CSE 20 4-30-20

Extend P4 by 1 day

Exercise

figure out how to swap int's x & y without a temp.

Ex. BetterSort.py

Instance

x	y	z	
3	1	2	
1	3	2	last comp. of x & y was unnecessary
1	2	3	

Instance

x	y	z	
3	2	1	
2	3	1	
2	1	3	last comp. of x & y was necessary
1	2	3	

Ex. Sort4.py

Instance	a	b	c	d
	4	3	2	1
	3	4	2	1
	3	2	4	1
	3	2	1	4
	2	3	1	4
	2	1	3	4
	1	2	3	4

chap 6: more on functions

Ex write a function that returns the absolute value of a number

$$|x| = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{if } x < 0 \end{cases}$$

Possible solution: (bad solution)

```
def f(x):
    if x < 0
        return -x
    elif x > 0:
        return x
# end f()
```

if $x = 0$, return None

Exercise: fix this!!

Ex. Words.py

Exercise

Write a function ~~called~~ with heading

```
def count_2_letter_words(L):
    :
    :
#end ....
```

that counts and returns the number of 2-letter words in a list of strings L.

chap 7: more on loops

Recall Python's List object

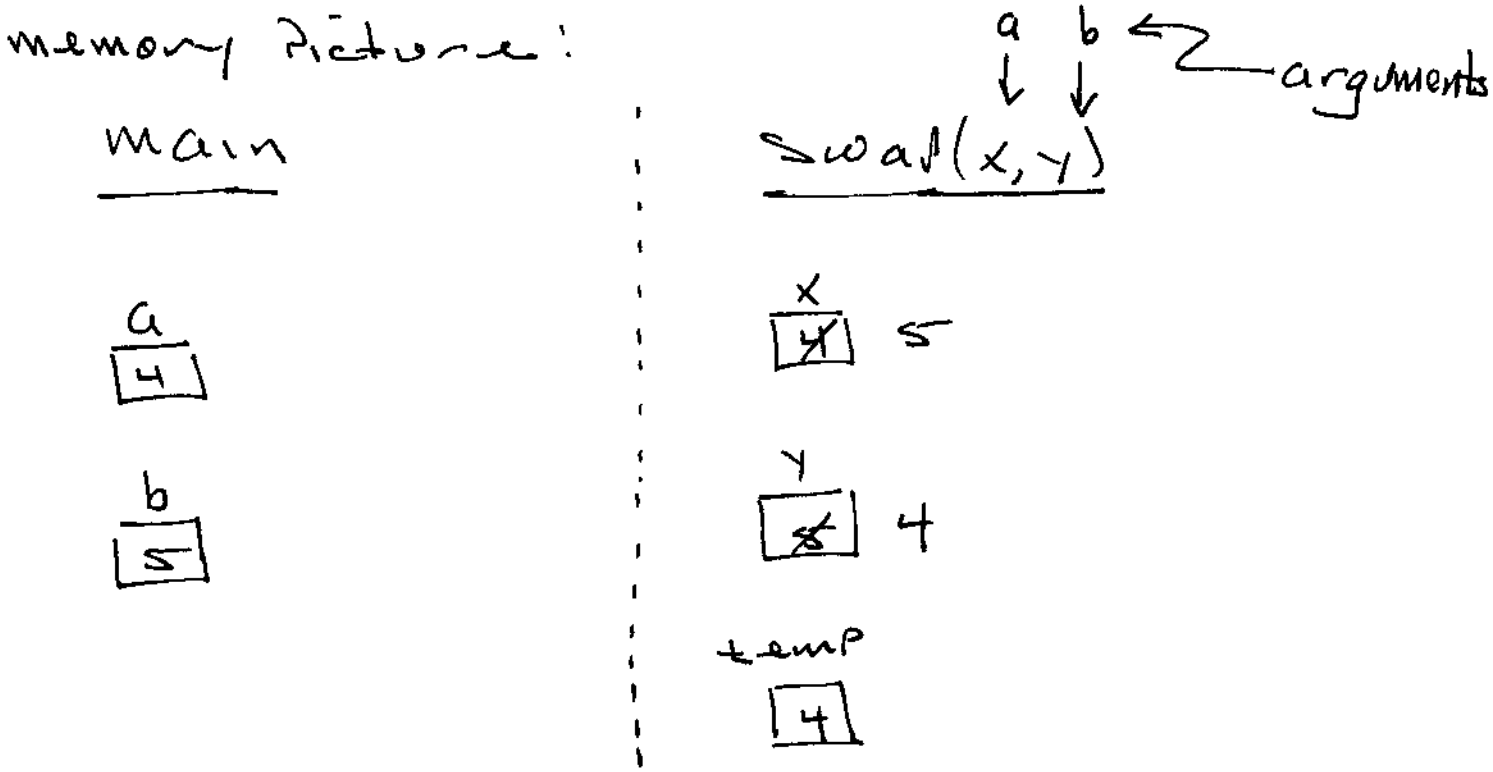
```
L = ['one', 'two', 'three']
L = [1, 2, 3]
L = ['one', 2, 3.0]
```

