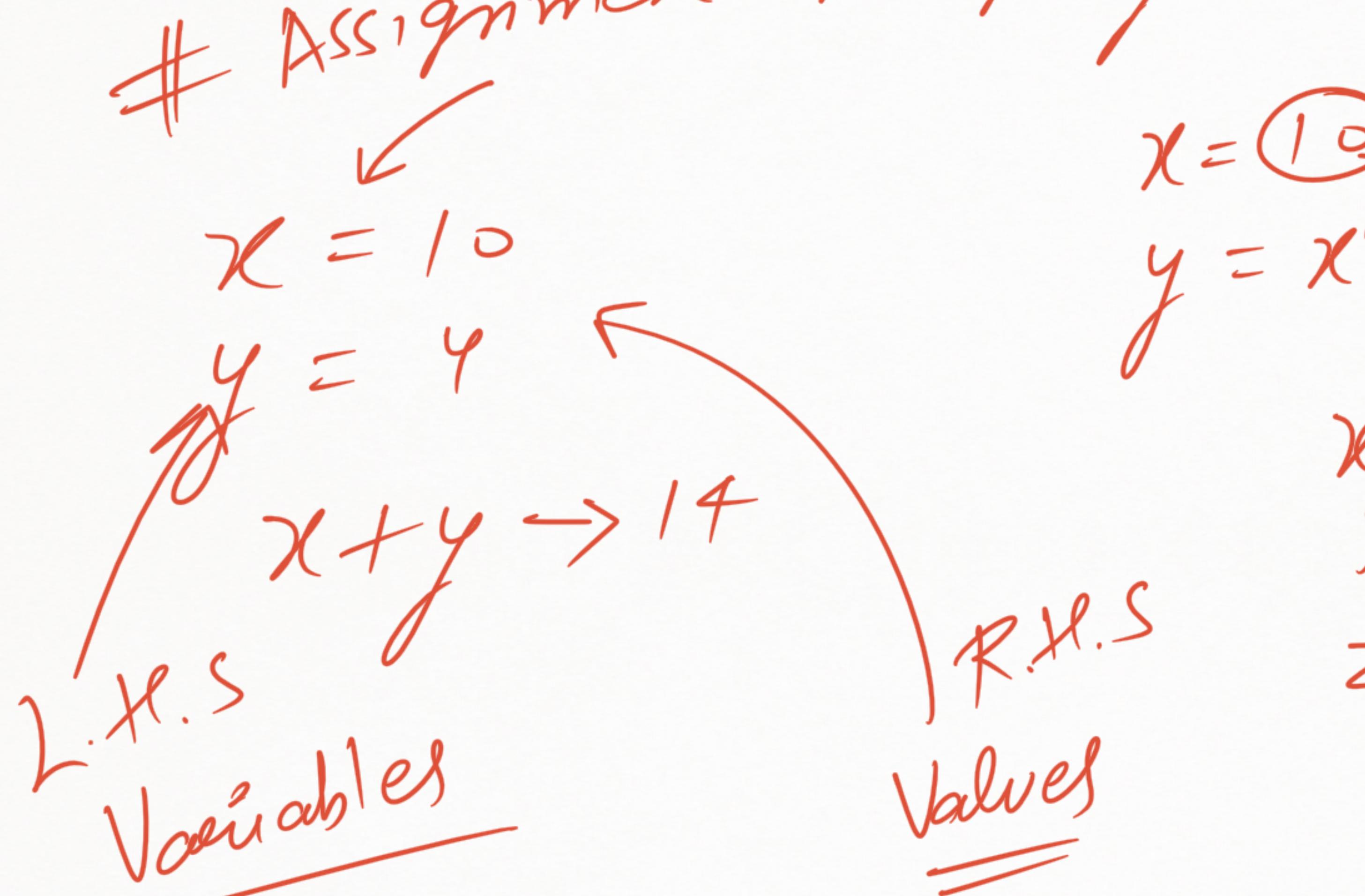


Assignment: Assigning a Value



Assignment operator " $=$ "

$$\begin{aligned}x &= 10 \\y &= 4 \\z &= \underline{x+y} \\&\quad \uparrow \\&\quad \text{Value}\end{aligned}$$

