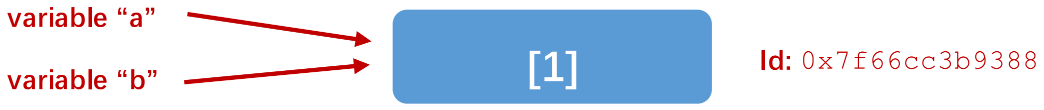
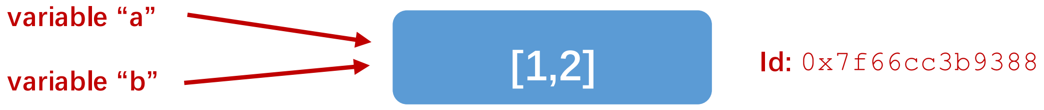
(Please correct me if I wrote something wrong or missed some points)

We know that for Python, everything is an object. A unique object id is assigned to an object when the object is initialized and it can never change during this object's lifetime. However, for mutable object (e.g. dict, list, set), their states can be changed after they are created (e.g. append), while immutable objects (e.g. int, float, str) don't allow any modification after creation.

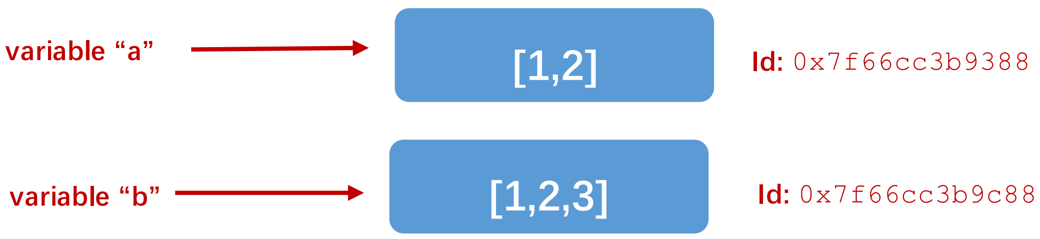
The object type for Q5 that we want to discuss about is mutable object (i.e. list). We can see that initially, we create a varaible named "a" to hold the "[1]" object instance. In other words, we use "a" to refer to the newly created list "[1]". Then we passes the reference to the function "foo" so that now the variable "b" also refers to the list "[1]". We can verify that by looking at the Ids for initial "a" and "b", which are both "`0x7f66cc3b9388`".



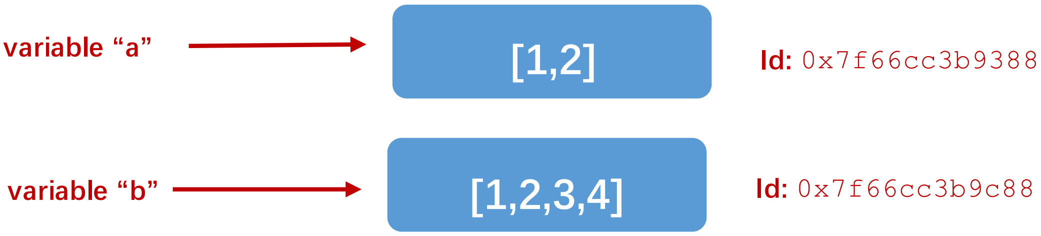
After that, we append 2 to "b". Since we are only changing the state of the existing mutable object, no new object will be created. In other words, we are modifying the original object "[1]" to "[1,2]", while "a" and "b" still refer to this original object. Of course the id remains the same.



Then, we do "b = b + [3]". This time it's a bit different because we are doing re-assignment for variable "b". It means that we are not modifying the original one but actually creating a new object "[1,2,3]" and let "b" refer to that new object. We can verify this point by looking that the Ids for "a" and "b" after this statement, which are different. Variable "a" still refer to the original object while "b" refer to the new object. (However, if we use "b += [3]", it is a modification on the original object so no new object will be created. "a" and "b" both refer to the original object "[1,2,3]".)



Then, we do "b.append(4)". The same as above, we know that it is a simple modification on the existing object that is referred by variable "b" and no new object will be created. Since now "a" and "b" refer to different objects, this modification has no influence on the object "[1,2]" that is referred by variable "a". The figure of final state is shown below:


variable “a”
[1,2]
Id: 0x7f66cc3b9388
variable “b”
[1,2,3,4]
Id: 0x7f66cc3b9c88