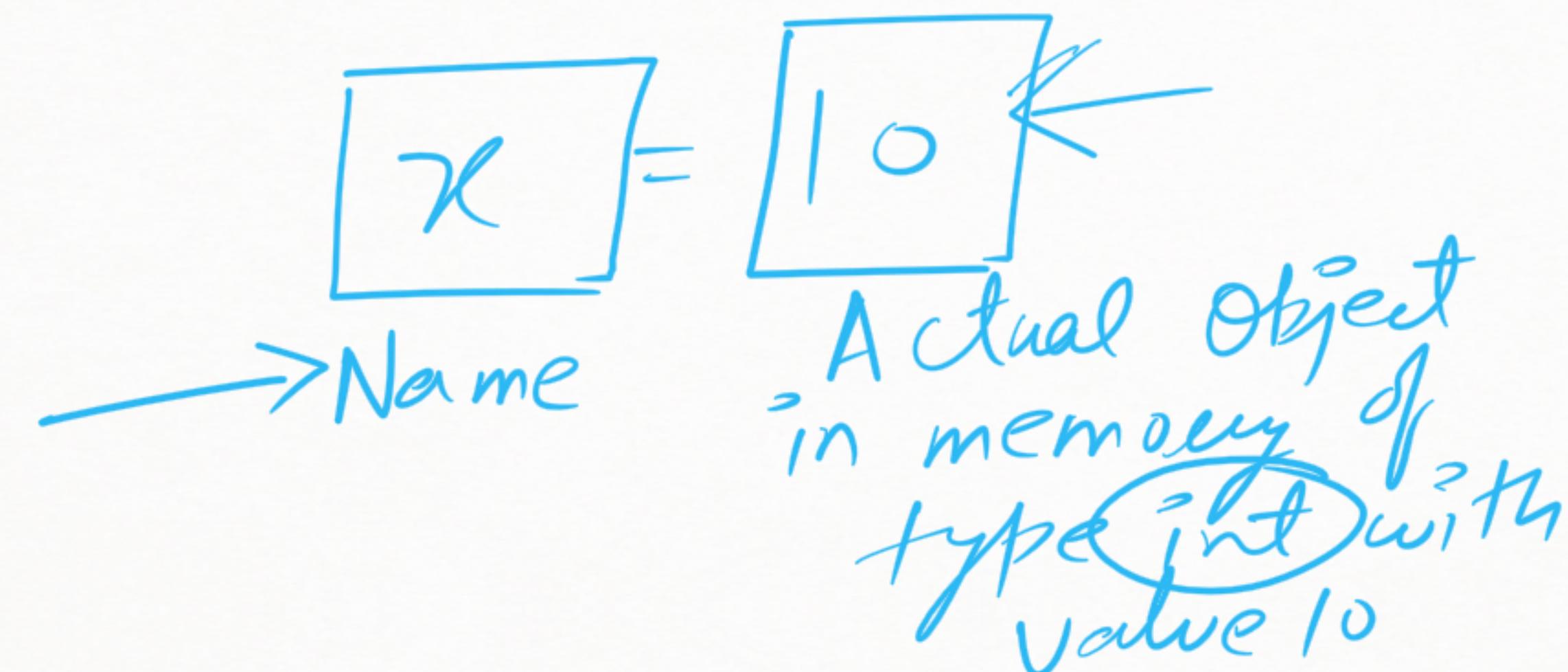


Assignment: Assigning



- # What is an Object?
- Everything in Python is Object
- All the data types are Object

$x = 10$
y = 4
Assignment

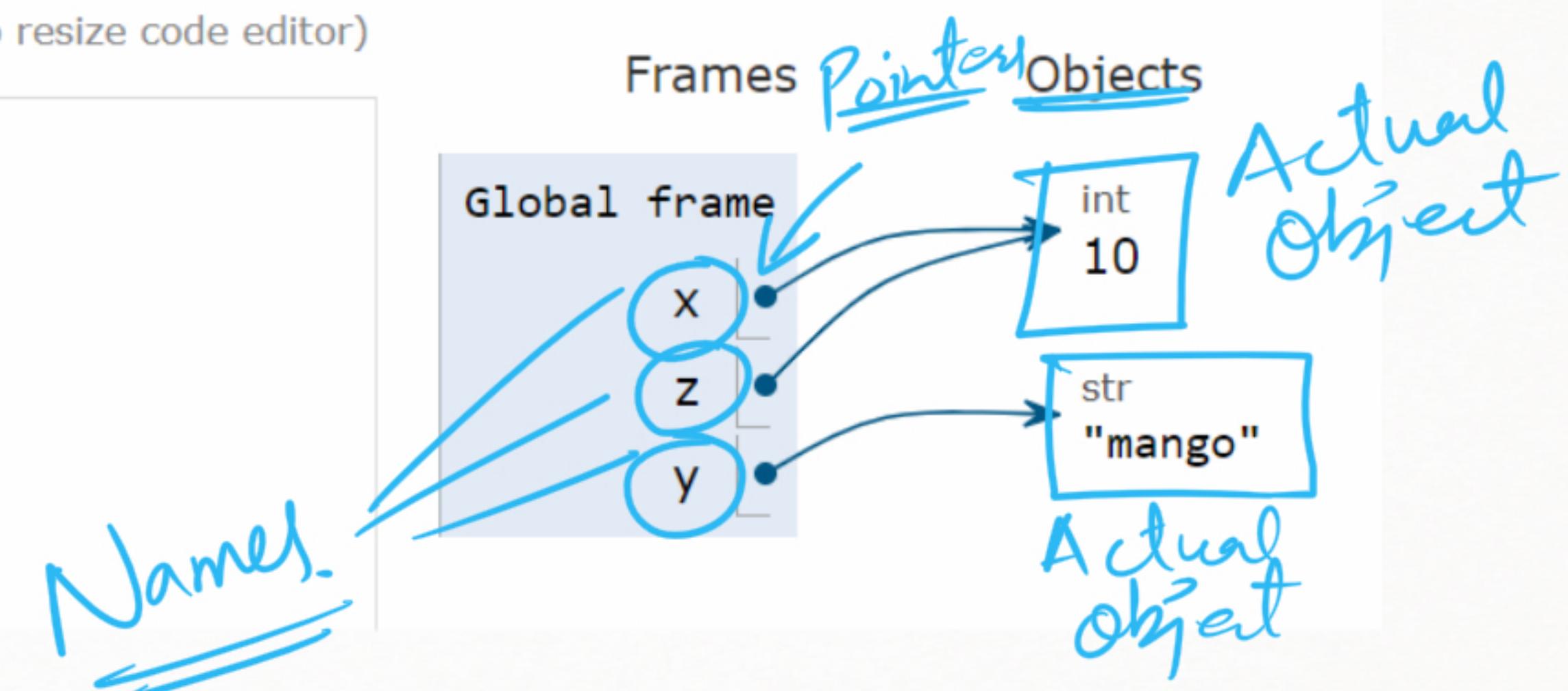
y Name has been given to an object of int data type with value 4.

