1 Functions

The keyword **def** introduces the function definition. This is followed by the function name and parenthesized list of formal parameters. The first statement of the function can be a string literal which is the function's documentation string, or docstring.

2 Strings

String Methods

string.rstrip(), string.lstrip(), string.strip()
Removes trailing(rstrip), leading(lstrip) or both trailing and leading(strip) whitespace from the string

string.find(pattern[,start[,end]])
Return first position in the string where the pattern occurs in the string or in the slice specified by start and end parameters(when present) and -1 if the pattern does not occur.

string.index(pattern[,l[,r]])
Same as find except that returns ValueError if the pattern is not found.

string.replace(fromstr, tostr[,n])
Return a string with all occurrences or the first n occurrences (when n is given) of fromstr replaced with tostr

string.split(separator[,n])
Return a list of strings obtained from the given string by splitting the string(at most n times when n is given) according to separator

string.join(list)
Return a string obtained by joining elements of 'list' separated by 'string'

string.capitalize()
Return a string with the first letter of the given string capitalized

string.lower()
Return the given string with all letters lowercase

string.upper()
Return the given string with all letters uppercase

3 Lists

List Methods

list.append(x)
Add an item to the end of list. Equivalent to a[len(a):] = [x]
list.extend(iterable)
Append all the items from the iterable. Equivalent to a[len(a):] = iterable
list.insert(i,x)
Insert item x at a position before the element at index i.
list.remove(x)
Remove the first item from the list whose value is equal to x and raise ValueError if x is not in the list
list.pop([i])
Remove the element at position i in the list if given or remove the last element in the list and return the removed element. The parameter i is optional indicated by the surrounding square brackets
list.clear()
Remove all items from the list. Equivalent to del a[:]
list.index(x,[start,[end]])
Return the zero-based index of the first item whose value is equal to x. If start or both start and end parameters are present, then limit the search to the resulting slice but report the index starting from the beginning of the list. Raise ValueError if item x is not present in the slice.
list.count(x)
Return the number of times x occurs in the list.
list.sort(key=none, reverse=False)
Sort the elements in the list in place
list.reverse()
Reverse the elements of the list in place
list.copy()
Returns a shallow copy of the list. Doesn't work in Python 2.7

map() and filter() are useful functions for manipulating lists

List Comprehensions

List comprehensions provide a concise way to create lists. It consists of bracket containing an expression followed by a for clause, then zero or more for or if clauses. The result is a new list resulting from evaluating the expression in the context of the following for and if clauses. It provides a useful notation for combining map and filter.

4 Dictionaries

Dictionaries

A dictionary is a collection of distinct pairs of items where the first item in the pair is called key and the second item in the pair is called value. The keys are all distinct within a dictionary and are used to index the dictionary. The keys must be immutable values. Dictionaries are extremely useful for manipulating information from text files, tables

Operations on Dictionaries

dictionary.keys()
Returns a sequence of all the keys in the dictionary
dictionary.values()
Returns a sequence of all the values in the dictionary
Key Membership
key **in** Dictionary