

Index → str, list, tuple

→ Starts with 0
↓
first item

$x = "Apple"$

list-of-fruits =

$x[4] \Rightarrow e$

$\begin{bmatrix} "Kiwi", "mango", "apple" \end{bmatrix}$

0 → 1 → 2
list-of-fruits[2] // apple.

"Python"
↑↑↑↑↑
0 1 2 3 4 5
→

```
fruits = ['kiwi', 'apple', 'mango', 'peach']
```

```
fruits[1]
```

```
'apple'
```

↑
0 1 2 3

```
fruits[5]
```

5th Index Position

IndexError

```
<ipython-input-5-1753250162d1> in <module>
```

```
----> 1 fruits[5]
```

IndexError: list index out of range

Traceback (most recent call last)

Negative Indexing

```
fruits = ['kiwi', 'apple', 'mango', 'peach']
```

```
fruits[1]
```

'apple'
list range
index out of range

fruits[0]

fruits[-4] → 'kiwi'

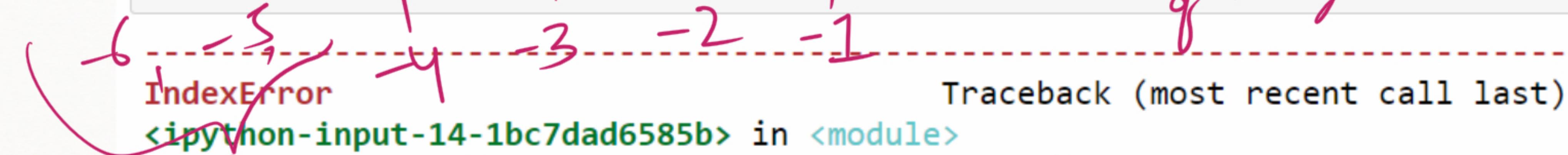
fruits[-1] ⇒ 'peach'

4, 5, ...

list index out of range

```
fruits = ['kiwi', 'apple', 'mango', 'peach']
```

```
print(fruits[-5])
```



IndexError

Traceback (most recent call last)

```
<ipython-input-14-1bc7dad6585b> in <module>
```

```
1 fruits = ['kiwi', 'apple', 'mango', 'peach']
```

```
2
```

```
3 print(fruits[-5])
```

IndexError: list index out of range

$$\text{len}(fruits) = 4$$

```
fruits = ['kiwi', 'apple', 'mango', 'peach']  
fruits[1]  
'apple'  
0 1 2 3  
4 items
```

