Lists in Python

a = [1, 2, 3]

Can be in different data types

a = ["ML", 7]

Elements can be deleted

a = [1, 2, 3]

del a[0]

print(a) # output: [2, 3]

Length of the list

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

n = len(a)

print(n) #output: 5

Sort

a = [8, 7, 4, 5]

a.sort()

print(a) # output: [4, 5, 7, 8]

Reverse

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

a.reverse()

print(a) # output: [5, 4, 3, 2, 1]

Checking whether an element is in the list

a = [1, 3, 5, 7, 9]

print(1 in a)

print(1 not in a)

print(2 in a)

If you have a list l1, then the following assignment: l2 = l1 does not make a copy of the l1 list, but makes the variables l1 and l2 point to one and the same list in memory. (aka reference in C++)

list\_1 = [1]

list\_2 = list\_1

list\_1[0] = 2

print(list\_2) # output: [2]

How to solve this? By **list slicing**

list\_1 = [1]

list\_2 = list\_1[:]

list\_1[0] = 2 print(list\_2) # the output = [1]

my\_list = [10, 8, 6, 4, 2]

new\_list = my\_list[1:3] # the output = [8, 6]

Initializing

a = [i for i in range(8)]

print(a) # output: [0, 1, 2, 3, 4, 5, 6, 7]

square = [x \*\* 2 for x in range(10)]

print(square) # output: [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

pow2 = [2 \*\* i for i in range(8)]

print(pow2) # output: [1, 2, 4, 8, 16, 32, 64, 128]

odds = [i for i in a if i % 2 != 0] # from list 'a'

print(odds) # output: [1, 3, 5, 7]

2D-list and initializing

a = [[i for i in range(3)] for j in range(3)]

print(a) # output: [[0, 1, 2], [0, 1, 2], [0, 1, 2]]