ode in Python 3.6

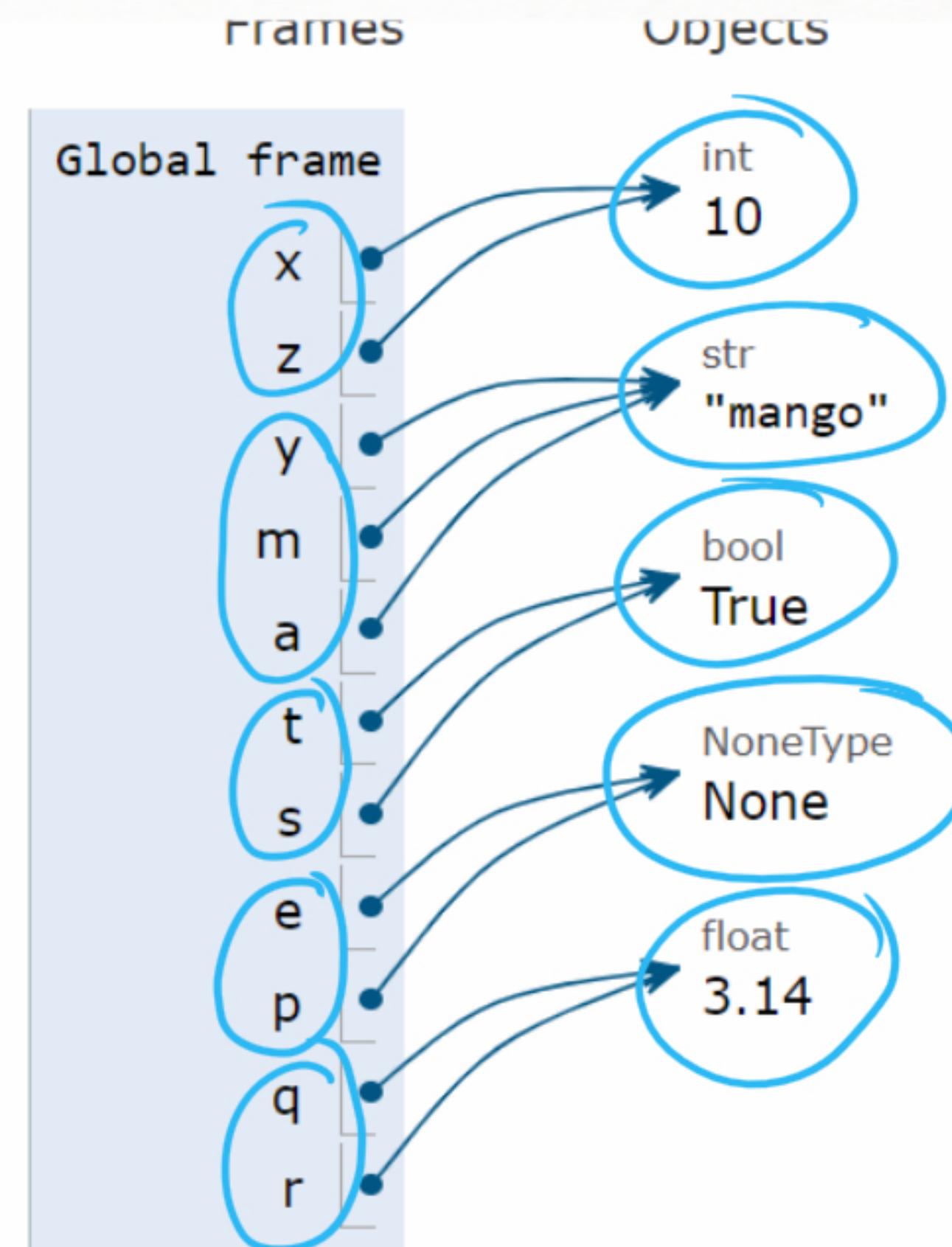
```
x = 10 ✓  
z = 10 ✓  
y = 'mango'
```

(drag lower right corner to resize code editor)

Non Collection-Datatypes:
→ int → bool → str
→ float → None



```
x = 10  
z = 10  
y = 'mango'  
m = 'mango'  
a = 'mango'  
t = True  
s = True  
e = None  
p = None  
q = 3.14  
r = 3.14
```

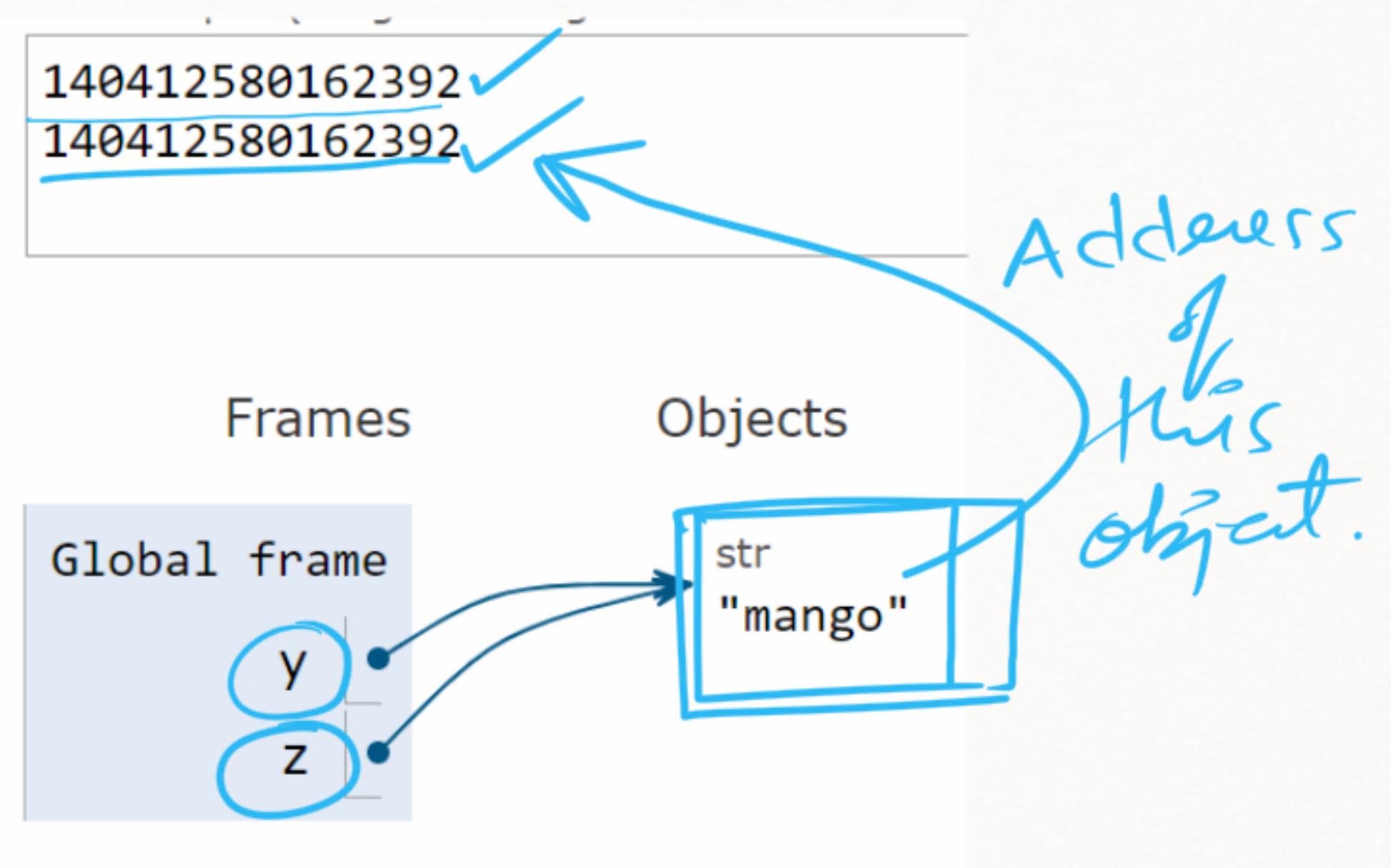


Code in | Python 3.6

```
y = 'mango'  
z = 'mango'  
  
print(id(y))  
print(id(z))
```

Address
of the object
which Name y
is pointing

(drag lower right corner to resize code editor)