Ex. how not to swap two values.

```
def swap(x, y):
    temp = x
    x = y
    y = temp
#end swap()
```

why doesn't swap work?

memory picture:



when swap(a,b) is called, a is copied to x, and b is copied to y. The copies are swaped by

```
temp = x
   x = y
   y = temp
```

when swap() returns, the originals, a and b are unchanged.

A list can be used to create an effective swap function.

```
def swap(L, i, j):
    temp = L[i]
    L[i] = L[j]
    L[j] = temp
#end swap()
```

Now in main ------

```
A = [1, 4, 3, 5, 7]
swap(A, 1, 3)
print(A)
```

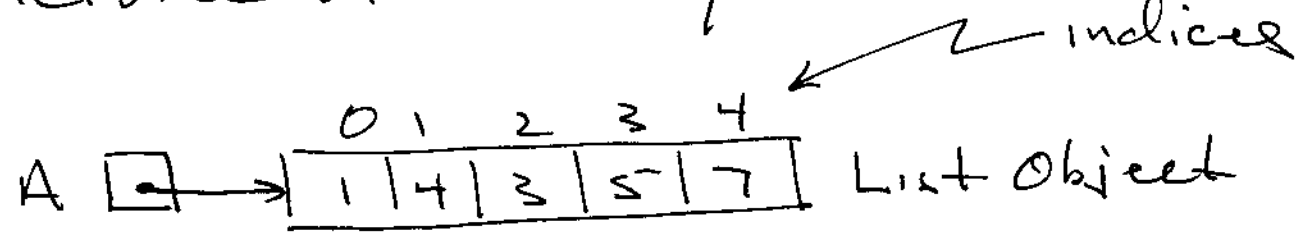
Output : [1, 5, 3, 4, 7]

swapped

How do List variables work?

A = [1, 4, 3, 5, 7]

Picture of memory:



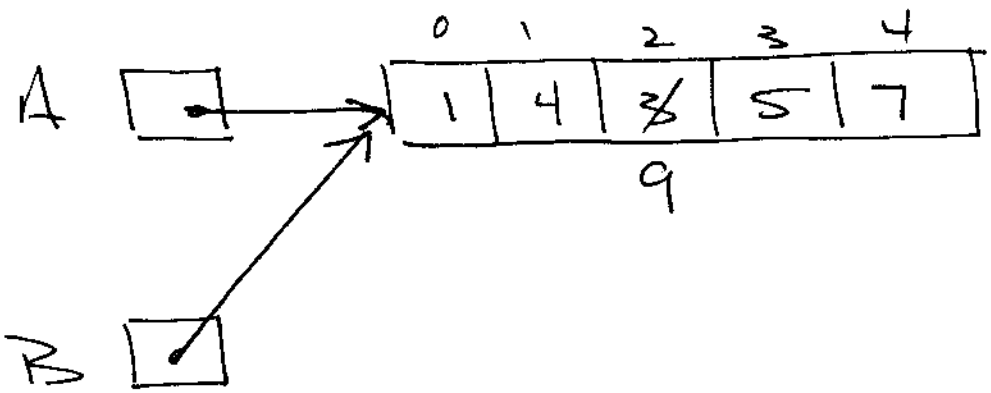
List reference variable

Now do

B = A

We have not created a new List object. instead we have 2 List reference variables storing the address of the same List Object.

