Dictionaries

Rose-Hulman Institute of Technology

Computer Science and Software Engineering

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Data *Collections*

- Frequently several individual pieces of data are related
- We can collect them together in one object
- Examples:
  - List or tuple: contains an ordered sequence of items
  - Custom object, like **Line**: contains two endpoints, a color, and the window in which it is drawn
  - Dictionary: contains key-value pairs

Coming soon!

Review: Lists

- Ordered collection of items
- Usually homogeneous, but not required
- Access is by position (index) in the list

```
>>> animals = ['dog', 'cat', 'cow']
>>> animals[1]
'cat'
>>> animals[1:3]
['cat', 'cow']
>>> animals[1] = ['pig']
>>> animals
['dog', ['pig'], 'cow']
```

More List Mutations

- Items can be added, removed, or replaced

```
>>> animals = ['dog', 'cat', 'cow']
>>> animals.append('pig')
>>> animals
['dog', 'cat', 'cow', 'pig']
>>> animals[1:3] = ['cow', 'cat', 'goat']
>>> animals
['dog', 'cow', 'cat', 'goat', 'pig']
>>> animals[1:2] = []
>>> animals
['dog', 'cat', 'goat', 'pig']
```

Dictionary

- A collection of key-value pairs
- Items are not stored in any particular order
- Dictionary keys work like list indexes
  - offices[‘Curt’] = “F222”
  - print(offices[‘Curt’])
- No duplicate keys
- Keys must be immutable
  - i.e., number, string, tuple

Try It!

- Experiment in PyDev console
- Try the following:

```
>>> myDict = {'name':'Dave', 'gpa':3.5}
>>> print (myDict)
>>> myDict['name']
>>> myDict['gpa']
>>>dir(dict)
```

Dictionary Methods

Suppose **dict1** is a dictionary…

Method Call	Result
dict1.get(k, d)	if **k** is a key in the dictionary return the value for that key, else return **d**
dict1.pop(k, d)	if **k** is a key in the dictionary remove **k** and return the value for it, else return **d**
k in dict1	if **k** is a key in the dictionary return **True**, otherwise return **False**
dict1.keys()	list of **dict1**'s keys
dict1.values()	list of **dict1**'s values
dict1.items()	list of **dict1**'s (key, value) pairs, as tuples

Look at dictionaryMethods.py

Examples

Q2-5

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Use Dictionaries For…

- A collection of similar objects with fast lookup by key
- Storing different properties of a single object

Use 1: Collection of similar objects

- Examples:
  - A movie database in which we use the title as the key and look up the director.
  - A phone database in which we use the person's name as the key and look up the phone number

concordance.py

Live Coding

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Use 2: Properties of a Single Object

- Represent a playing card as a dictionary
- Properties: 'cardName', ' suit', 'value'

```
# A card is represented by a dictionary
# with keys cardName, suit, and value
def makeCard (cardName, suit, value):
    card = {}
    card['suit'] = suit
    card['cardName'] = cardName
    card['value'] = value
    return card
```

Exercise: **blackjackWithDictionaries**