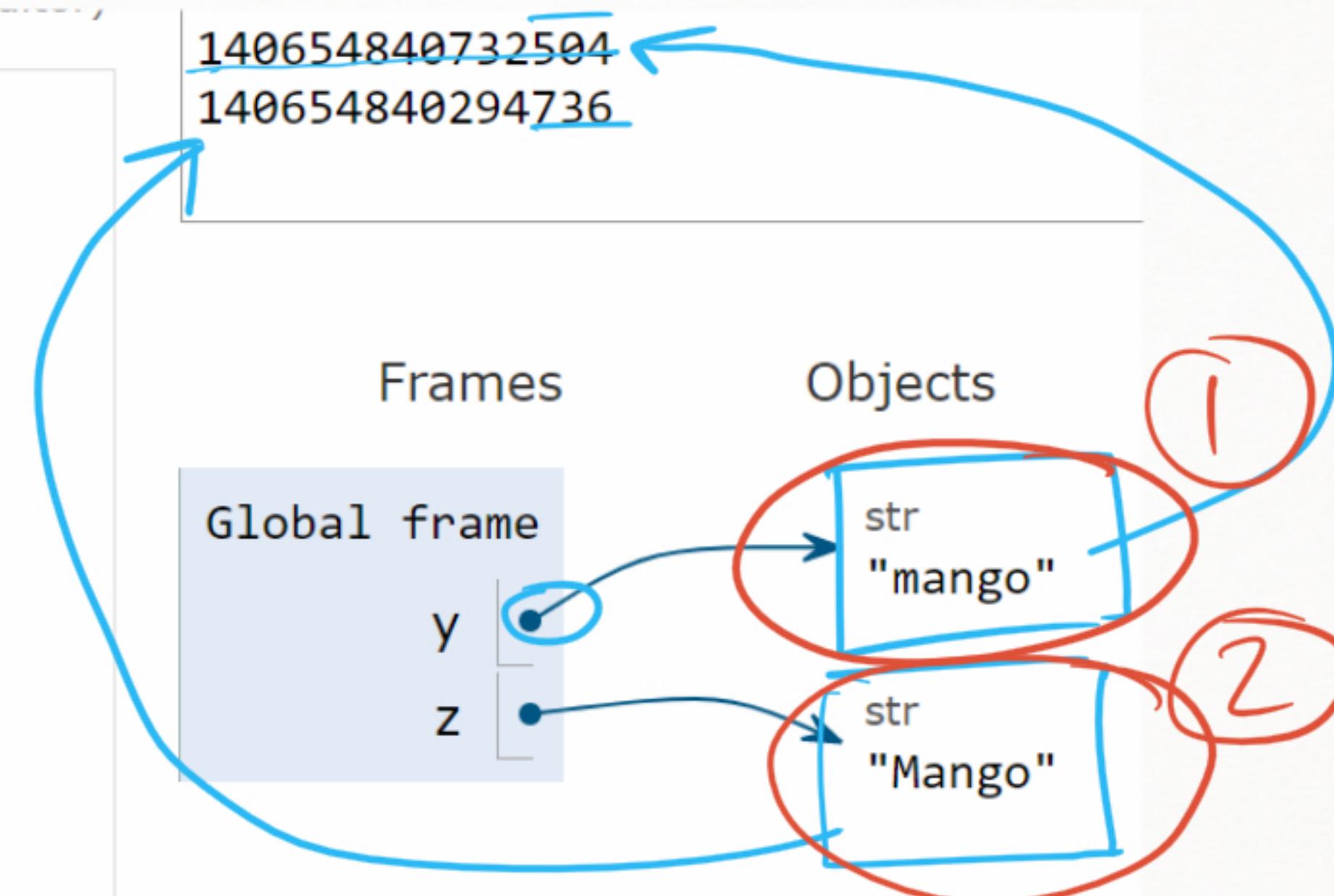


```
Code in Python 3.6
```

```
y = 'mango'  
z = 'Mango'
```

```
print(id(y))  
print(id(z))
```

How many
are created in memory?
object
⇒ 2



```
x = ['python', 24, True, False, None, 3.14]
```

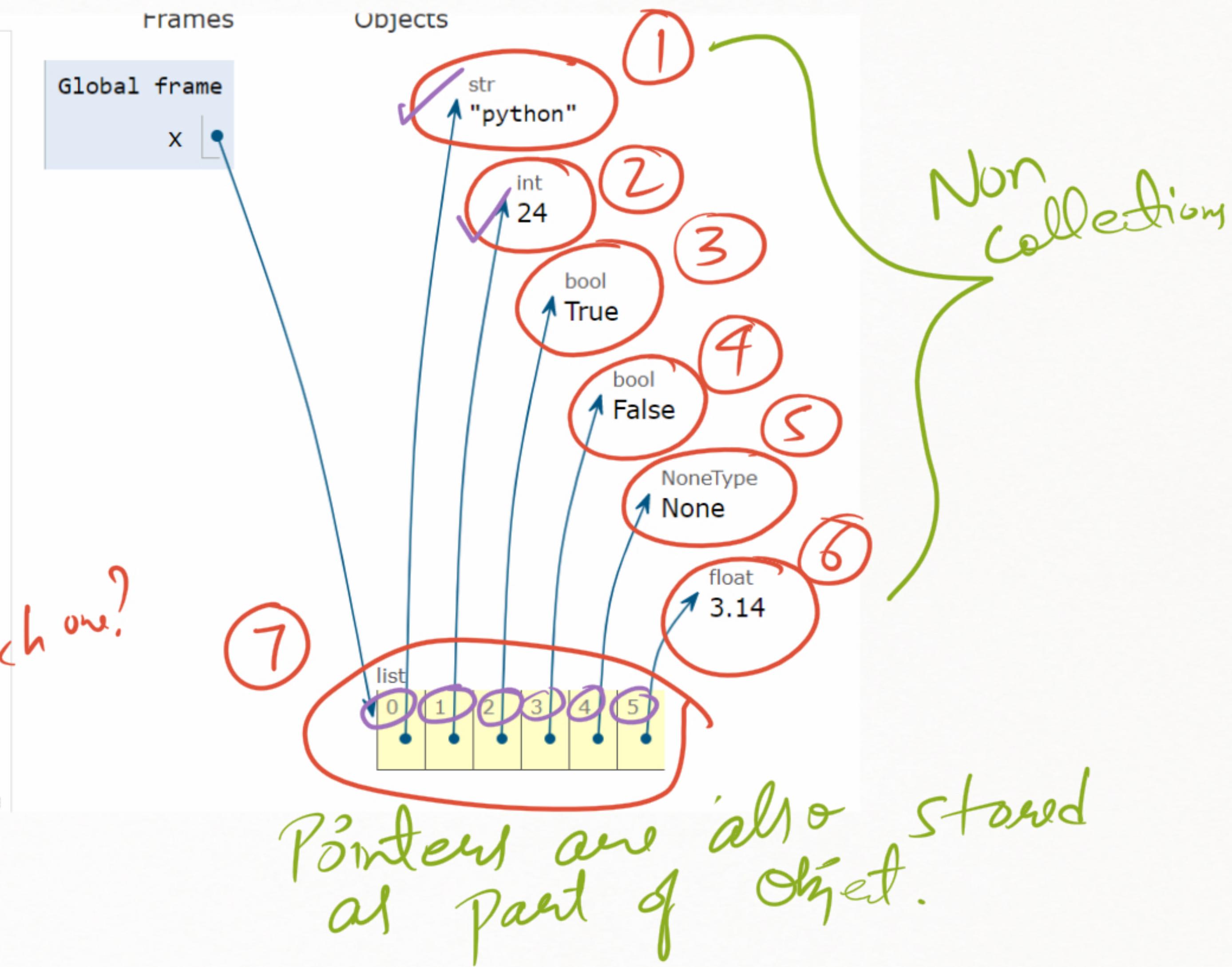
How many objects
are created here?

