Python Cheat Sheet

by Aiman Fayyaz via cheatography.com/159132/cs/33566/

List Methods

append(value)	adds a value at the end of the list
clear()	empties the list
copy()	creates a copy of the list
count(value)	counts the number of times a value appears in the list
extend(iterable)	adds the elements of an iterable at the end of the list
index(value)	returns the index where the specified value first occurs
insert(index, value)	inserts the new value at the specified index
pop(index)	removes the element at the specified index
pop()	removes the last value from the list
remove(value)	removes the first occurrence of the specified value
reverse()	reverses the order of the list
sort()	sorts the list values in ascending order(default)
sort(reverse = True)	sorts the list in descending order

Random Module

randint(a,b)	returns a random integer between a and b(both inclusive)
choice(iterable)	returns a random element from the given iterable
choices(iterable)	returns a list containing a random element from the given iterable
random()	returns a random floating point number between 0.0 and 1.0(both exclusive)
uniform(a,b)	returns a random floating point number between a and b(both inclusive)
shuffle(iterable)	changes the original sequence of the iterable randomly
sample(iterable, **k**=n)	returns a sample of size k(k<=len) from the iterable

Tuple Methods

count(element)	returns the number of times an element occurs in a tuple
index(element)	returns the index of the first occurrence of the specified element

Built-in Function

ord(char)	returns the unicode value of a character
chr(unicode)	returns the character that represents the unicode
bin(n)	converts n into its binary equivalent and returns it as a string
oct(n)	converts n into its octal equivalent
hex(n)	converts n into its hexadecimal equivalent
sum(iterable,start(optional))	sums all the values of an iterable, start is an optional value to be added to the sum
dir(object)	lists all the attributes of an object

Dictionary Methods

clear()	removes all the dictionary elements
copy()	creates a copy of the dictionary
get(key)	gets the value of the specified key
keys()	returns a list of all the keys in the dictionary
values()	returns a list of all the values in the dictionary
items()	returns a list of all elements in the dictionary as tuples of key-value pairs
pop(key)	removes the specified key and its value
popitem()	removes the last inserted key and its value
dict.fromkeys	creates a dictionary with the specified keys and value(optional), default is none
setdefault(key-value pair)	returns the value of the specified key, inserts the key if it does not already exist
update(key-value pairs/dictionary)	updates the dictionary with the given values, inserts keys if they do not already exist

Set Methods

add(element)	adds an element to the set
clear()	empties the entire set
copy()	creates a copy of the list
x.difference(iterable(s))	returns a new set containing all the elements of set x that are not present in the other iterables
x.difference_update(iterable(s))	removes all elements from set x that are present in other iterables
pop()	removes any element from the set and returns it
discard(element)	removes a specified element from the set, does not raise an error if element not found
remove(element)	removes a specified element from the set, raises an error if element not found
intersection(iterable(s))	returns a new set containing those elements that are common to all iterables
x.intersection_update(iterable(s))	keeps only those elements in set x that are common to all iterables and removes the rest

Set Methods (cont)

isdisjoint(iterable(s))	returns False if the iterables have at least one common element, otherwise True
x.issubset(y)	returns True if set x is a subset of y(any iterable)
x.issuperset(y)	returns True if x is a superset of y(any iterable)
x.union(iterable(s))	returns a set containing elements present in all the iterables, duplicates are excluded
x.update(iterable(s))	updates the set x by adding all the elements from the other iterables, duplicates are excluded
union(iterable(s))	returns a set which is the union of all the iterables
update(iterable(s))	updates the set with the union of all iterables