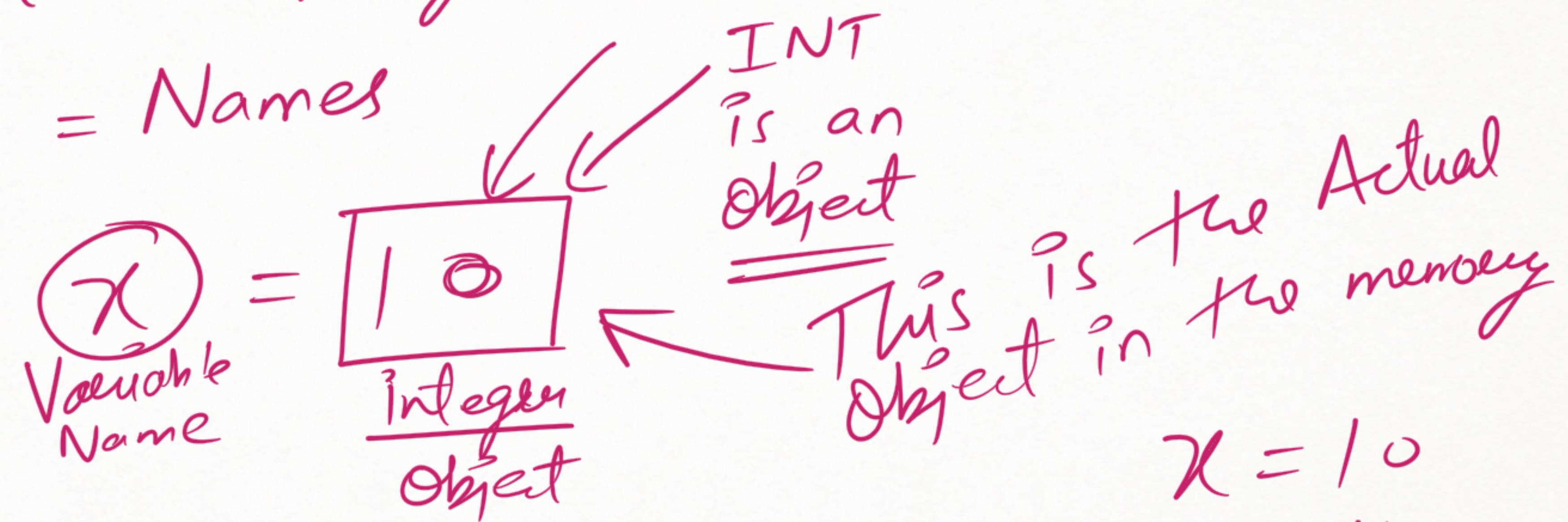
$$\text{len}(\text{fruits}) = 4$$

$$4 - 2 = 2$$

$$4 - 1 = 3$$

Assignment: Assigning

Variables = Names



$$x = 10$$

$$y = 4$$

→ Everything in Python is Object
→ All the data types are Objects

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

How are Objects Stored in Memory

Code in Python 3.6

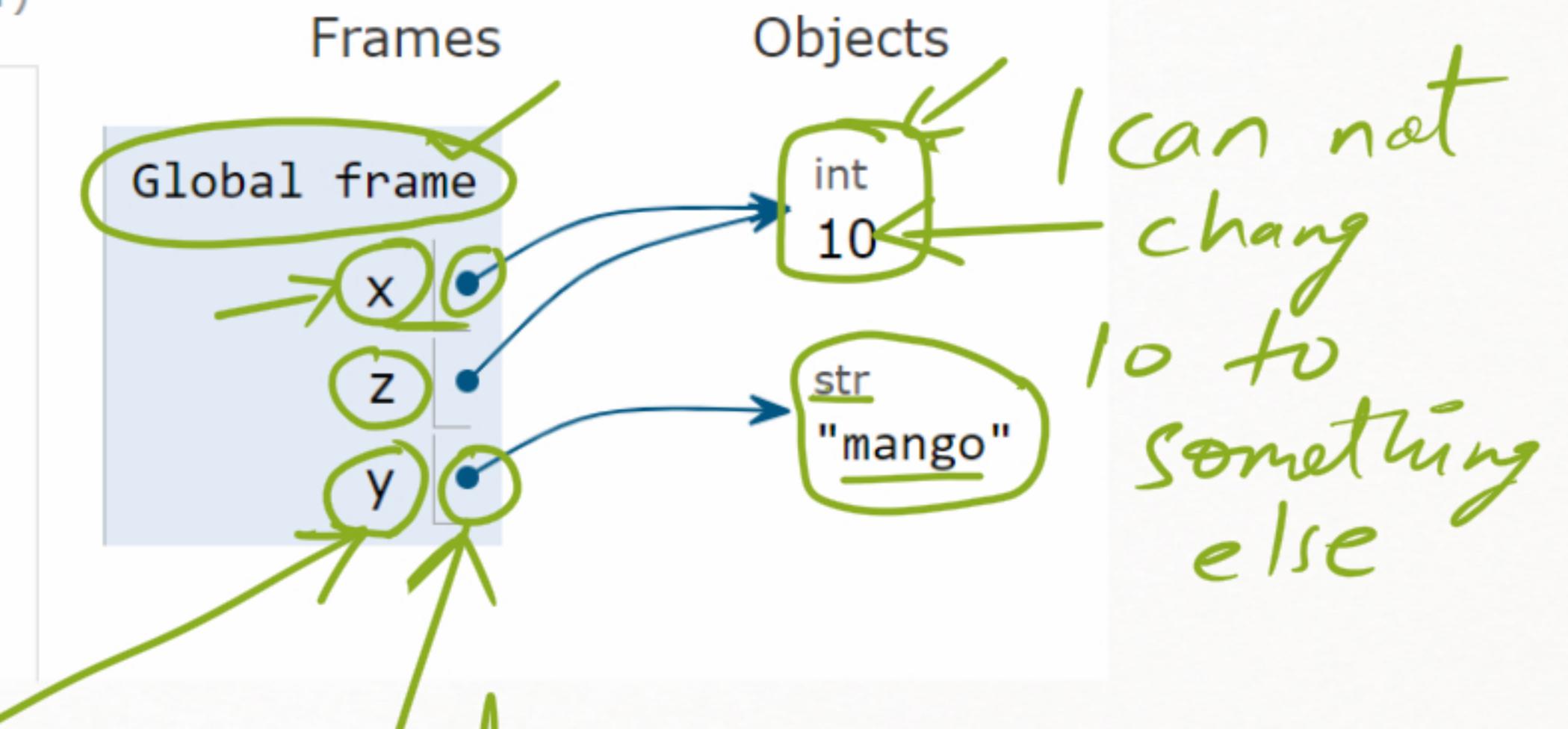
(drag lower right corner to resize code editor)

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

Actual objects

Name.

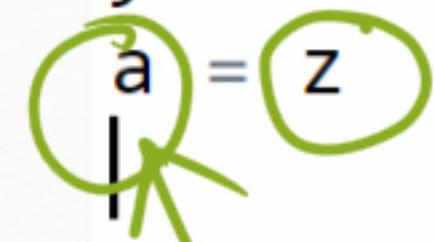
Pointer
Pointing to the
actual object.