⇒ 7

How many Assignments

⇒ 1

Names & which one?
How many
⇒ 1 (x)



```
x = {'python':3,'java':8}
```

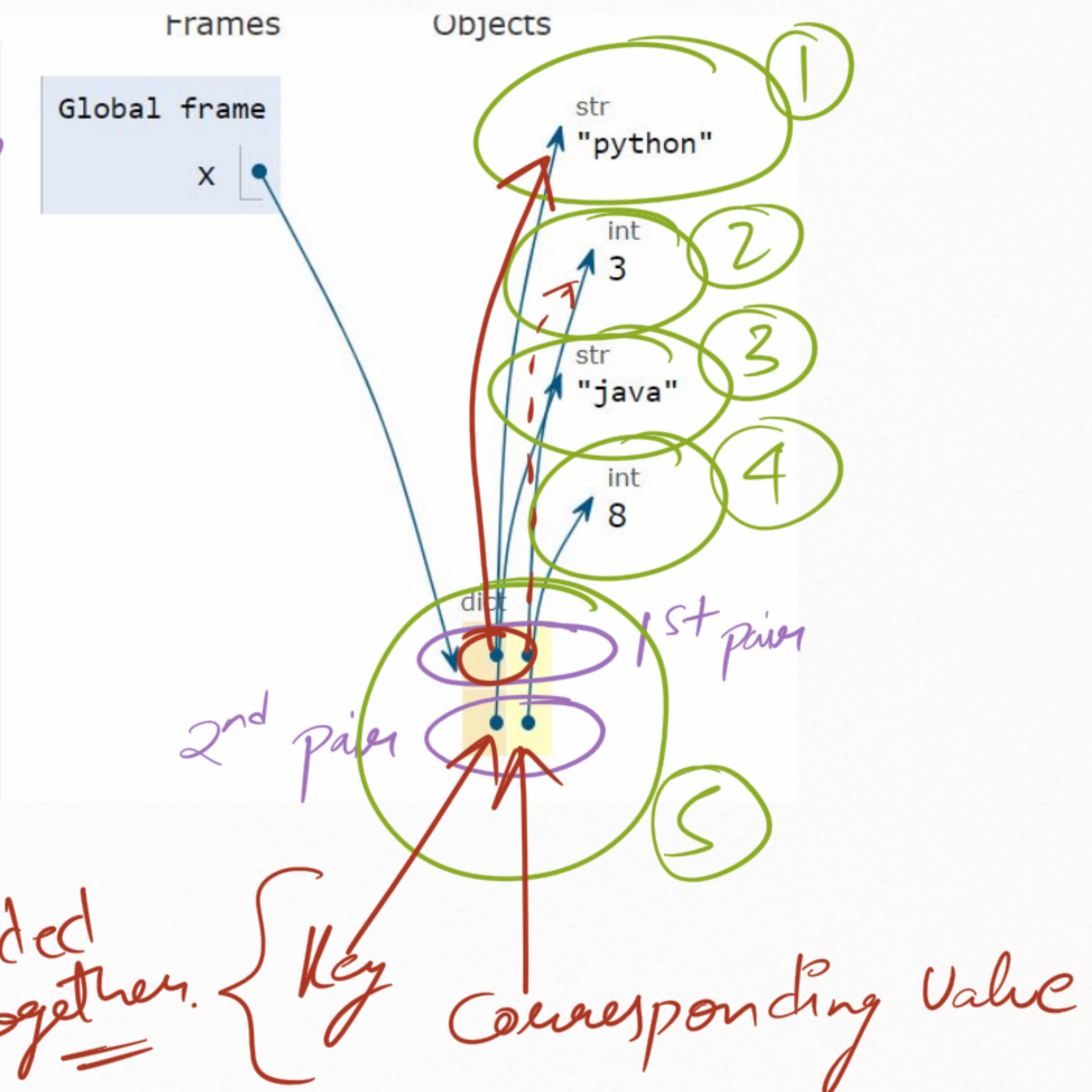
How many Key:Value pairs
are here?

⇒ 2

How many objects are here?

⇒ 5

No Indexing



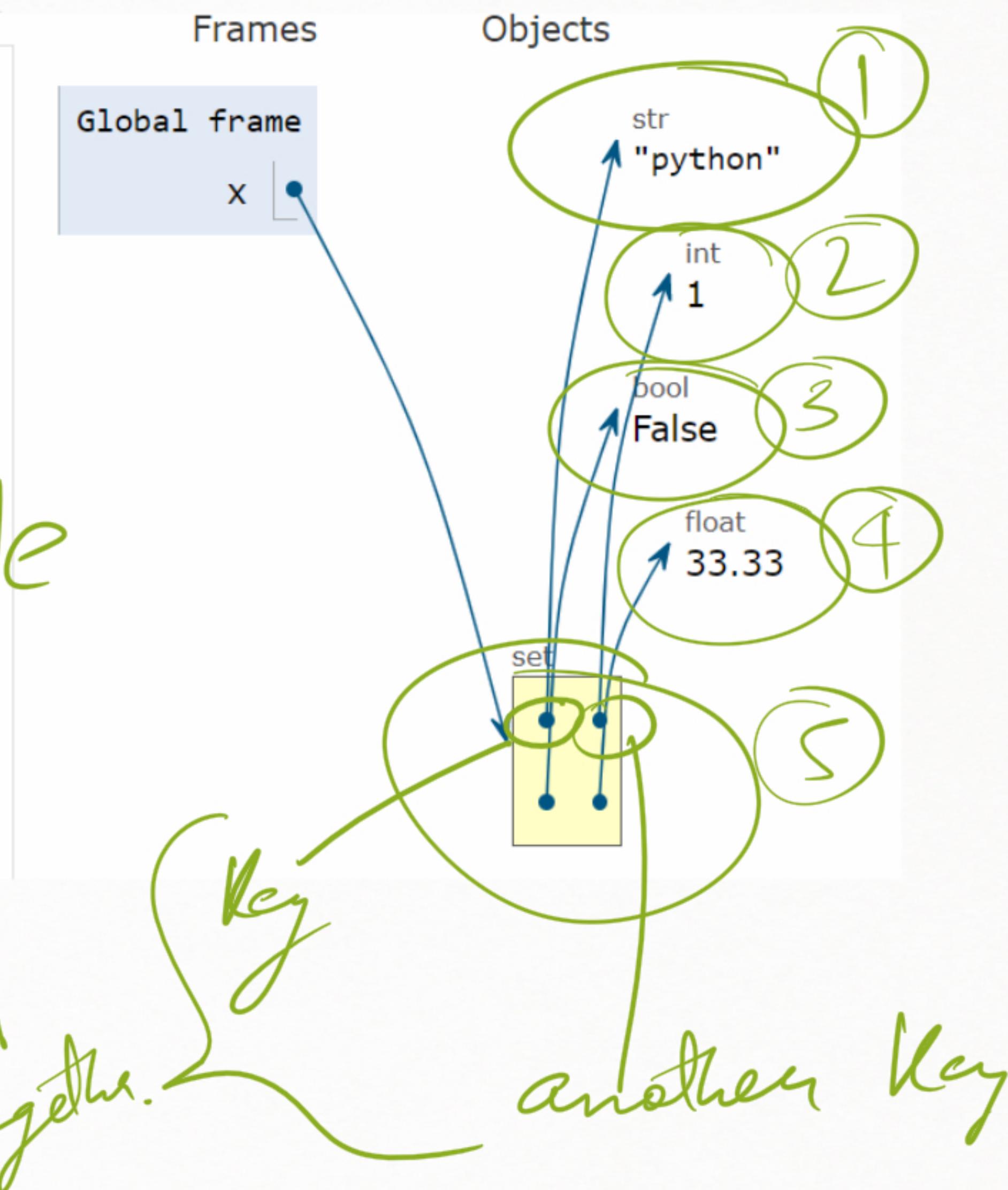
```
x = {'python', 'python', 1, 1, 1, 1, 1, 1, False, 33.33}
```

How many objects are here?

⇒ 5

⇒ No indexing possible

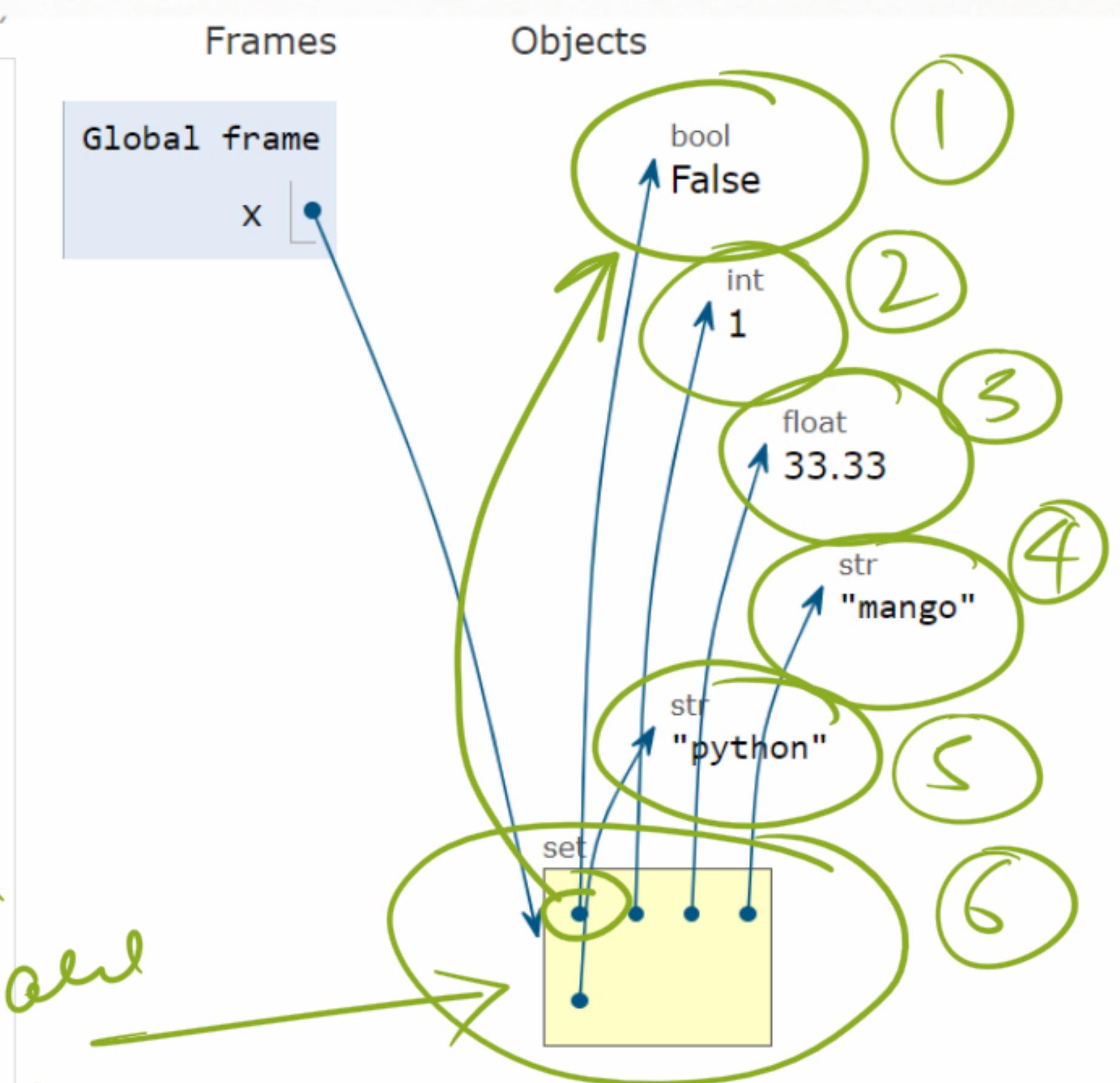
Not
Binded
Together.



```
x = {'python', 'python', 1, 1, 1, 1, 1, 1, False, 33.33, 'mango'}
```

