, alien_1, alien_2]

for alien in aliens:
    print(alien)
```

```
{'color': 'green', 'points': 5}
{'color': 'yellow', 'points': 10}
{'color': 'red', 'points': 15}
```

- **a list of dictionaries**
- we have 3 dictionaries
- we can store them all in a list

In [36]:

```
# 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)

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

# Show how many aliens have been created.
print("Total number of aliens: " + str(len(aliens)))
```

```
{'color': 'green', 'points': 5, 'speed': 'slow'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
{'color': 'green', 'points': 5, 'speed': 'slow'}
...
Total number of aliens: 30
```

- we make an empty list
- we create 30 aliens with the **range()**
- our action for the loop is
    - we create a dictionary *new_alien*
    - we append this to the list *aliens*
- we want to see the first 5 items of the list
- we do that with a for loop again
    - the print(...) is just to make a new line
- at the end we show how many items we created
    - we want to know our lenght of our list
    - don#t forget to make it str()

In [37]:

```python
# 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[0:3]:
    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("...")
```

```
{'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'}
...
```

- we want to change the first three elements of our list
- we create again an empty list
- we create also 30 green aliens again
- we say that for the first 3 elements of our list (slice)
    - if the aliens are green (which they are) --> we just want to modify green ones!!!
        - we change the color to yellow
        - we change the speed to medium
        - we change the points to 10
- we show again the first five aliens

In [39]:

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

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

for alien in aliens[0:3]:
    if alien['color'] == 'green':
        alien['color'] = 'yellow'
        alien['speed'] = 'medium'
        alien['points'] = 10
    elif alien['color'] == 'yellow':
        alien['color'] = 'red'
        alien['speed'] = 'fast'
        alien['points'] = 15

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

```
{'color': 'red', 'points': 15, 'speed': 'fast'}
{'color': 'red', 'points': 15, 'speed': 'fast'}
{'color': 'red', 'points': 15, 'speed': 'fast'}
{'color': 'yellow', 'points': 5, 'speed': 'slow'}
{'color': 'yellow', 'points': 5, 'speed': 'slow'}
...
```

- we changed the color of the 30 aliens into yellow
- we added an elif statement which will make all yellow aliens
    - red
    - fast
    - and give them 15 points

In [41]:

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

# Summarize the order.
print("You ordered a " + pizza['crust'] + "-crust pizza " +
     "with the following toppings:")
for topping in pizza['toppings']:
    print("\t" + topping)
```

```
You ordered a thick-crust pizza with the following toppings:
        mushrooms
        extra cheese
```

- **a list in a dictionary**
- we can store a list in a dictionary
- the key-value pair looks like this:
    - the key is the same like in the previous examples
    - the value is a list though!

In [44]:

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

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

```
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
```

- we have a dictionary of favorite languages for each user
- we have our first for loop which prints each users name
- we have our second for loop inside the for loop which prints each language from the dictionary
- we have 2 variables (*name* and *languages*) and use the *item()* function to itemize the dictionary

In [47]:

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

for name, languages in favorite_languages.items():
    if len(languages) == 1:
        print("\n" + name.title() + "'s favorite languages is:")
        print("\t" + language.title())
    else:
        print("\n" + name.title() + "'s favorite languages are:")
        for language in languages:
            print("\t" + language.title())
```

```
Jen's favorite languages are:
        Python
        Ruby

Sarah's favorite languages is:
        Ruby

Edward's favorite languages are:
        Ruby
        Go

Phil's favorite languages are:
        Python
        Haskell
```

- we modified the version with the **len()** function
- we check the lenght of the dictionaries item
- if its just 1 we say a persons favorite language **is:**
- else we say **are:**

In [50]:

```
users = {
    'aeinstein': {
        'first': 'albert',
        'last': 'einstein',
        'location': 'princeton',
    },
    'mcurie': {
        'first': 'marie',
        'last': 'curie',
        'location': 'paris',
    },
}

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

    print("\tFull name: " + full_name.title())
    print("\tLocation: " + location.title())
```

```
Username: aeinstein
        Full name: Albert Einstein
        Location: Princeton