Code in Python 3.6

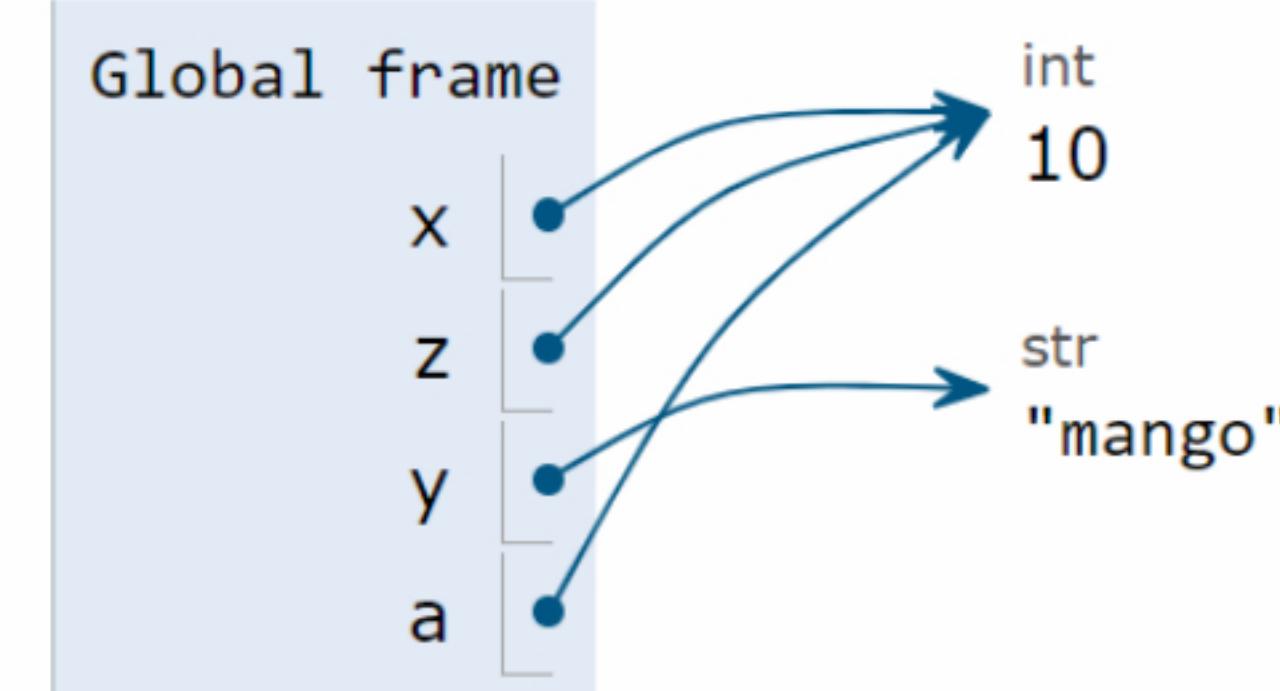
(drag lower right corner to resize code editor)

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



a will start pointing to
where z is pointing already.

Frames Objects



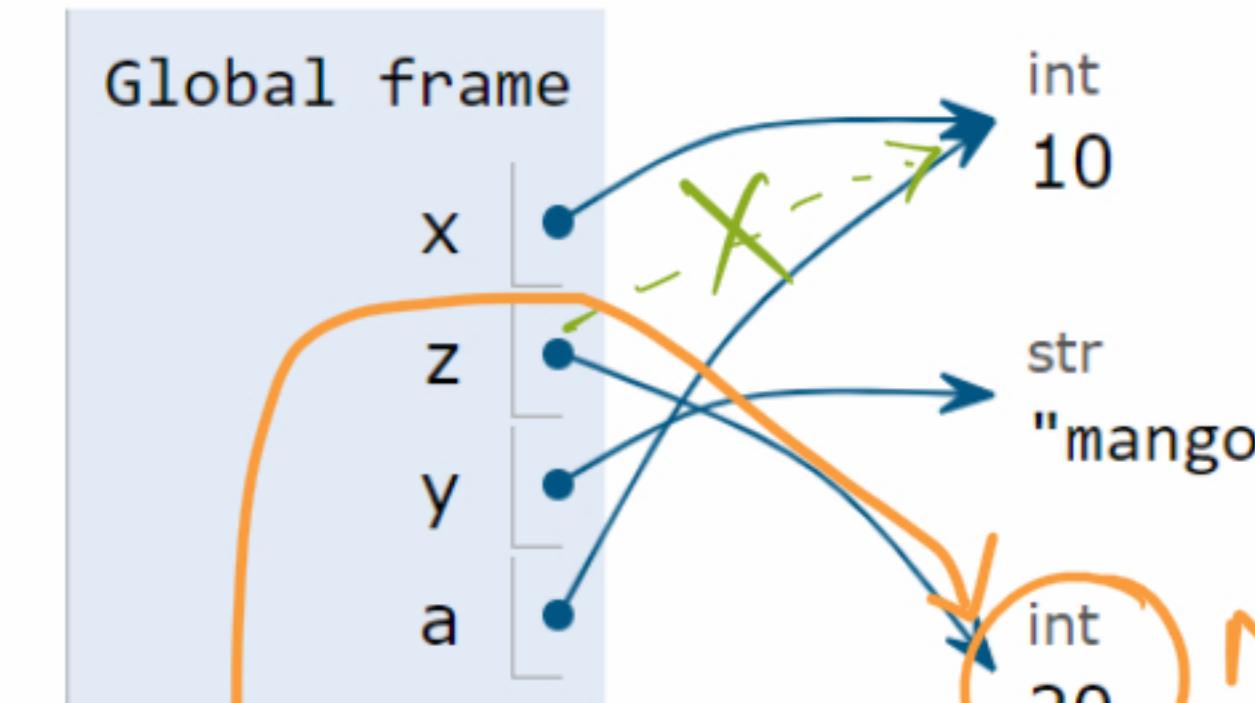
ode in Python 3.6

(drag lower right corner to resize code editor)

```
x = 10  
z = 10  
y = 'mango'  
a = z  
z = 20
```