Sets are
Unordered.

Index
positions are
not here

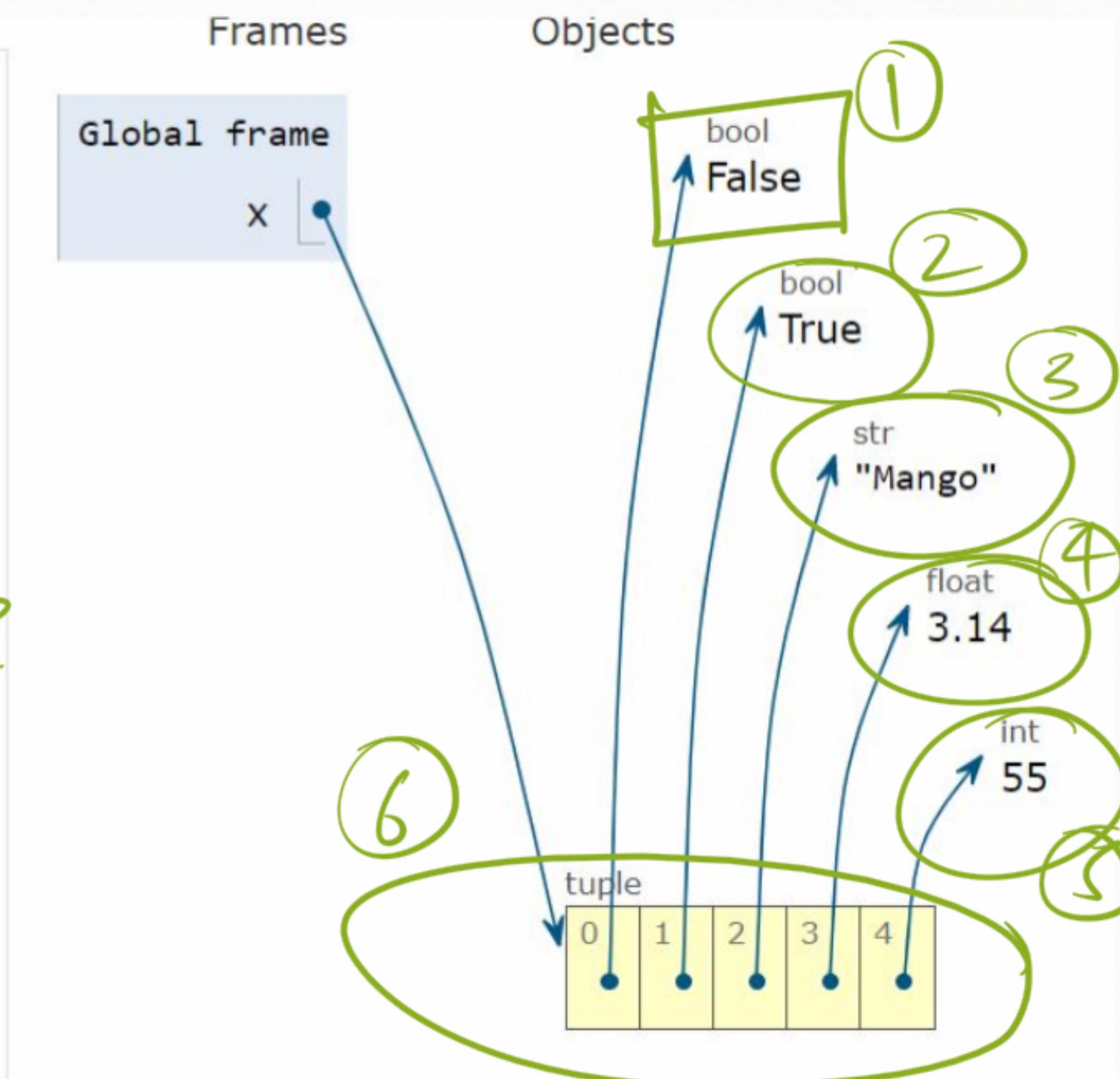


```
x = (False, True, 'Mango', 3.14, 55)
```

How many objects are here?
⇒ 6

How many immutable objects are here?

⇒ all 6



```
x = ('mango','banana','pyton')
x[2] = 'java'
```

```
-----  
TypeError                                     Traceback (most recent call last)
<ipython-input-9-9337ed9677dc> in <module>
      1 x = ('mango','banana','pyton')
----> 2 x[2] = 'java'

TypeError: 'tuple' object does not support item assignment
```