Write code in Python 3.6

```
1 x = 10  
2 y = 10  
3 print(x)  
4
```

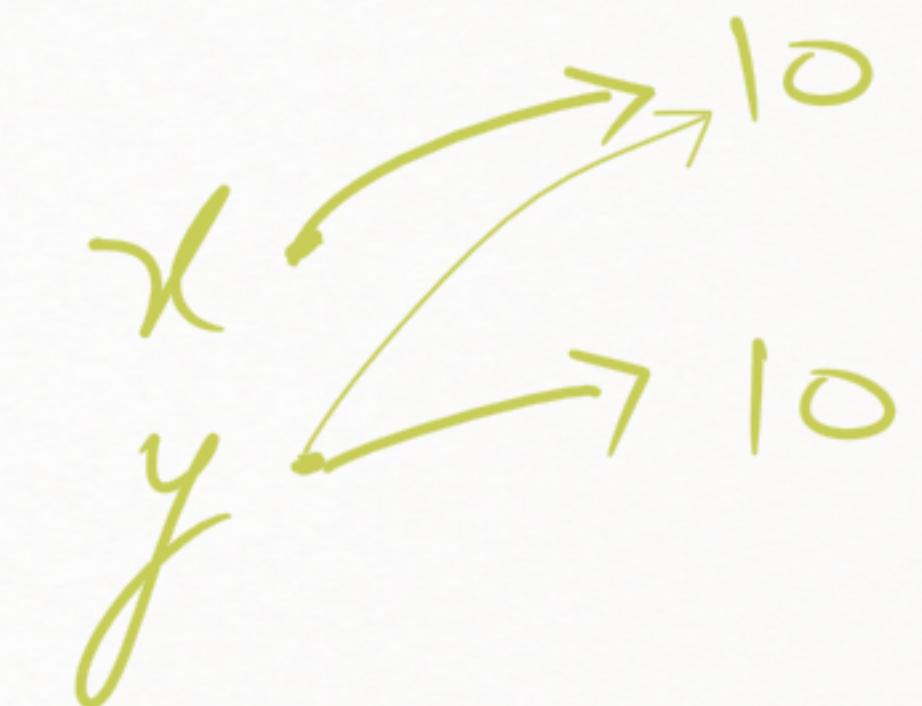
(drag lower right corner to resize code editor)

Print output (drag lower right corner to resize)