You have created
another object of type
int with value
20 in the memory

Frames Objects



New object

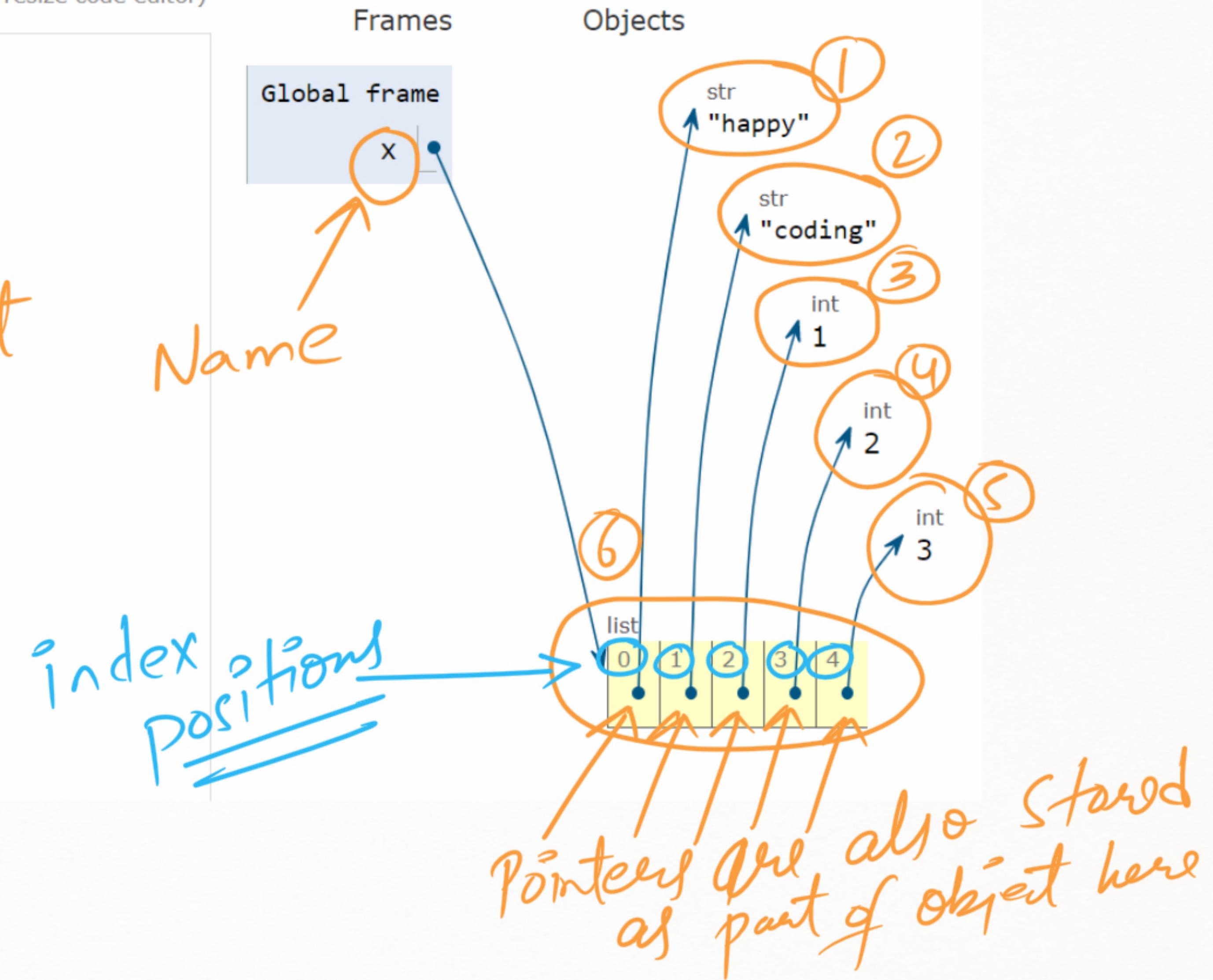
ode in Python 3.6

(drag lower right corner to resize code editor)

```
x = ['happy', 'coding', 1, 2, 3]
```