Picture



Try this:

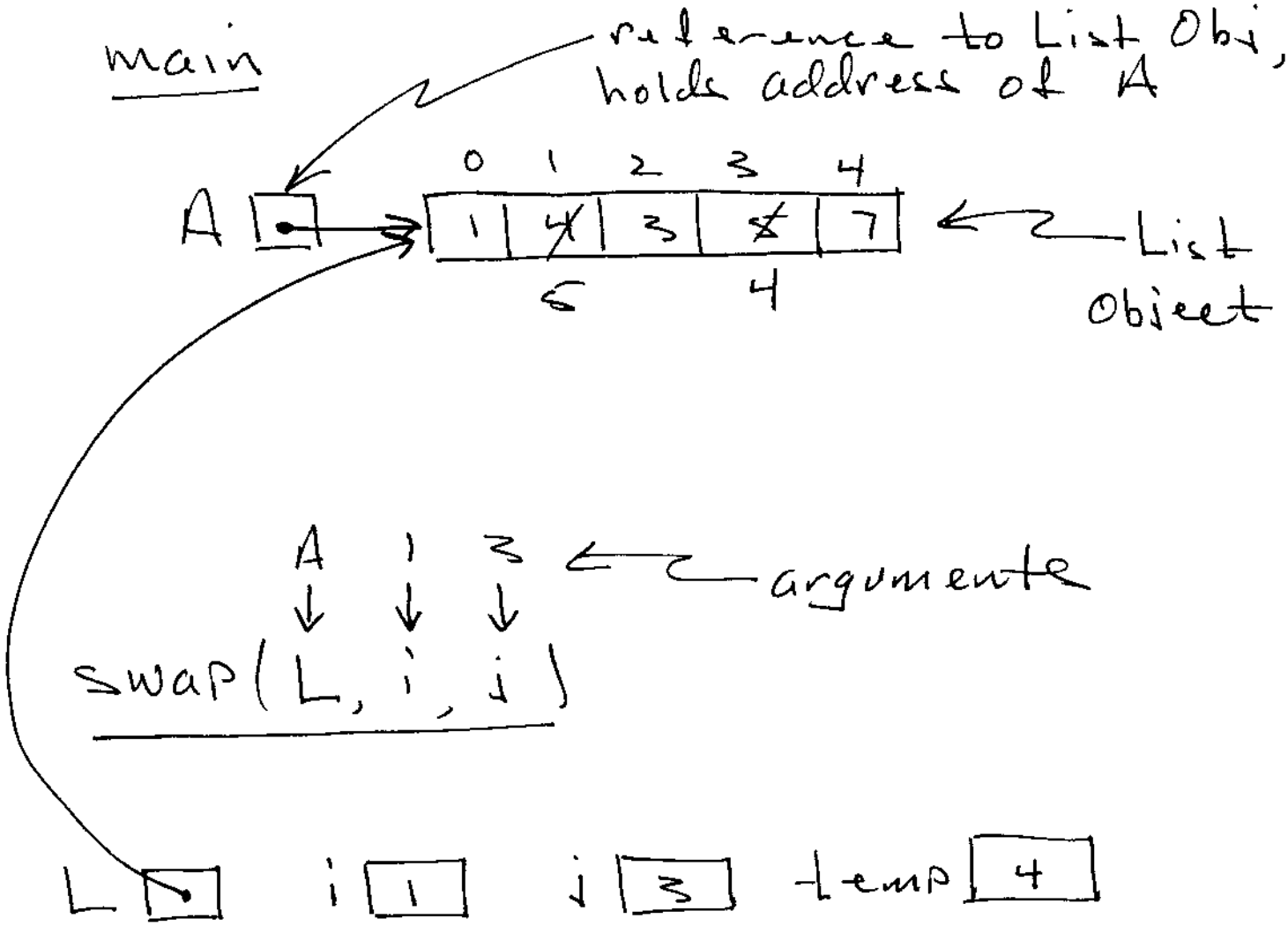
B[2] = 9

print(A[2]) # output 9, not 3

The reference variable B can now be used to alter the values in the list object.

Print(A) # output: [1, 4, 9, 5, 7]

why did the new swap() work ?
look at memory :



the body of swap() now Changes values in list A !!

```
temp = L[i]
L[i] = L[j]
L[j] = temp
```