Tuples are immutable

```

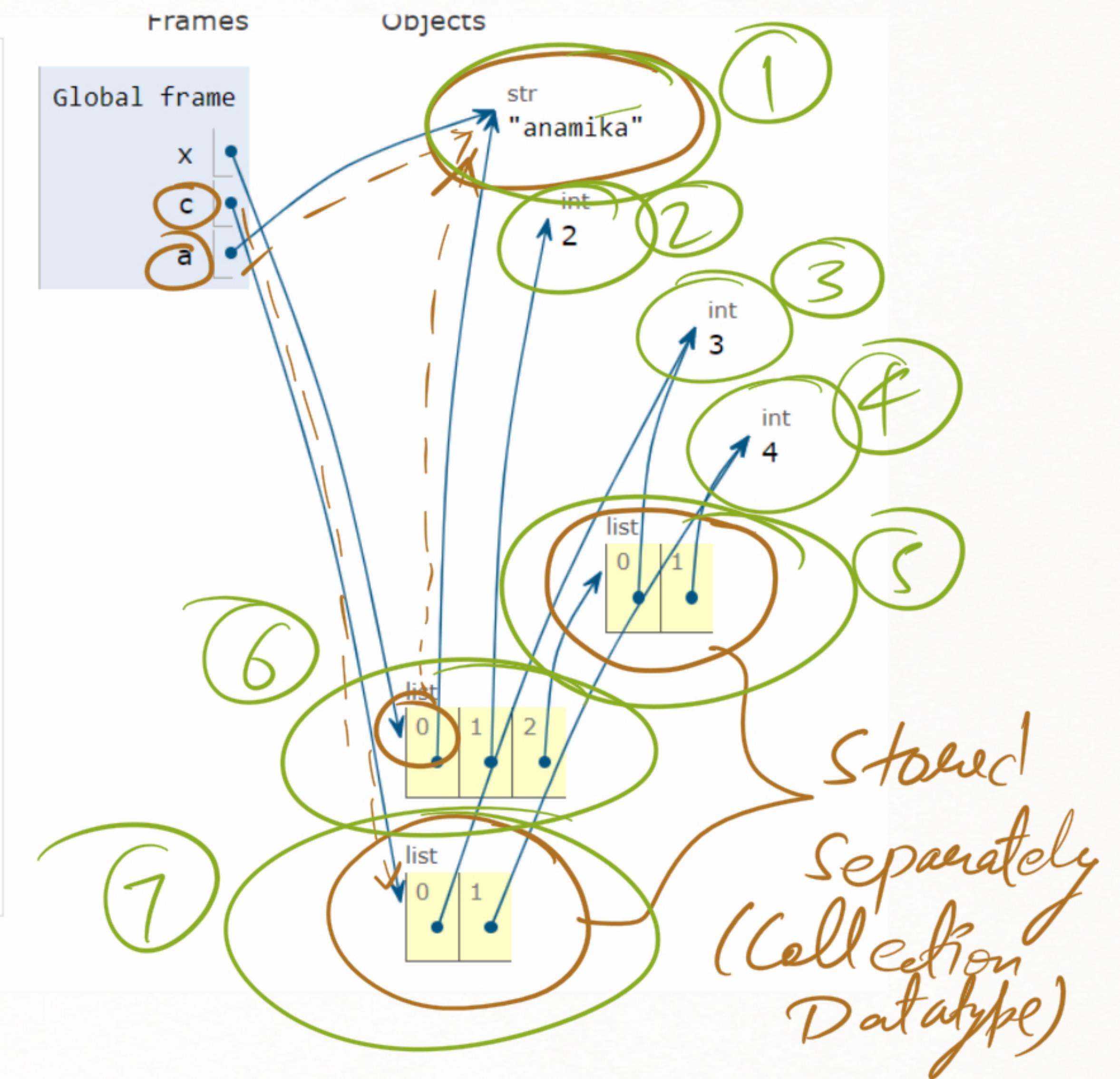
x = ['anamika', 2, [3, 4]]
c = [3, 4]
a = 'anamika'
    
```

← str

How many objects
are here?

⇒ 7.

that just executed
1 line to execute

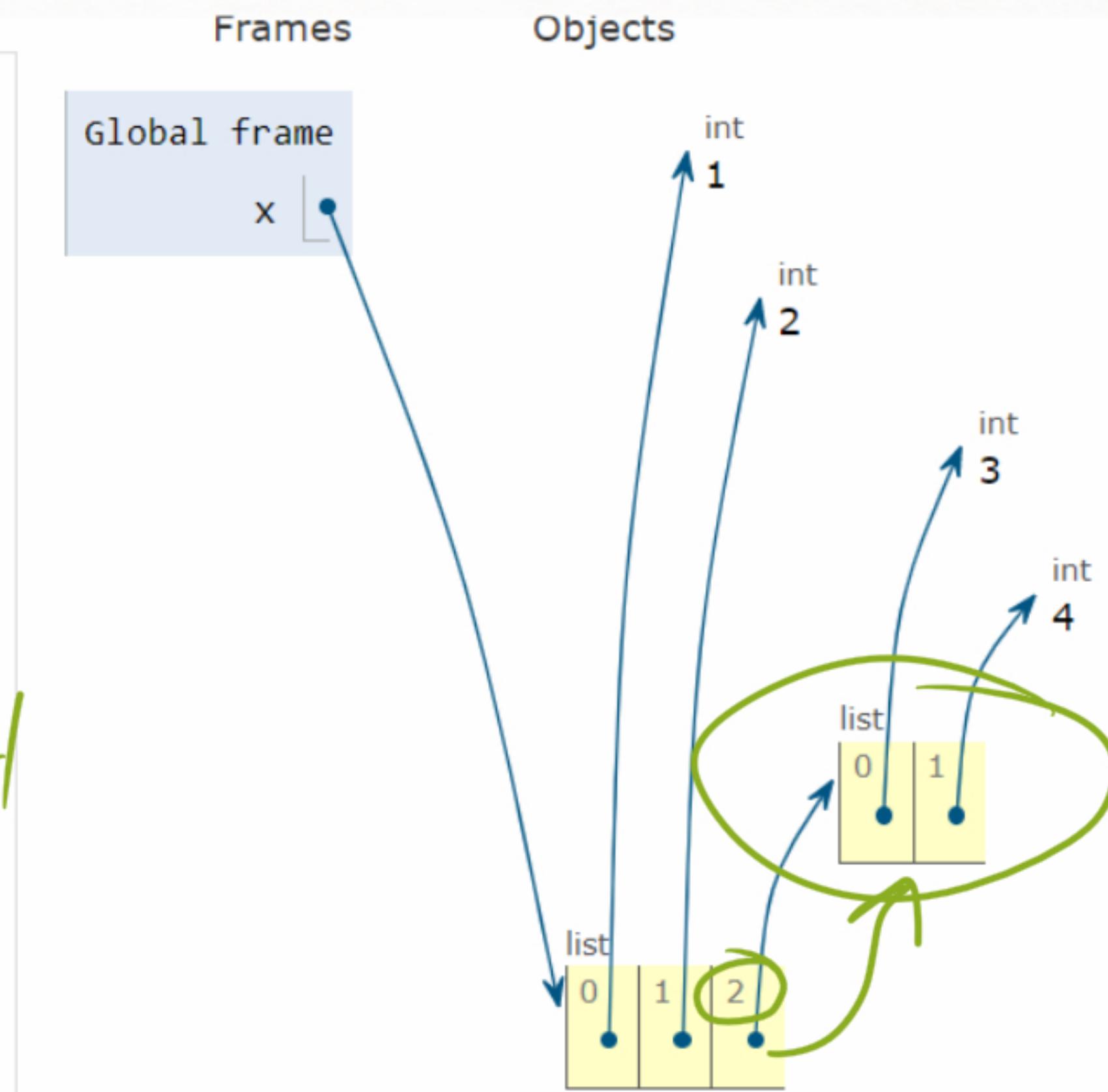


```
x = [1, 2, [3, 4]]
```

⇒ How many objects
are here?

$\Rightarrow 6$

Pistil is considered as a collection.



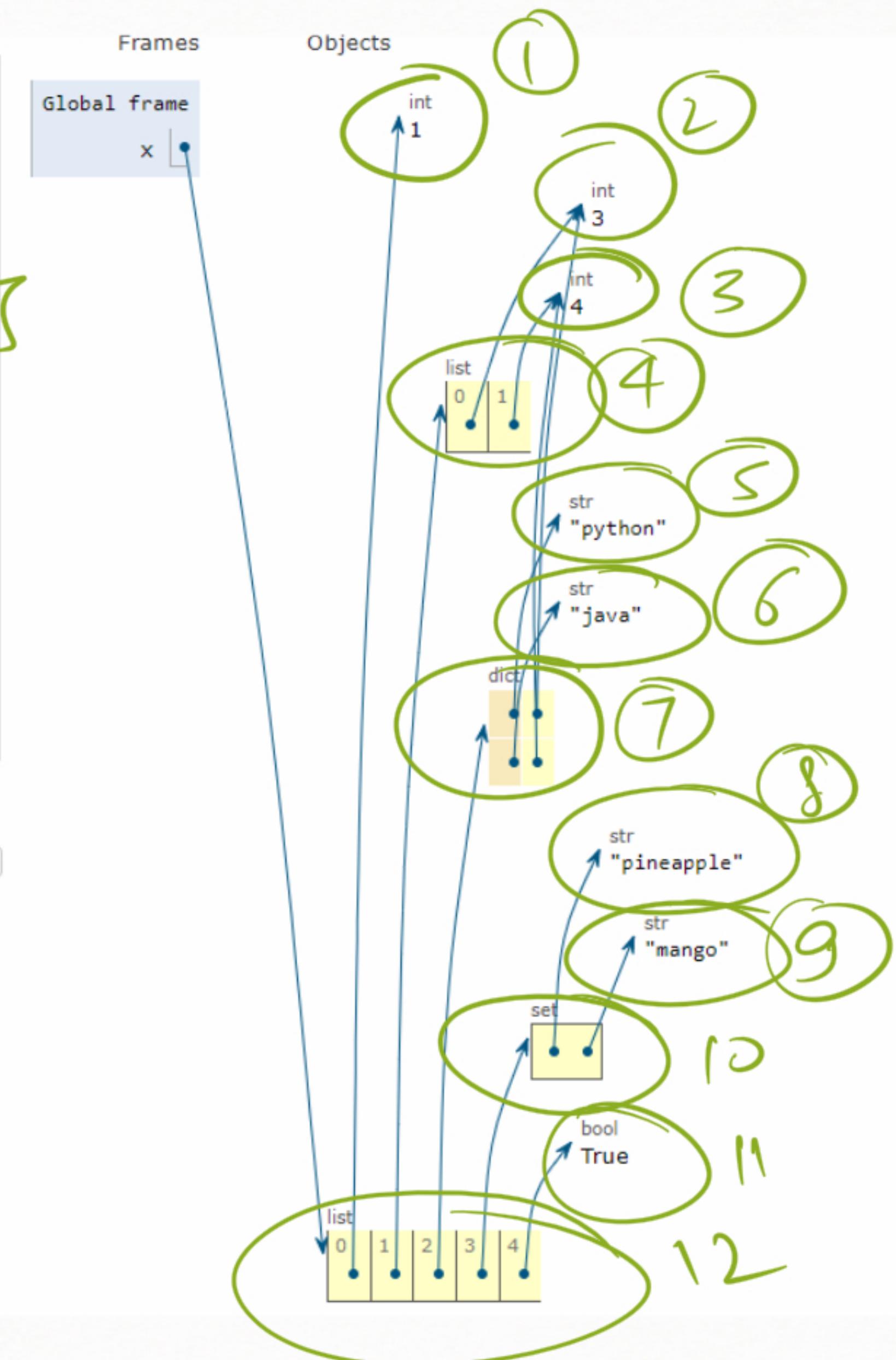
How many objects
=> 12

```
x = [1,[3,4],{'python':3,'java':4},  
['mango','pineapple'],True]
```