How many objects?
=> 6

How many Assignment
=> 1



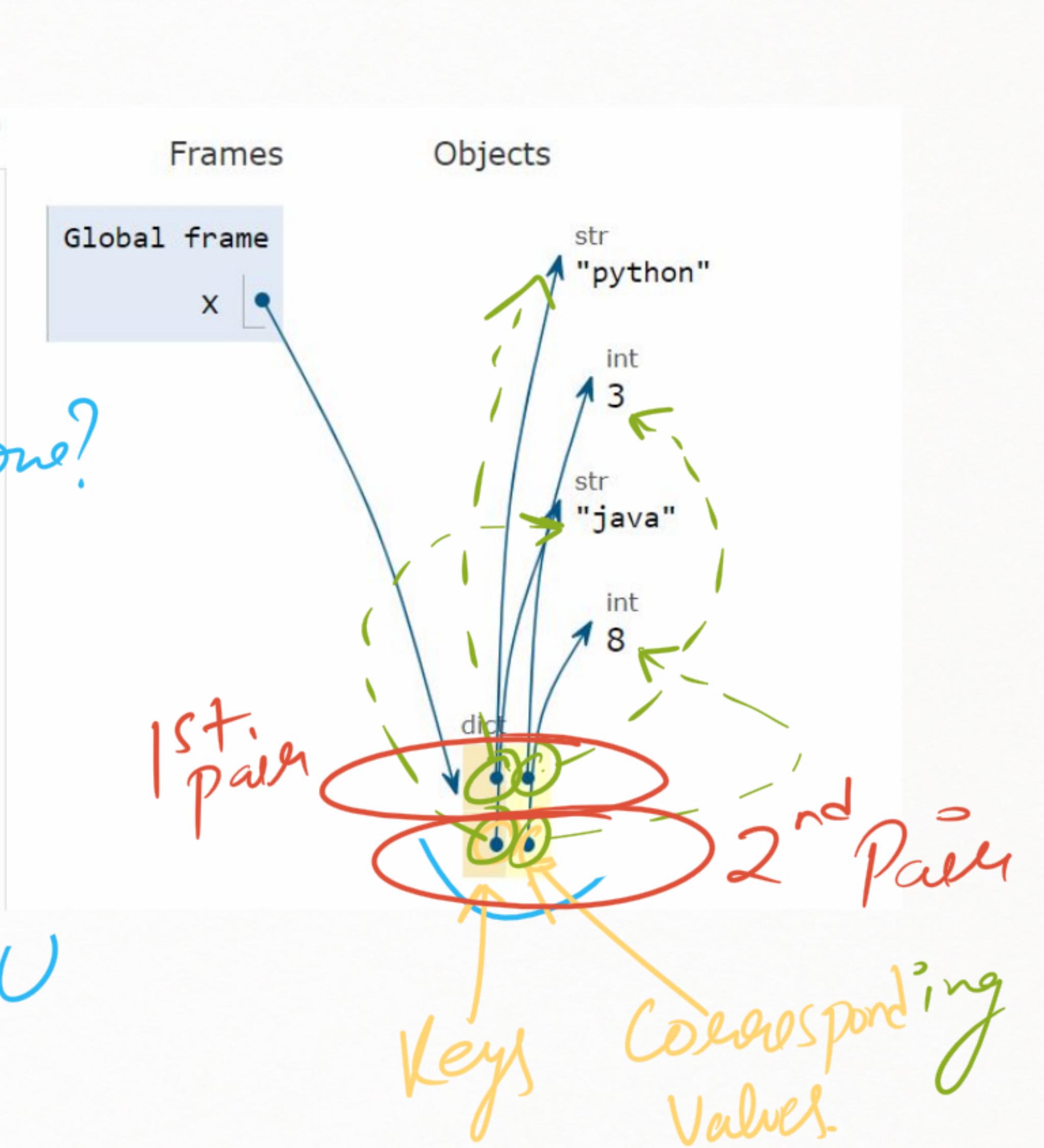
Code in Python 3.6

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

How many Objects?
=> 5

How many Names, which one?
=> 1, x

Dict => "Key": "Value"
=> Indexing Possible => N/A



Code in Python 3.6

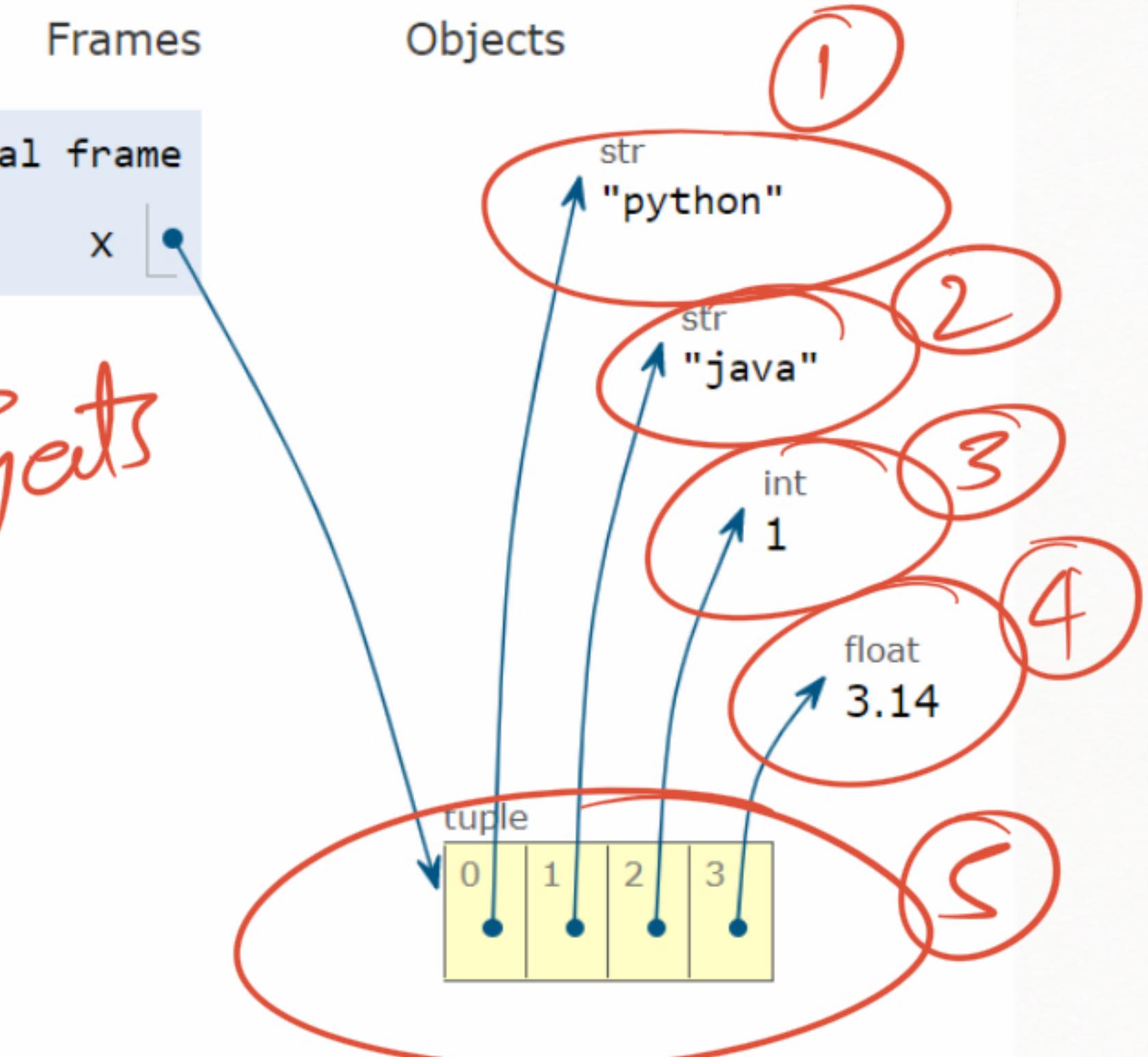
(drag lower right corner to resize code editor)

```
x = ('python', 'java', 1, 3.14)
```

How many objects
=> 5

How many immutable
are here?
= 1 + 1 + 1 + 1 + 1

= 5 ✓



ode in Python 3.6

(drag lower right corner to resize code editor)

```
x = {'python', 'java', 1, 3.14, 'python', 1, 1, 1, 1, 1, 1, 1, 'java'}
```

How many objects?

