**LIST**

Join two lists

>>> [1]+[2,3]

[1,2,3]

Index out of boundary

>>> test = [1,2,3]

>>> test[4:]

[]

Reverse a list

list[::-1]

list.reverse()

whether inverse value is the same

all(val==val[::-1] for val in integers)

Join elements by ‘-‘ in the list

'-'.join(temp)

Print two lines

>>> print('a\nb')

a

b

Convert string/input to int

arr = list(map(int, input().split()))

>>> list(map(int,['2','3']))

[2, 3]

Insert value in list (index,value)

>>> test = [1,3]

>>> test.insert(2,4)

>>> test.insert(10,4)

>>> test

[1,3,4,4]

Count value in list

>>> test.count(1)

1

Get index in list

>>> test.index(3)

0

Append a list to a list

list.extend(seq)

Pop the last item / we can also identify value by index

>>> test

[1, 3, 1]

>>> test.pop()

1

Delete elements based on index

del list[2]

Remove one element in the list / remove the first one

>>> test = [1,2,3]

>>> test.remove(2)

>>> test

[1, 3]

Looping till all 1's are removed

while (list1.count(1)):

list1.remove(1)

**TUPLE**

make a tuple

1. tuple(a\_list)

2. (1,2,3)

**STRING**

Uppercase the first letter and lowercase the rest

Str.title()

str.capitalize()

get 26 letters

>>> import string

>>> string.ascii\_lowercase

'abcdefghijklmnopqrstuvwxyz'

Fill with --

Str.center(width, fillchar)

>>> 'sdfsadf'.center(15,'-')

'----sdfsadf----'

Split by ‘ ’

‘xxxx’.split()

Split word to characters

List(str)

Reverse string

'abc'[::-1]

Check alpha and digit

Str.isalnum()

Str.[isalnum()](https://www.tutorialspoint.com/python/string_isalnum.htm)

Str.[isdigit()](https://www.tutorialspoint.com/python/string_isdigit.htm)

remove space

str.strip()

cannot use index to change elements in a string

**DICTIONARY**

dict.update(dict2)

unlike use dict[key], when using get no error

dict.get(key, default = None)

del dict['Name']; # remove entry with key 'Name'

# remove all entries in dict

dict.clear()

# delete entire dictionary

del dict

dict.keys()

dict.values()

dict.items()

dict.has\_key('Age')

**NUMBER**

math.ceil( x )

math.floor( x )

>>> divmod(5,2)

(2, 1)

**MAP**

Instead of using loop

1.

# Return double of n

def addition(n):

    return n + n

# We double all numbers using map()

numbers = (1, 2, 3, 4)

result = map(addition, numbers)

print(list(result))

2.

numbers1 = [1, 2, 3]

numbers2 = [4, 5, 6]

result = map(lambda x, y: x + y, numbers1, numbers2)

print(list(result))

**SET**

>> a = {2, 4, 5, 9}

>> b = {2, 4, 11, 12}

>> a.union(b) # Values which exist in a or b

{2, 4, 5, 9, 11, 12}

>> a.intersection(b) # Values which exist in a and b

{2, 4}

>> a.difference(b) # Values which exist in a but not in b

{9, 5}

>>> s = set('HackerRank')

>>> s.add('H')

>>> print s

set(['a', 'c', 'e', 'H', 'k', 'n', 'r', 'R'])

s.remove(value)

subset

x = {"a", "b", "c"}

y = {"f", "e", "d", "c", "b", "a"}

x.issubset(y)

True

import collections

a = [1,1,1,1,2,2,2,2,3,3,4,5,5]

counter=collections.Counter(a)