Questions

**Not yet graded**

seqmut-1-5: Could aliasing cause potential confusion in this problem?

b = ['q', 'u', 'i']

z = b

b[1] = 'i'

z.remove('i')

print(z)

Top of Form

A. yes  
B. no

✔️ Yes, b and z reference the same list and changes are made using both aliases.Bottom of Form

seqmut-1-6: Could aliasing cause potential confusion in this problem?

sent = "Holidays can be a fun time when you have good company!"

phrase = sent

phrase = phrase + " Holidays can also be fun on your own!"

Top of Form

A. yes  
B. no

Bottom of Form

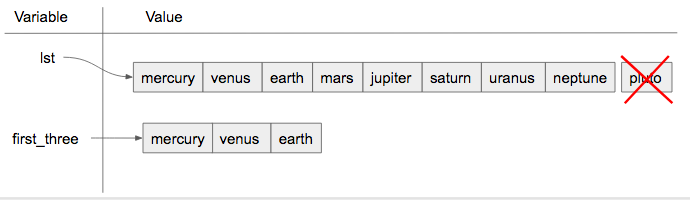
✔️ Since a string is immutable, aliasing won't be as confusing. Beware of using something like item = item + new\_item with mutable objects though because it creates a new object. However, when we use += then that doesn't happen.

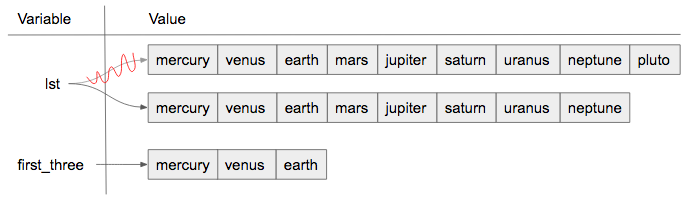
seqmut-1-1: Which of these is a correct reference diagram following the execution of the following code?

lst = ['mercury', 'venus', 'earth', 'mars', 'jupiter', 'saturn', 'uranus', 'neptune', 'pluto']

lst.remove('pluto')

first\_three = lst[:3]





Top of Form

A. I.  
B. II.  
C. Neither is the correct reference diagram.  
Check meCompare me

Bottom of Form

✔️ Yes, when we are using the remove method, we are just editing the existing list, not making a new copy.

seqmut-1-7: Which of these is a correct reference diagram following the execution of the following code?

x = ["dogs", "cats", "birds", "reptiles"]

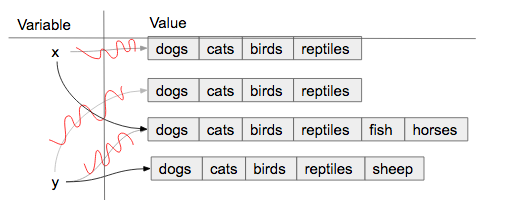
y = x

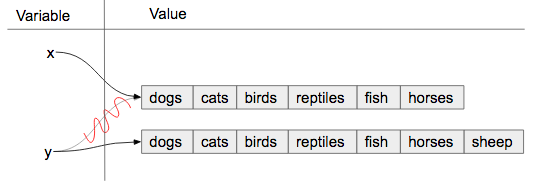
x += ['fish', 'horses']

y = y + ['sheep']









Top of Form

A. I.  
B. II.  
C. III.  
D. IV.

Bottom of Form

✔️ Yes, the behavior of obj = obj + object\_two is different than obj += object\_two when obj is a list. The first version makes a new object entirely and reassigns to obj. The second version changes the original object so that the contents of object\_two are added to the end of the first.

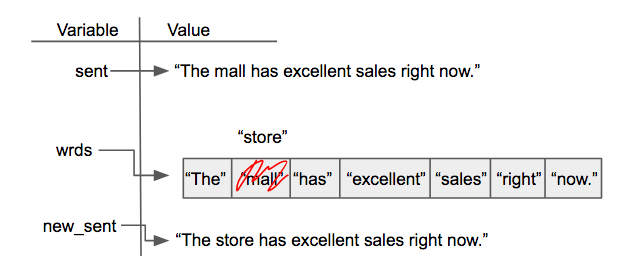
seqmut-1-8: Which of these is a correct reference diagram following the execution of the following code?

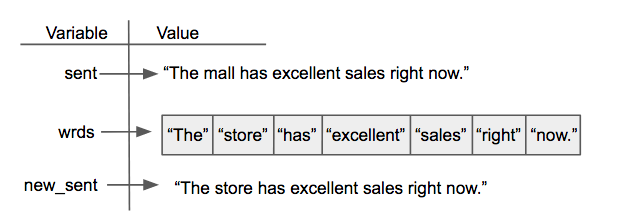
sent = "The mall has excellent sales right now."

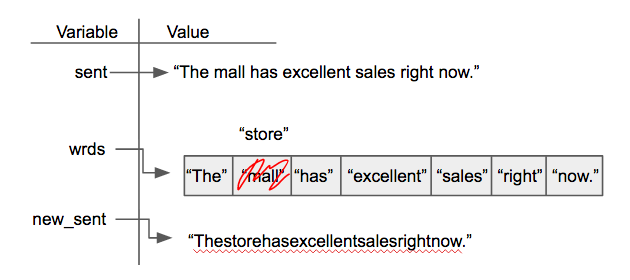
wrds = sent.split()

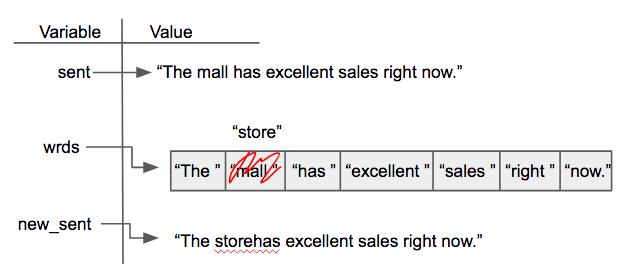
wrds[1] = 'store'

new\_sent = " ".join(wrds)









Top of Form

A. I.  
B. II.  
C. III.  
D. IV.

Bottom of Form

✔️ Yes, when we make our own diagrams we want to keep the old information because sometimes other variables depend on them. It can get cluttered though if there is a lot of information.