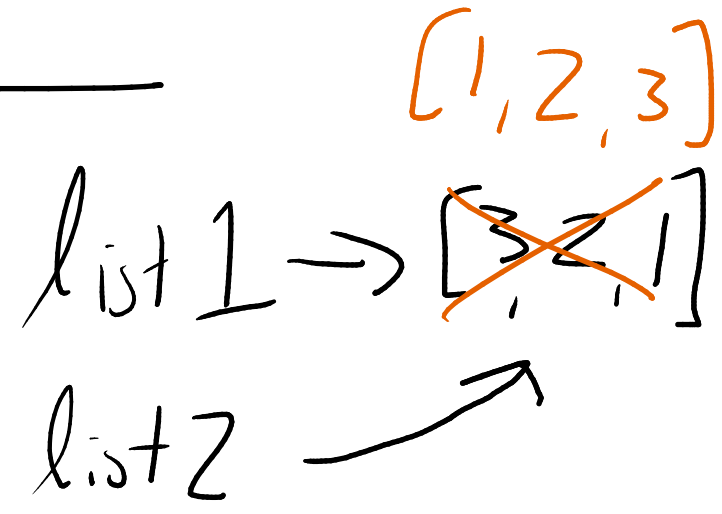
# Why are side effects potentially problematic?

---

Python

```
list1 = [3, 2, 1]
list2 = list1
```




```
# millions of lines of code.....
list2.sort()
print(list1)
```

→ [1, 2, 3]

Confusion, pain.

```
x = 3
def doit():
    x = 8

doit()
print(x)
```

ugly

---

```
x = 3
def doit1():
    modify x to 8
def doit2():
    modify x to 10
do_in_parallel(doit1, doit2)
print(x)
```

(pseudocode)

side effects are the problems

unpredictable output!

C!

Lower level than Scheme

Imperative (mostly), mostly not functional

Not object-oriented

---

What is an assignment statement?

$x = 7$

Two different models that describe what this means.

Different languages use different models.

Reference model (Python, Scheme, Java objects)

$x = 7$

"$x$ is a reference, or a label, or an association with the 7"


$y = x$

"y is associated w/ the same data that $x$ is"

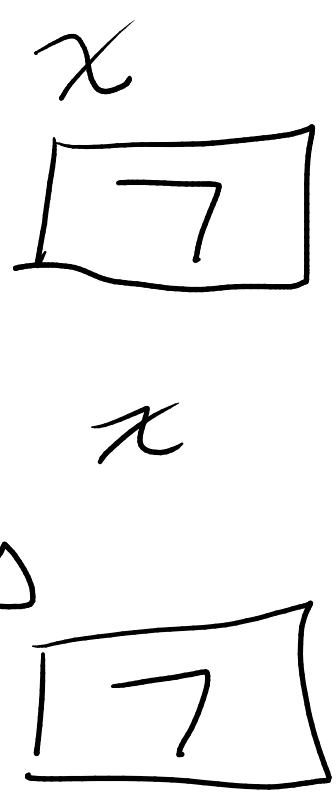
Value model (C, Java primitives)

