CS521 O2
Information Structures with Python

Lecture 7

Guanglan Zhang

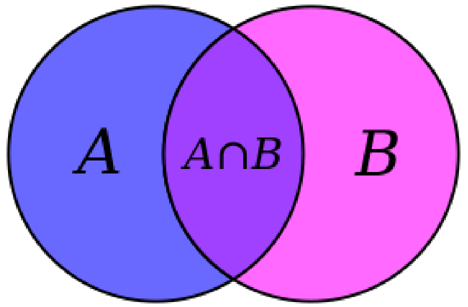
guanglan@bu.edu

Some slides adapted from Prof. Eugene Pinsky

Table of Content

Sets

- We can think of a set as a well-defined collection of distinct objects, typically called elements or members



- Python sets are collections of unordered, unique elements

- A set is mutable, but the elements contained in the set are immutable

- The elements can be objects of different types

- To create a set
  - ✓ Use constructor *set(<iterable>)*
  - ✓ Or use curly braces, {<obj>, ..., <obj>}
  - ✓ Python interprets empty curly braces {} as an empty dictionary
  - ✓ The only way to define an empty set is to use *set()*

Functions and Operations for Sets

- Membership operators *in*, and *not in* can be applied to sets
- *for* iteration works on sets
- Function *len()* can be applied to sets
- Functions *sum()*, *min()*, and *max()* can be applied to some sets

Set operations

- There are a host of operations on set objects that mimic the operations that are defined for mathematical sets, such as union and intersection
- Set operations can be done in two ways: by operator or by method
- To perform set union: set1.union(<set2>) or set1 | set2
- To get intersection: set1.intersection(<set2>) or set1 & set2
- To get all elements in set1 but not in set2: set1. difference(<set2>) or set1 - set2
- To get all elements in either set1 or set2, but not both:
  set1. symmetric_difference(<set2>) or set1 ^ set2
- To check if two sets have anything in common: set1.isdisjoint(<set2>)

Set operations (cont’)

- To check if set1 is a subset of set2: set1.issubset(<set2>) or set1 <= set2
- To check if set1 is a proper subset of set2: set1 < set2 (no corresponding method)
- To check if set1 is a superset of set2: set1.issuperset(<set2>) or set1 >= set2
- To check if set1 is a proper superset of set2: set1 > set2 (no corresponding method)

Common Methods

method	str	list	tuple	set	dict
clear	n	y	n	y	y
copy	n	y	n	y	y
count	y	y	y	n	n
index	y	y	y	n	n
pop	n	y	n	y	y
remove	n	y	n	y	n
update	n	n	n	y	y

Built-in methods to modify sets

- set.clear()
  clear a set

- set.remove(<elem>)
  remove an element from a set. Raise an exception if it is not in the set

- set.discard(<elem>)
  remove an element from a set. Do nothing if it is not in the set

- set.pop()
  remove a random element from a set. Raise an exception if the set is empty

- set.add(<elem>)
  add a single element to a set; No effect if the element is already in the set

- set1.update(set2[, set3 ...]), same as x1 |= x2 [| x3 ...]
  update a set with the union of itself and others

How to sort data in Python

- Buitl-in method *list.sort([key, reverse])* sort data in the list in place
- Function *sorted(<interable>[, key, reverse])* return a new list containing data from the iterable in ascending order
- When passing the entire dictionary as the iterable to the sorted() function, it returns a list that contains only the sorted keys
- When a key function is given, apply it once to each list item and sort them
- A lambda function is a small anonymous function. Use it when we require a nameless function for a short period of time.
- A lambda function can take any number of arguments, but can only have one expression (no statement is allowed)
- In Python, it is often used as an argument to a higher-order function (a function that takes in other functions as arguments), such as *filter()*

A simple function

- In programming, a function is a self-contained block of code that encapsulates a specific task or related group of tasks.

Define a function in Python:

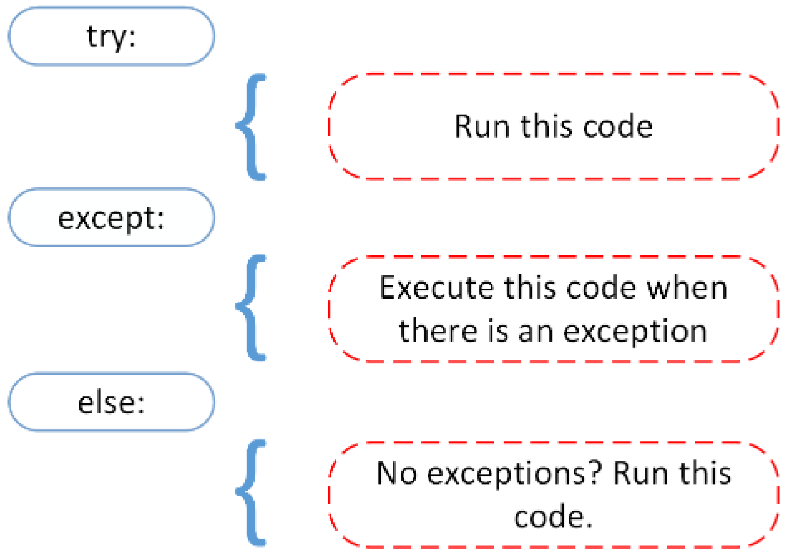
- Use *def* keyword, followed by function name, and parameters
- Docstring – a string comment appearing in the first line after the class or method header
- Statement(s) to do some calculation/action
- Optional *return* statement
- To use a function, we need to know the function’s interface:
  - ✓ What arguments (if any) it takes
  - ✓ What values (if any) it returns

Python special variables

- There are a number of special variables and methods whose name is preceded and followed by __ (two underscores before and after)
- __name__ defines the namespace that a Python module is running in
- When we run the script, the __name__ variable equals __main__
- When we import the containing script, __name__ variable equals the name of the script
- __doc__ prints out the docstring that appears in a class or method

Handling Exceptions: try and except Block

- In Python, an error can be a syntax error or an exception
- The Python interpreter finds any invalid syntax during the parsing stage, the 1st stage
- If your code is free of SyntaxError, you may get other exceptions raised
- The try and except block is used to catch and handle exceptions; Using the else statement, we can execute a block of code only in the absence of exceptions


try:
Run this code
except:
Execute this code when there is an exception
else:
No exceptions? Run this code.

Exercises

1. use set comprehension to construct y_set that only contains negative elements from x_set = {1,-5,-7, 3,-2}

2. Use 2 different ways to change the content of x_set from {1, 2, 3} to {4, 5, 6}

3. Given the list a = [‘apple’, ‘Kiwi’, ‘Orange’]. Generate a list containing all the items in a, sorted in order of increasing string length.

Key takeaways

- Python sets are mutable collections of unordered, unique elements
- Elements contained in a set can be immutable objects of different types
- A lambda function is a small anonymous function that can take any number of arguments, but can only have one expression
- The try and except block is used to catch and handle exceptions