

---

# Reference in Python

---

**IT5003: Data Structures and Algorithms  
(AY2019/20 Semester 1)**

# Python: Behavior of Variables

```
a = 123
```

```
b = a
```

```
b += 1
```

```
print(a) 123
```

```
a = [123]
```

```
b = a
```

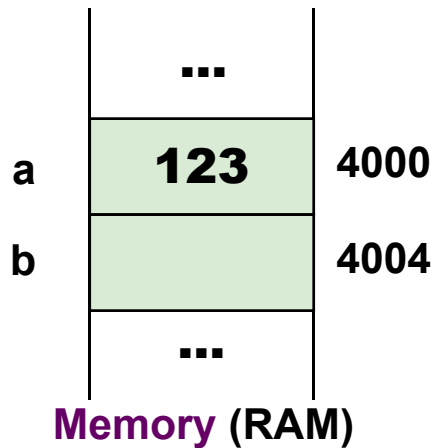
```
b[0] += 1
```

```
print(a) [124]
```

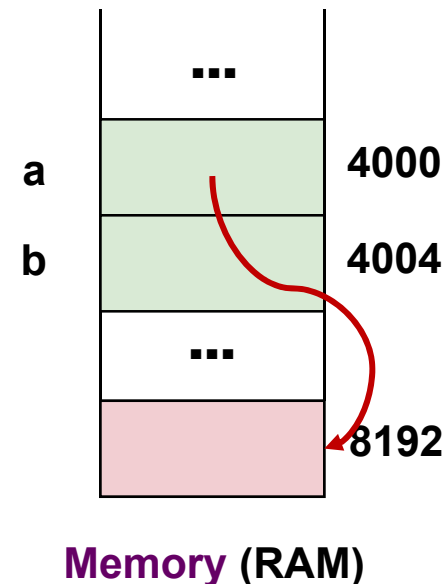
- What's happening here?
- In general, it is important to know how a language handles variable
  - e.g. Scope, Typing, Function Behavior etc

# Python: Everything is a Reference

- Instead of the more typical approach:
  - *"Variable name is attached to a memory location"*



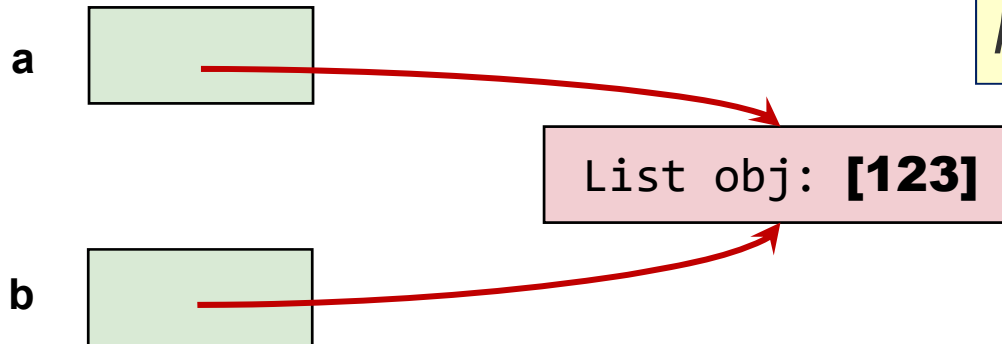
- **Python** uses the *"Variable is a reference to the actual object"* approach:



# Python: Behavior of Variables

- This is how "a" and "b" looks like after "b=a"

```
a = [123]  
b = a  
b[0] += 1  
print(a)
```

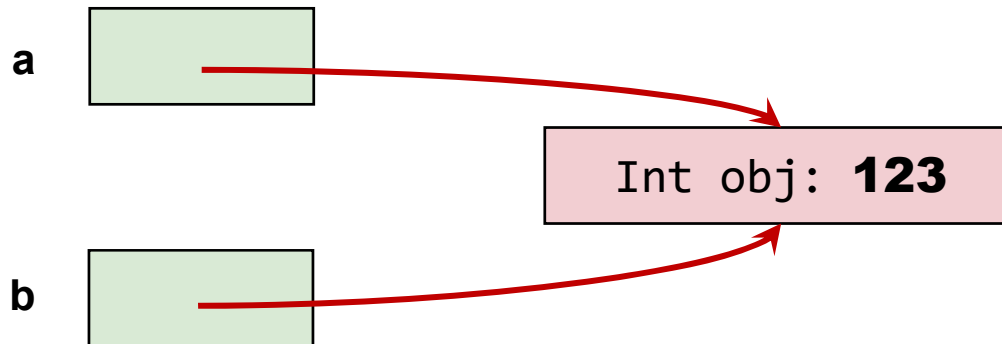


- Can you explain the behavior of the code now?

# Python: Behavior of Variables

- In addition, there are certain "objects" that are **immutable** (cannot be changed):
  - Integer, floating point, tuple, ..., etc
  - Modification **returns a new object**

```
a = 123  
b = a  
b += 1  
print(a)
```



- What happen when we execute "**b += 1**"?



END