$x = 7$

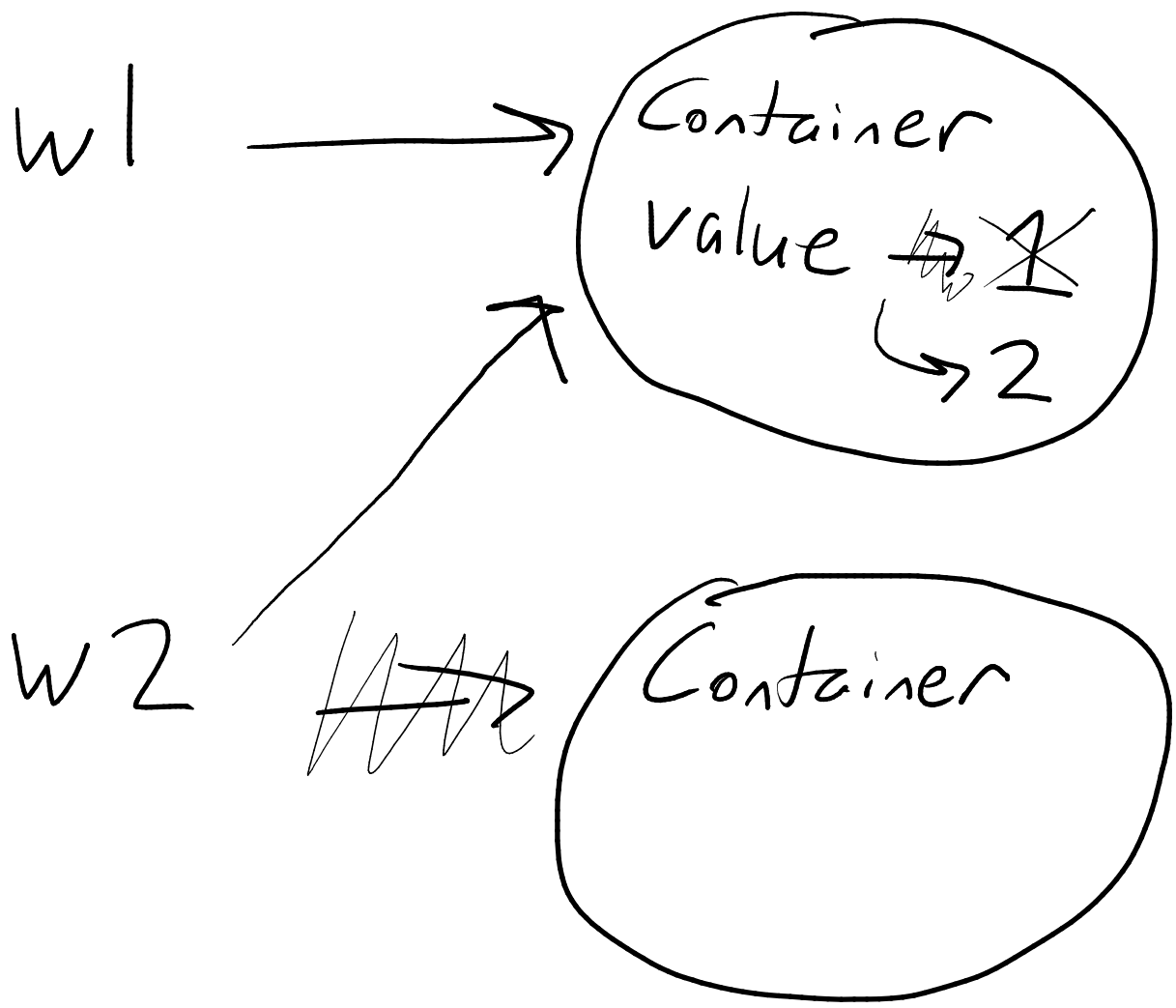
"$x$ is the name of a container that holds a 7"

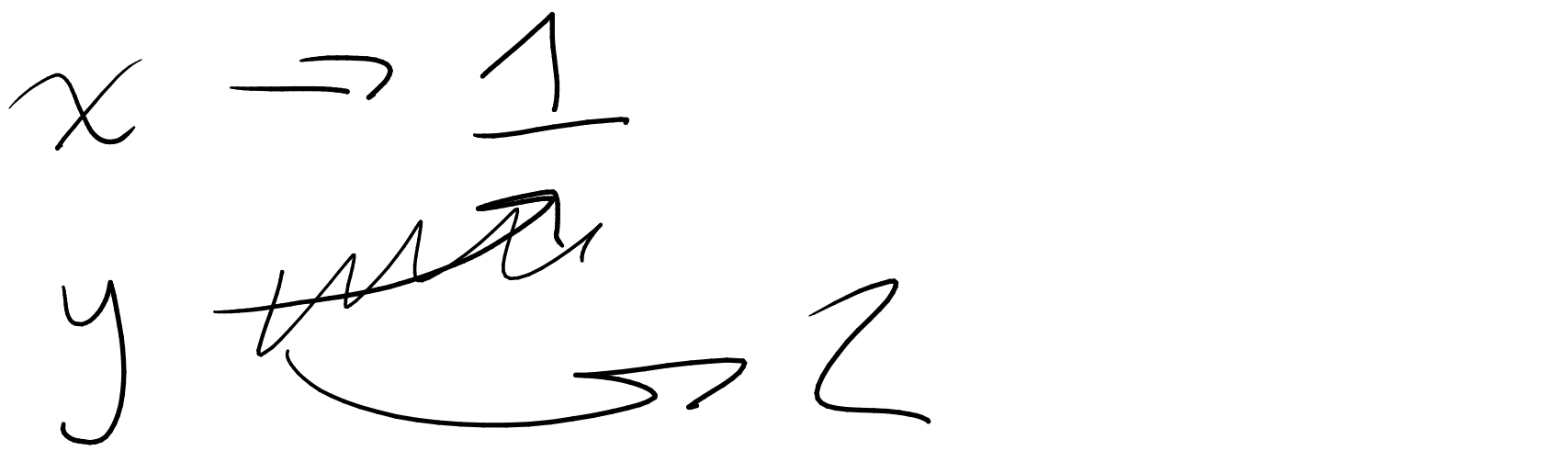
$y = x$

"copies the data in container $x$ to container $y$"



# Python container example




(after class)

```
x = 1
y = x
x = 2
print(y)
```