10

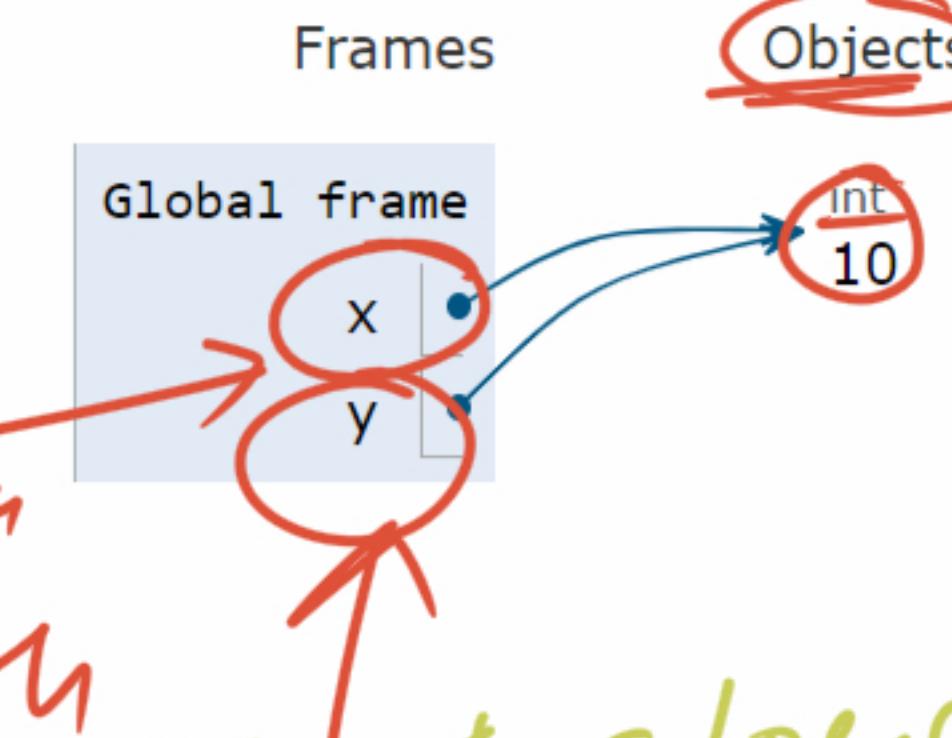
R.H.S

VS



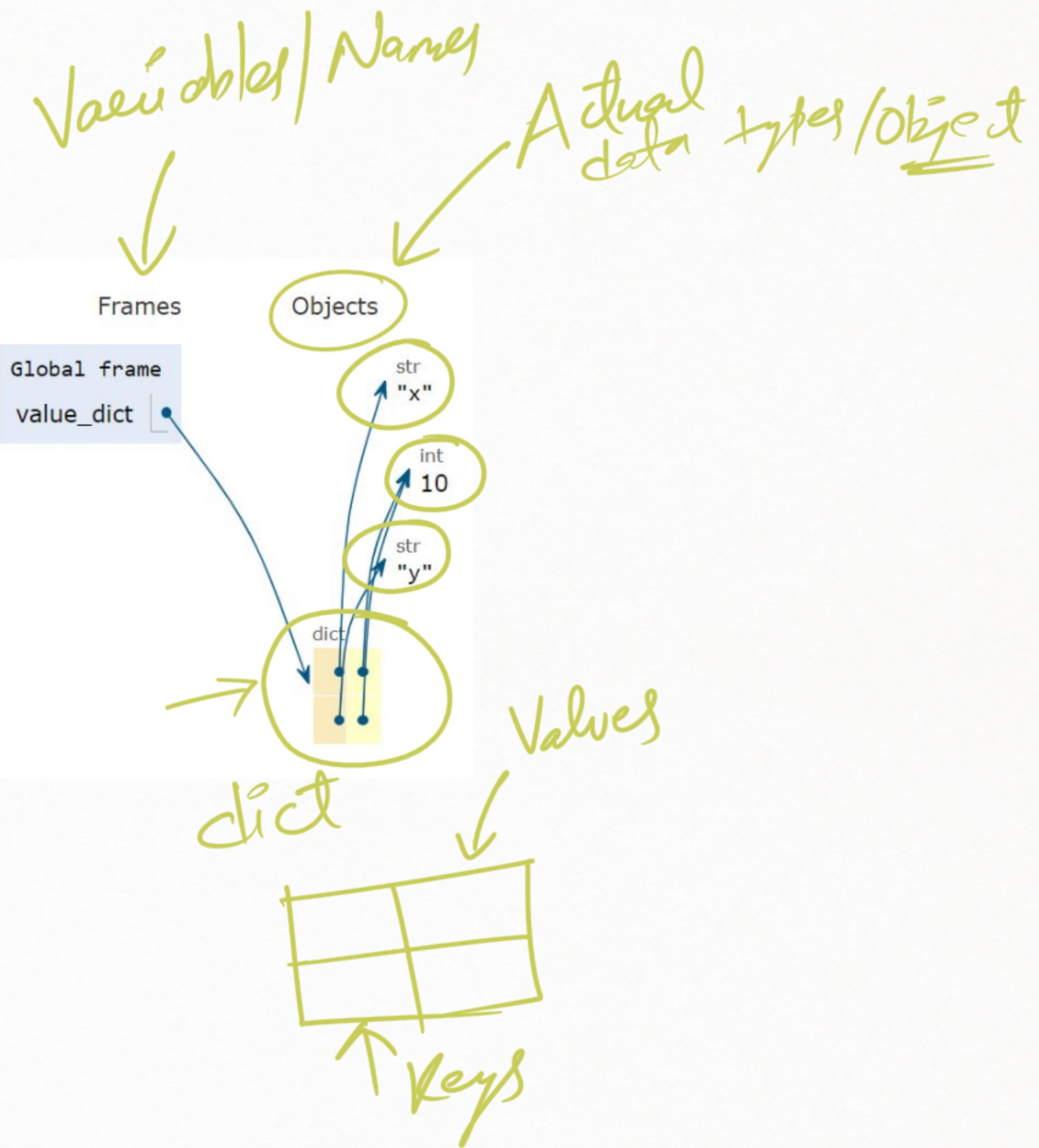
Pointing
in memory
value 10
a name
to integer
with
`x` is just

If `g` state `y = 10`
then `y` will also point
to the same `10` in the
memory.



Write code in Python 3.6 (drag lower right corner to resize code editor)

```
1 value_dict = {'x':10,'y':10}
2
```



$\text{id}()$ → Memory location of the stored object.

Print output (drag lower right corner to resize)

10

Frames Objects

Global frame



$x \rightarrow 10$
 $y \rightarrow 10$

Memory location

In [44]: # we are studying id

```
x = 10  
y = 10
```

```
print(id(x))  
print(id(y))|
```

```
1481963584  
1481963584
```

type gives the datatype

In [50]:

```
x = 'abhinandan'  
y = 45.21  
z = 786  
a = None  
b = [1,2,3]  
c = (2,4,6)  
d = {'python':3,'numpy':1,'pandas':1}
```

} literal/Naive/Basic

Collections

In [52]:

```
print(type(x))  
print(type(y))  
print(type(z))  
print(type(a))  
print(type(b))  
print(type(c))  
print(type(d))
```

```
<class 'str'>  
<class 'float'>  
<class 'int'>  
<class 'NoneType'>  
<class 'list'>  
<class 'tuple'>  
<class 'dict'>
```