⇒ 5

How many Name

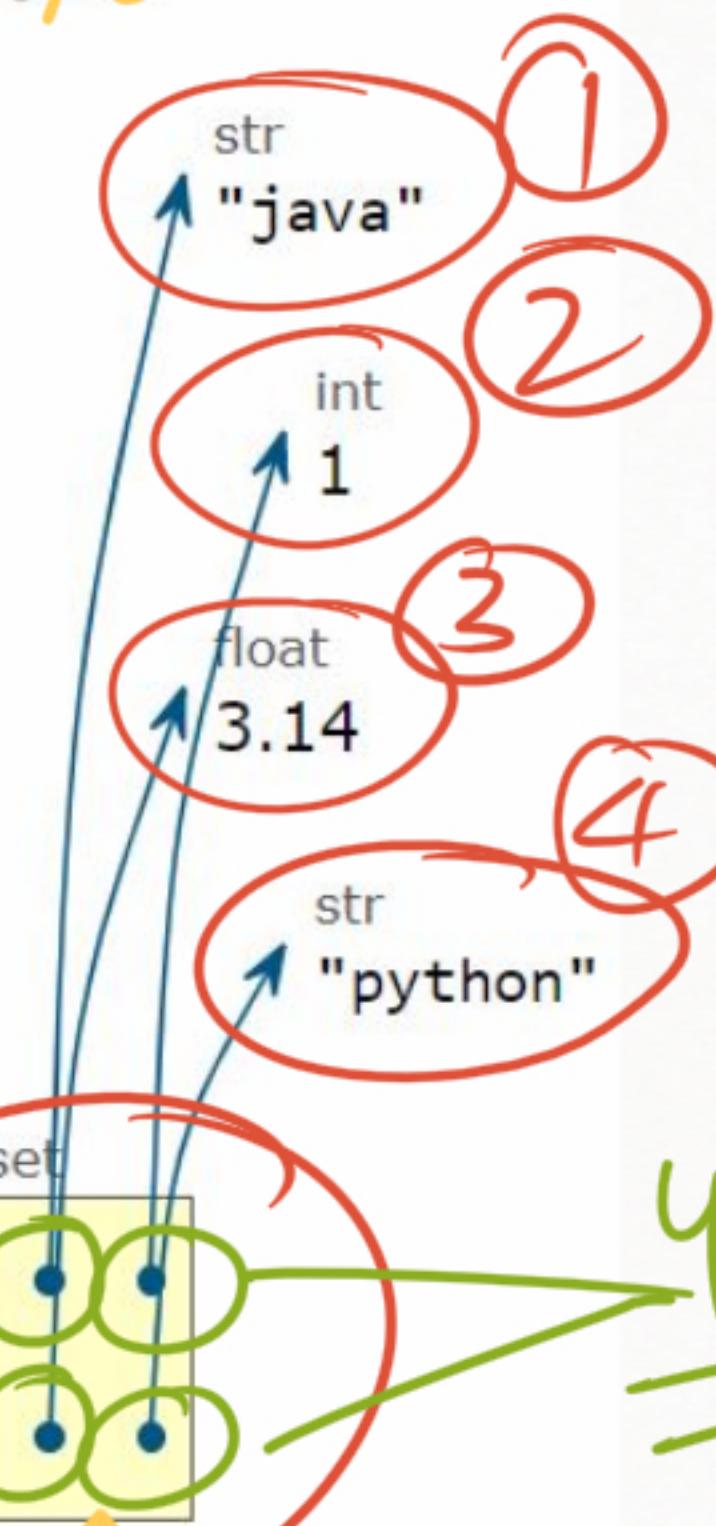
⇒ 1, (2)

