9.5. Aliasing

Since variables refer to objects, if we assign one variable to another, both variables refer to the same object:

Save & Run 5/13/2021, 1:59:12 PM - 2 of 2 Show in CodeLens

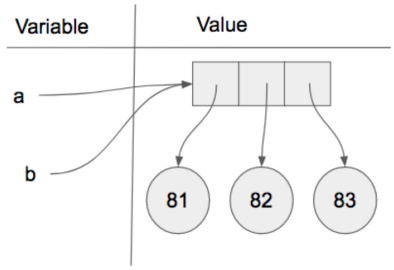
```
a = [81, 82, 83]
b = a
print(a is b)

```

```
True
```

Activity: 1 -- ActiveCode (ac8_4_1)

In this case, the reference diagram looks like this:



Because the same list has two different names, `a` and `b`, we say that it is **aliased**. Changes made with one alias affect the other. In the codelens example below, you can see that `a` and `b` refer to the same list after executing the assignment statement `b = a`.

Save & Run 5/13/2021, 1:59:16 PM - 2 of 2 Show in CodeLens

```
a = [81,82,83]
b = [81,82,83]
print(a is b)

b = a
print(a == b)
print(a is b)

b[0] = 5
print(a)

```

```
False
True
True
[5, 82, 83]
```

Activity: 2 -- ActiveCode (ac8_4_2)

Although this behavior can be useful, it is sometimes unexpected or undesirable. In general, it is safer to avoid aliasing when you are working with mutable objects. Of course, for immutable objects, there's no problem. That's why Python is free to alias strings and integers when it sees an opportunity to economize.

Check your understanding

seqmut-4-1: What is the value of y after the following code has been evaluated:

```
w = ['Jamboree', 'get-together', 'party']
y = ['celebration']
y = w
```

- (●) A. ['Jamboree', 'get-together', 'party']
- ( ) B. ['celebration']

C. ['celebration', 'Jamboree', 'get-together', 'party']

D. ['Jamboree', 'get-together', 'party', 'celebration']

Check me Compare me

✔ Yes, the value of y has been reassigned to the value of w.

Activity: 3 -- Multiple Choice (question8_1_3)

seqmut-4-2: What is printed by the following statements?

```
alist = [4,2,8,6,5]
blist = alist
blist[3] = 999
print(alist)
```

A. [4,2,8,6,5]

B. [4,2,8,999,5]

Check me Compare me

✔ Yes, since alist and blist both reference the same list, changes to one also change the other.

Activity: 4 -- Multiple Choice (question8_4_1)

You have attempted 5 of 4 activities on this page

9.4. Objects and References">

9.4. Objects and References">

✔ Completed. Well Done!

9.6. Cloning Lists">

9.6. Cloning Lists">Next Section - 9.6. Cloning Lists

| Back to top