*Python List Basics Study Sheet Last Update 7 Sep ‘23*

list item properties:

are ordered, are mutable, can be different data types

declare a list: my\_list = [‘a’, ‘b’, ‘c’]

cast a set/tuple to list: my\_list = list(my\_set)

write the membership operator and what it returns:

*element* in *my\_list* #returns True or False

locate where ‘e’ is in a list, starting from 3rd element:

my\_list.index(‘e’, 3) #finds first instance only, error if not found

get last index: myList[-1]

given: another\_list = my\_list

does this clone the list? no- my\_list and another\_list refer to same obj

Cloning a list (two methods): B = A[:] *or* my\_list.copy()

*given: myList = [‘a’, ‘b’, ‘c’, ‘d’]*

What do each of the following operations do (carry over changes)?

myList[2] = ‘g’ [a, b, g, d] (reassignment)

myList[1:2] = [‘x’, ‘y’] [a, x, y, g, d] (fits all elements into slice)

myList[1:4] = [‘s’] [a, s, d] (still removes the 1:4 slice)

*given: myList = [‘a’, ‘b’, ‘c’], write code for the following ops*

Add ‘e’ after ‘b’: myList.insert(2, ‘e’)

Add ‘k’ at the end: myList.append(‘k’)

Add elements of other iter: myList.extend(otherList)

create new list from elements of L1 and L2: my\_list = L1 + L2

Remove ‘c’ from list: myList.remove(‘c’) #finds first instance

Remove ‘s’ from list: Illegal; causes error if ‘s’ isn’t in list!

*write the state of myList = [a, b, c] after the following ops*

myList.append([d, e]): [a, b, c, [d, e]]

myList.extend([f, g]): [a, b, c, [d, e], f, g]

Remove last item: myList.pop() / del myList[-1]

Remove first item: myList.pop(0) / del myList[0]

delete specified index: del myList[index]

delete specified slice: del myList[slice start: slice end]

what does the pop method return? the value of the popped element

*given: myList = [1, 2, 3, 4], write the value of each var after:*

a, b, c = myList error; disagreement btwn elements and vars

\*a, b, c = myList a = [1, 2], b = 3, c = 4

a, \*b, c = myList a = 1, b = [2, 3], c = 4

a, b, \*c = myList a = 1, b = 2, c = [3, 4]

\*a, b, \*c = myList error; can only have one starred var

a, b, \*c, d, e = myList a = 1, b = 2, c = [], d = 3, e = 4

Turn list into set: set(myList) #good for removing duplicates

Find length of list: len(myList)

Sum of all elements: sum(myList)

Clear entire list: myList.clear()

Loop over a list backwards (two ways):

item in ( reversed(myList) *or* myList[::-1] )

Actually reverse the list: myList.reverse()

Print every other element starting at index 1: for item in my\_list[1::2]:

print(item)

Sort list in-place: myList.sort()

Make new sorted list, reverse order: L1 = sorted(myList, reverse=True)

*given myList = [7, 5, 3, 2]*

give the sum of elements using reduce: reduce(lambda x, y: x+y, myList)

output of: list(accumulate(myList, lambda x, y: x-y)): [7, 2, -1, -3]