that just executed
line to execute

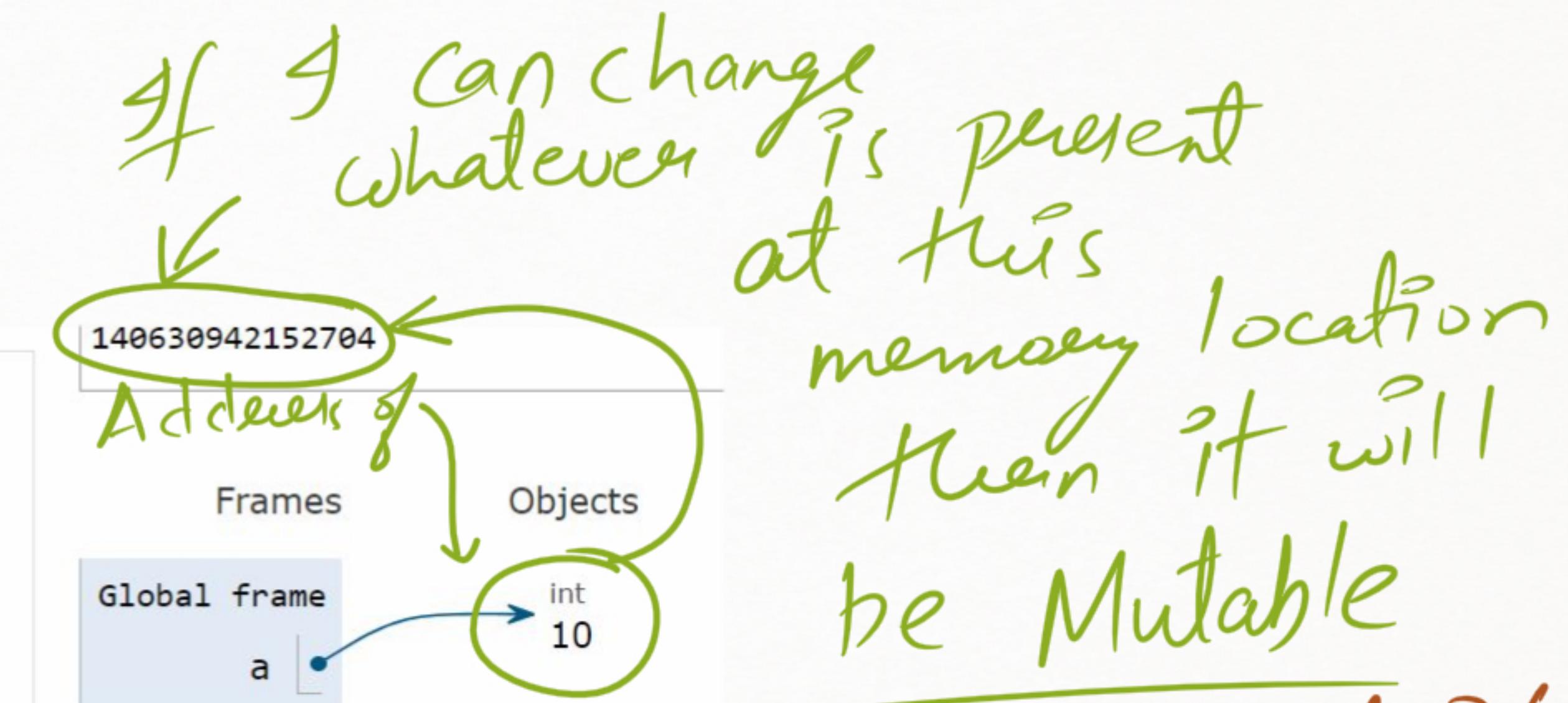
<< First < Prev Next > Last >>

Done running (2 steps)



Immutability

```
a = 10  
print(id(a))
```



code in Python 3.6

```
a = 10  
print(id(a))  
a = 20  
print(id(a))
```

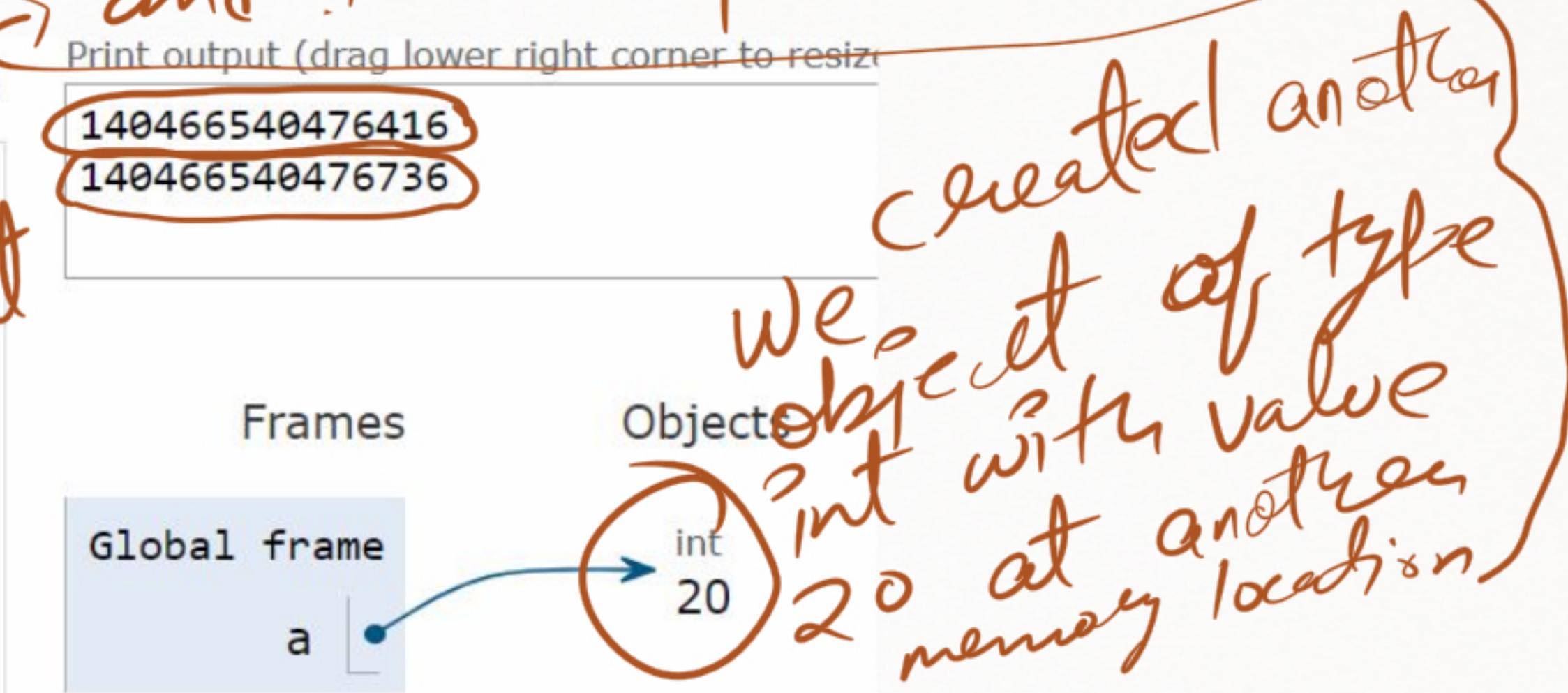
(drag lower right corner to resize code editor)

Print output (drag lower right corner to resize)

```
140466540476416  
140466540476736
```

You might think that I am able to change 10 to 20. But this is NOT what happens!

and then we pointed a to it.