Set x = {key, key², ...}

len(x) || ④

Duplicates
Frames
and Objects
are not stored

Global frame
x



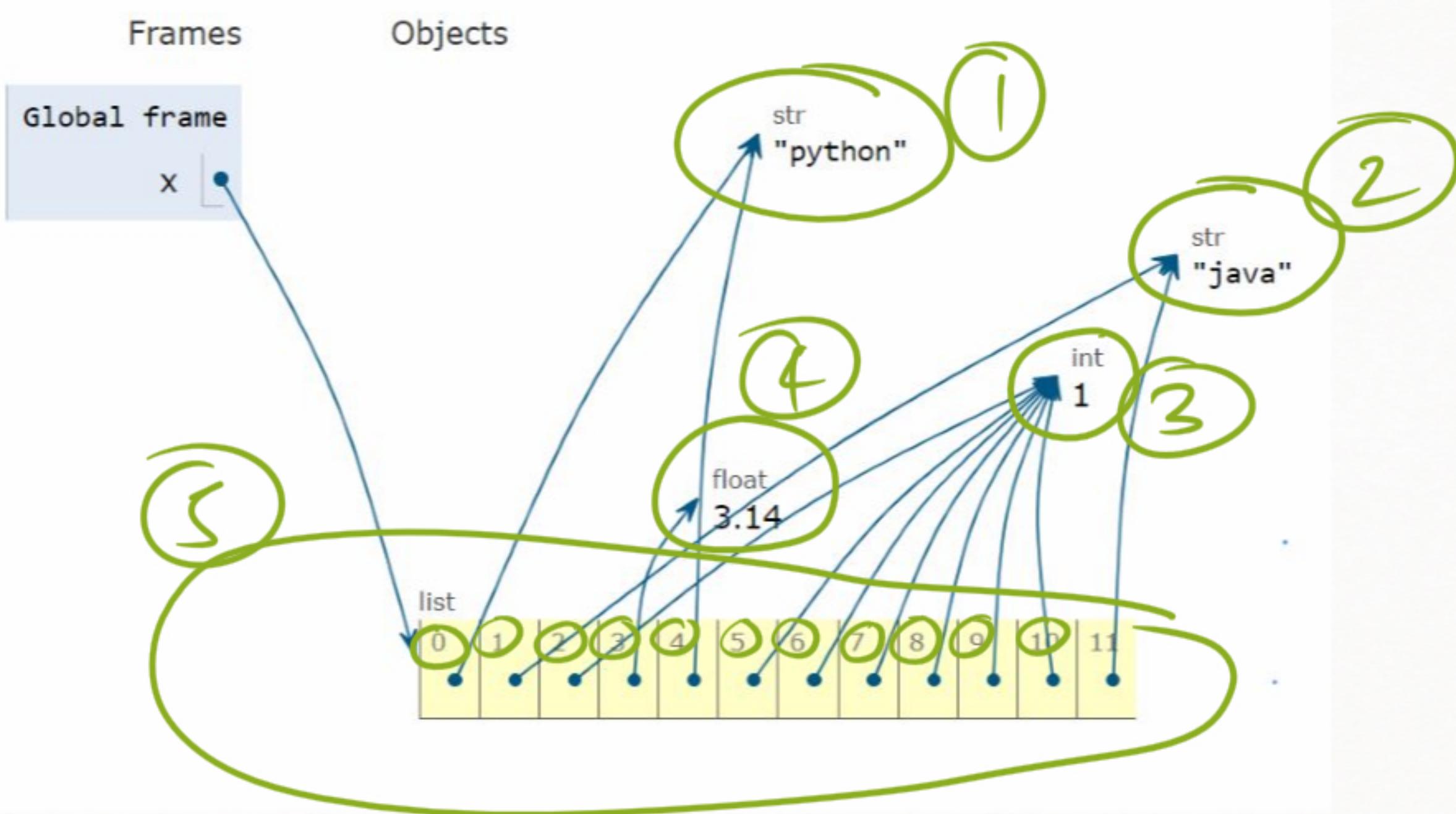
?index positions
are not visible here
No etc points
for duplcat

Code in Python 3.6

(drag lower right corner to resize code editor)

```
x = ['python', 'java', 1, 3.14, 'python', 1, 1, 1, 1, 1, 1, 'java']
```