Username: mcurie
        Full name: Marie Curie
        Location: Paris
```

- **a dictionary in a dictionary**
- we first define a dictionary called users with two keys
    - the keys have dictionaries as values
- we create a for loop with two variables (*username* and *user_info*)
- which loops through all the items of our dictionary
    - first it prints the username by using the username variable, which is the first defined variable in our for loop
    - then the it stores into a new variable *full_name* from the item *user_info* (which is the second defined variable of our for loop) --> the two values '*first*' and '*last*'
    - the same is done also for the location key which uses again our second defined variable from the for loop (user_info) and puts in into the variable *location*
- at the end it simply prints our new two variables *full_name* and *location* in a print statement

Tasks

- 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.
- 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.
- 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 exercise 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.
- 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.
- 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 infor- mation you have stored about it.
- 6-12. Extensions: We're now working with examples that are complex enough that they can be extended in any number of ways. Use one of the example pro- grams from this chapter, and extend it by adding new keys and values, chang- ing the context of the program or improving the formatting of the output.

In [117]:

```
# 6.7
nico = {
    'first_name': 'nico',
    'last_name': 'foschia',
    'age': 27,
    'city': 'paris',
}

laure = {
    'first_name': 'laure',
    'last_name': 'laurier',
    'age': 24,
    'city': 'mannheim',
}

daphne = {
    'first_name': 'daphne',
    'last_name': 'daph',
    'age': 24,
    'city': 'paris',
}

people = [nico, laure, daphne]

for person in people:
    print(person)
```

```
{'first_name': 'nico', 'last_name': 'foschia', 'age': 27, 'city': 'p
aris'}
{'first_name': 'laure', 'last_name': 'laurier', 'age': 24, 'city':
'mannheim'}
{'first_name': 'daphne', 'last_name': 'daph', 'age': 24, 'city': 'pa
ris'}
```

In [118]:

```
#6.8
dog = {
    'kind of animal': 'nice',
    'owners name': 'paul'
}

bird = {
    'kind of animal': 'bad',
    'owners name': 'josef',
}

pets = [dog, bird]

for pet in pets:
    print(pet)
```

```
{'kind of animal': 'nice', 'owners name': 'paul'}
{'kind of animal': 'bad', 'owners name': 'josef'}
```

In [129]:

```
#6.9
favorite_places = {
    'jen': ['rome', 'milan'],
    'sarah': ['berlin'],
    'edward': ['berlin', 'prague'],
    'phil': ['budapest', 'new york'],
}

for name, places in favorite_places.items():
    if len(places) == 1:
        print("\n" + name.title() + "'s favorite place is:")
        print("\t" + place.title())
    else:
        print("\n" + name.title() + "'s favorite places are:")
        for place in places:
            print("\t" + place.title())
```

```
Jen's favorite places are:
        Rome
        Milan

Sarah's favorite place is:
        Milan

Edward's favorite places are:
        Berlin
        Prague

Phil's favorite places are:
        Budapest
        New York
```

In [138]:

```
#6.10
favorite_numbers = {
    'nico': [16, 17, 18],
    'roman': [26, 37, 38],
    'daphne': [6, 7, 8],
    'simon': [1, 7, 18],
    'laure': [16, 1, 180],
}

for person, numbers in favorite_numbers.items():
    print("\n" + person.title() + "'s favorite numbers are:")
    for number in numbers:
        print("\t" + str(number))
```

```
Nico's favorite numbers are:
        16
        17
        18

Roman's favorite numbers are:
        26
        37
        38

Daphne's favorite numbers are:
        6
        7
        8

Simon's favorite numbers are:
        1
        7
        18

Laure's favorite numbers are:
        16
        1
        180
```

- 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 infor- mation you have stored about it.

In [142]:

```
cities = {'frankfurt': {
    'country': 'germany',
    'population': '100',
    'fact': 'big',
} ,
         'mannheim': {
    'country': 'germany',
    'population': '10000',
    'fact': 'huge',

         } ,
          'stuttgart': {
    'country': 'germany',
    'population': '10',
    'fact': 'small',

          } ,
         }

for cityname, city_info in cities.items():
    print("\nCityname: " + cityname.title())
    print("\nCityinfo: " + str(city_info))
```

```
Cityname: Frankfurt

Cityinfo: {'country': 'germany', 'population': '100', 'fact': 'big'}

Cityname: Mannheim

Cityinfo: {'country': 'germany', 'population': '10000', 'fact': 'hug
e'}

Cityname: Stuttgart

Cityinfo: {'country': 'germany', 'population': '10', 'fact': 'smal
l'}
```