In [11]:

```
a = 10  
b = a  
print(id(a))
```

1549006912

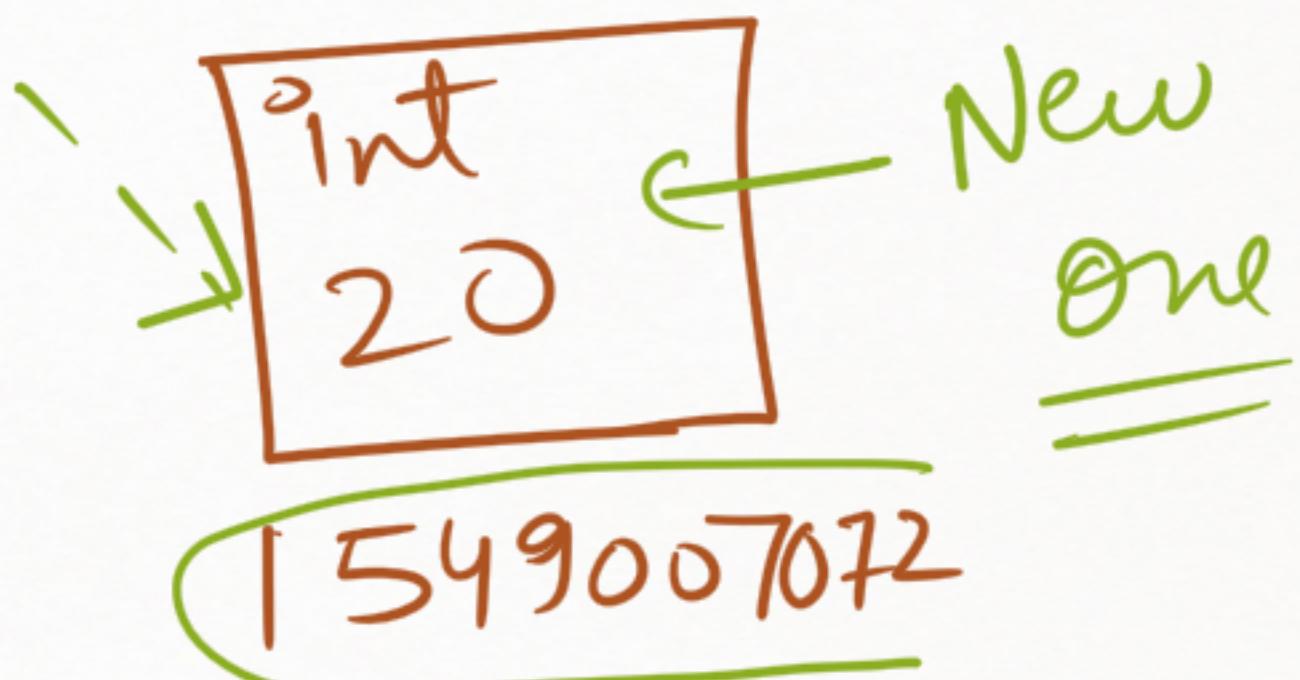
}



In [12]:

```
a = 20  
print(id(a))✓
```

1549007072



In [13]:

```
print(id(b))
```

1549006912

```
fruits = ['kiwi', 'apple', 'mango']
print(id(fruits))
```

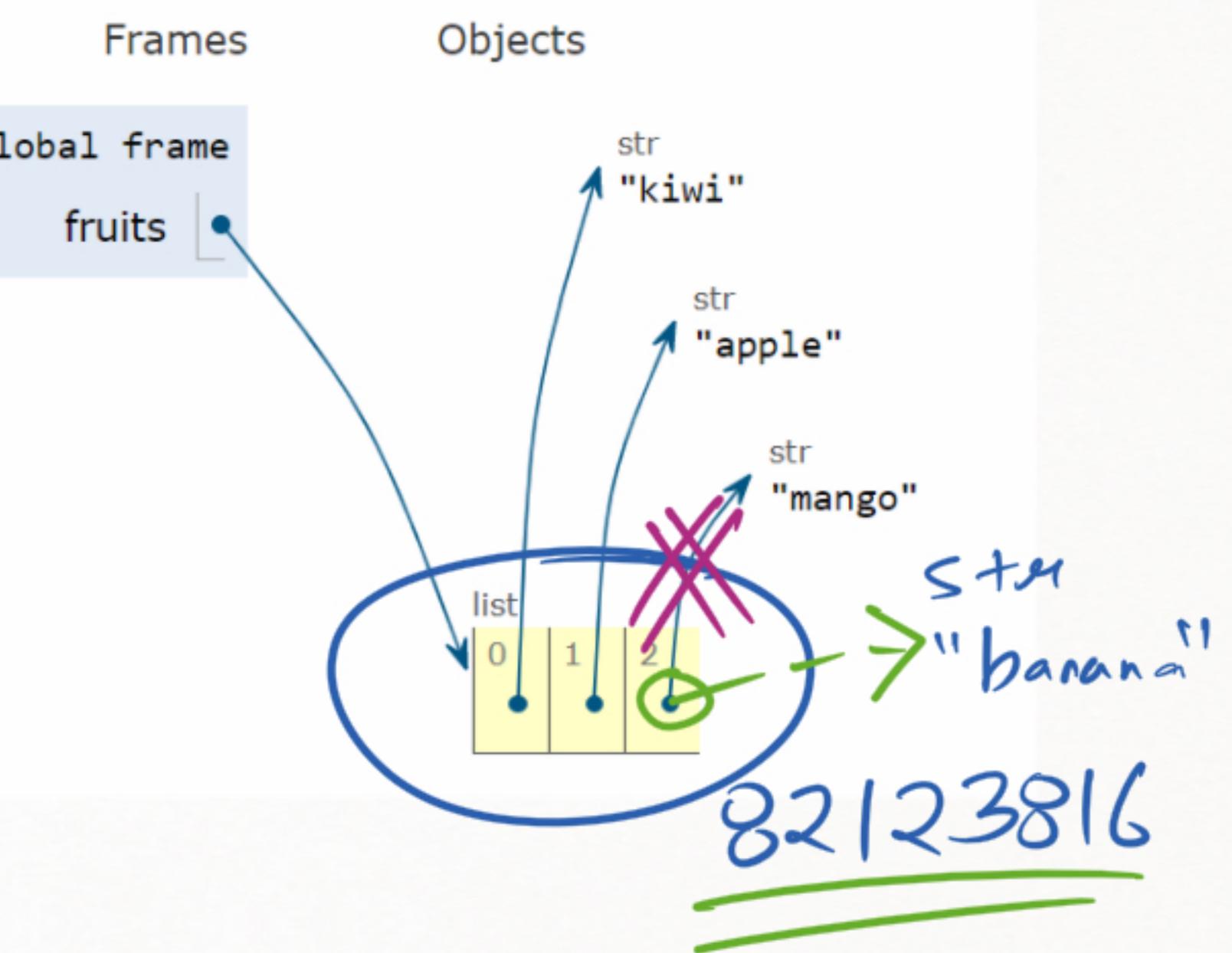
82123816

```
fruits[2] = 'banana'
print(fruits)
print(id(fruits))
```

['kiwi', 'apple', 'banana']

82123816

List are
Mutable



We changed the
part which is
present directly
in the list.