How many objects
= 5
len(x) // 12



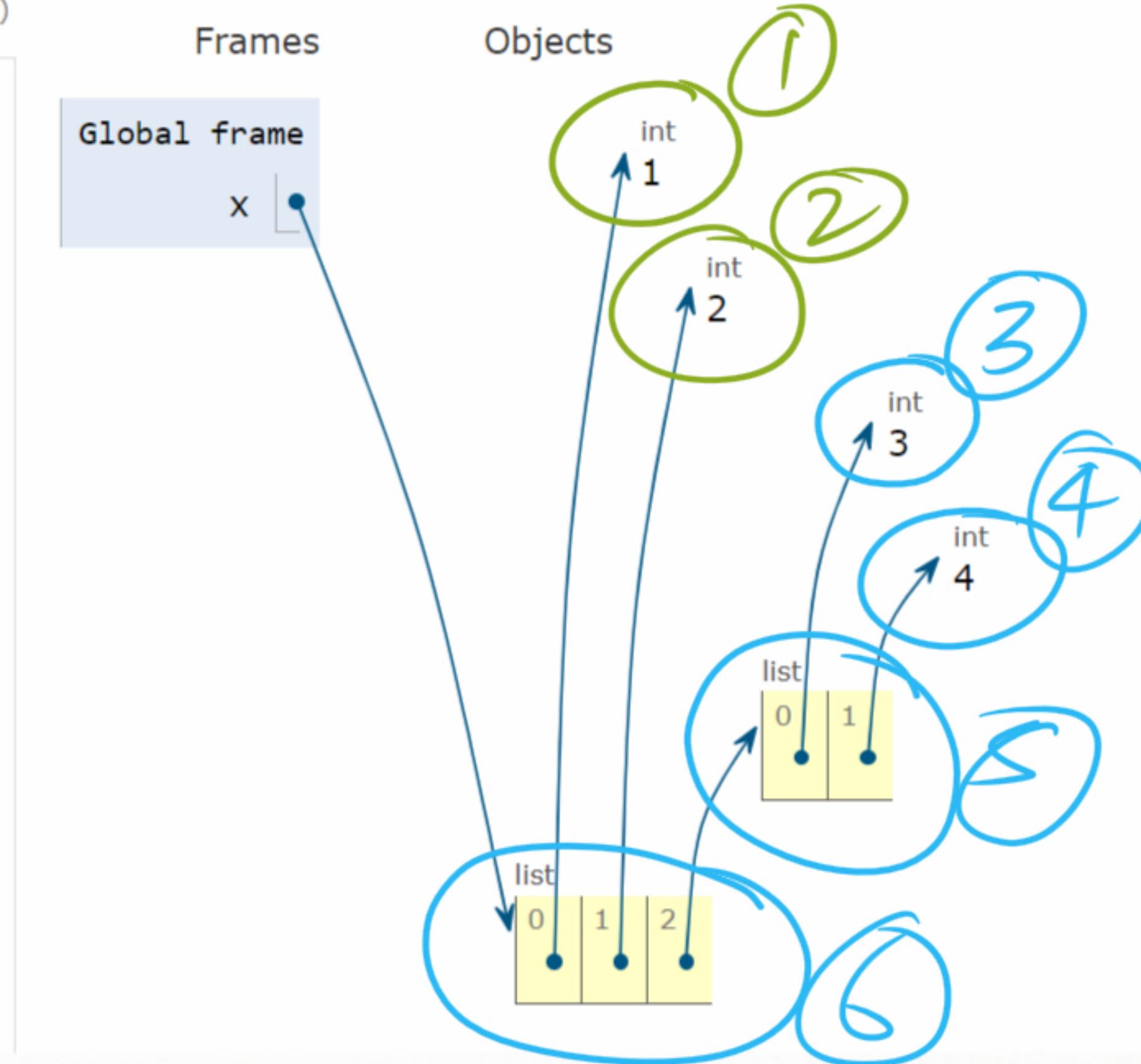
ode in Python 3.6

(drag lower right corner to resize code editor)

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

How many objects
⇒ 6

How many Names
⇒ 1 (n)



code in | Python 3.6

(drag lower right corner to resize code editor)

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

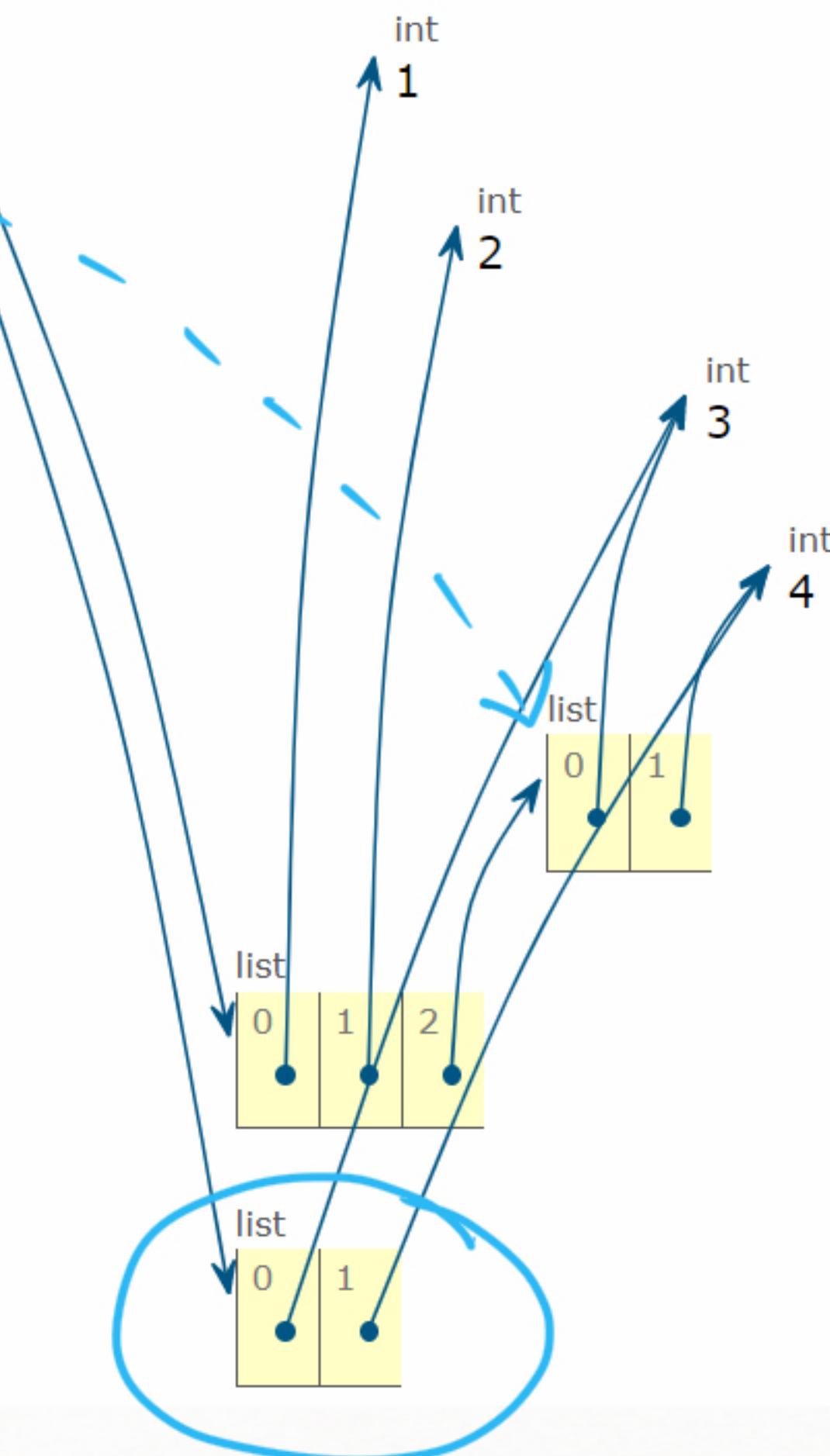
New list will be created in memory but Non Collection Data types 3, 4 will not be all created.

Frames

Global frame

x
y

Objects



```
x = [{'python':3,'java':8},{3,4,5},True,False]
```

How many objects?

=> 11

that just executed
line to execute

<< First < Prev Next > Last >>

Done running (1 steps)

