Section Overview

What You Will Learn

Dictionaries
Creating
Adding and removing items
Searching
Nesting
Looping

Dictionaries

Part I

Dictionaries

Part II

Dictionaries

- Hold key-value pairs called items.
- AKA associative arrays, hash tables and hashes.

```
dictionary_name = {key_1: value_1, key_N: value_N}

dictionary_name = {}

dictionary_name[key]
```

```
contacts = {'Jason': '555-0123', 'Carl': '555-0987'}
jasons_phone = contacts['Jason']
carls_phone = contacts['Carl']

print('Dial {} to call Jason.'.format(jasons_phone))
print('Dial {} to call Carl.'.format(carls_phone))
```

```
Dial 555-0123 to call Jason.
Dial 555-0987 to call Carl.
```

```
contacts = {'Jason': '555-0123', 'Carl': '555-0987'}
contacts['Jason'] = '555-0000'
jasons_phone = contacts['Jason']
print('Dial {} to call Jason.'.format(jasons_phone))
```

```
Dial 555-0000 to call Jason.
```

```
contacts = {'Jason': '555-0123', 'Carl': '555-0987'}
contacts['Tony'] = '555-0570'
print(contacts)
print(len(contacts))
```

```
{'Jason': '555-0123', 'Carl': '555-0987', 'Tony':
'555-0570'}
3
```

```
contacts = {'Jason': '555-0123', 'Carl': '555-0987'}
del contacts['Jason']
print(contacts)
```

```
{'Carl': '555-0987'}
```

```
contacts = {
    'Jason': ['555-0123', '555-0000'],
    'Carl': '555-0987'
}
print('Jason:')
print(contacts['Jason'])
print('Carl:')
print(contacts['Carl'])
```

```
Jason:
['555-0123', '555-0000']
Carl:
555-0987
```

```
contacts = {
    'Jason': ['555-0123', '555-0000'],
    'Carl': '555-0987'
}

for number in contacts['Jason']:
    print('Phone: {}'.format(number))
```

```
Phone: 555-0123
Phone: 555-0000
```

```
contacts = {
    'Jason': ['555-0123', '555-0000'],
    'Carl': '555-0987'
}

if 'Jason' in contacts.keys():
    print("Jason's phone number is:")
    print(contacts['Jason'][0])

if 'Tony' in contacts.keys():
    print("Tony's phone number is:")
    print(contacts['Tony'][0])
```

```
Jason's phone number is:
555-0123
```

```
contacts = {
    'Jason': ['555-0123', '555-0000'],
    'Carl': '555-0987'
}

print('555-0987' in contacts.values())
```

```
True
```

Loops

```
for key_variable in dictionary_name:
    # Code block
    # dictionary_name[key_variable]
```

```
for contact in contacts:
    # Code block
for person in people:
    # Code block
```

```
contacts = {
    'Jason': '555-0123',
    'Carl': '555-0987'
}
for contact in contacts:
    print('The number for {0} is
{1}.'.format(contact, contacts[contact]))
```

```
The number for Carl is 555-0987.
The number for Jason is 555-0123.
```

Looping with two variables

```
for key_variable, value_variable in dictionary_name.items():
    # Code block
```

```
contacts = {
    'Jason': '555-0123',
    'Carl': '555-0987'
}
for person, phone_number in contacts.items():
    print('The number for {0} is
{1}.'.format(person, phone_number))
```

```
The number for Carl is 555-0987.
The number for Jason is 555-0123.
```

```
contacts = {
    'Jason': {
        'phone': '555-0123',
        'email': 'jason@example.com'
    },
    'Carl': {
        'phone': '555-0987',
        'email': 'carl@example.com'
    }
}
```

```
for contact in contacts:
    print("{}'s contact info:".format(contact))
    print(contacts[contact]['phone'])
    print(contacts[contact]['email'])
```

```
Jason's contact info:
555-0123
jason@example.com
Carl's contact info:
555-0987
carl@example.com
```

Section Summary

Summary

Dictionaries hold key-value pairs, called items.

```
dictionary_name = {key_1: value_1, key_N: value_N}
```

Access the values stored in a dictionary by key.

```
dictionary_name[key]
```

Summary

You can add or change values in a dictionary through assignment.

```
dictionary_name[key] = value
```

Remove items from a dictionary using the del statement.

```
del dictionary_name[key]
```

Summary

To determine if a value exists use the `value in dictionary_name.values()` syntax, which returns a boolean.

The `values()` dictionary method returns a list of the values stored in that dictionary.

Summary

Loop through a dictionary using:

```
for key_variable in dictionary_name:
```

Dictionary values can be of any data type, including other dictionaries.