Lecture

Mutability

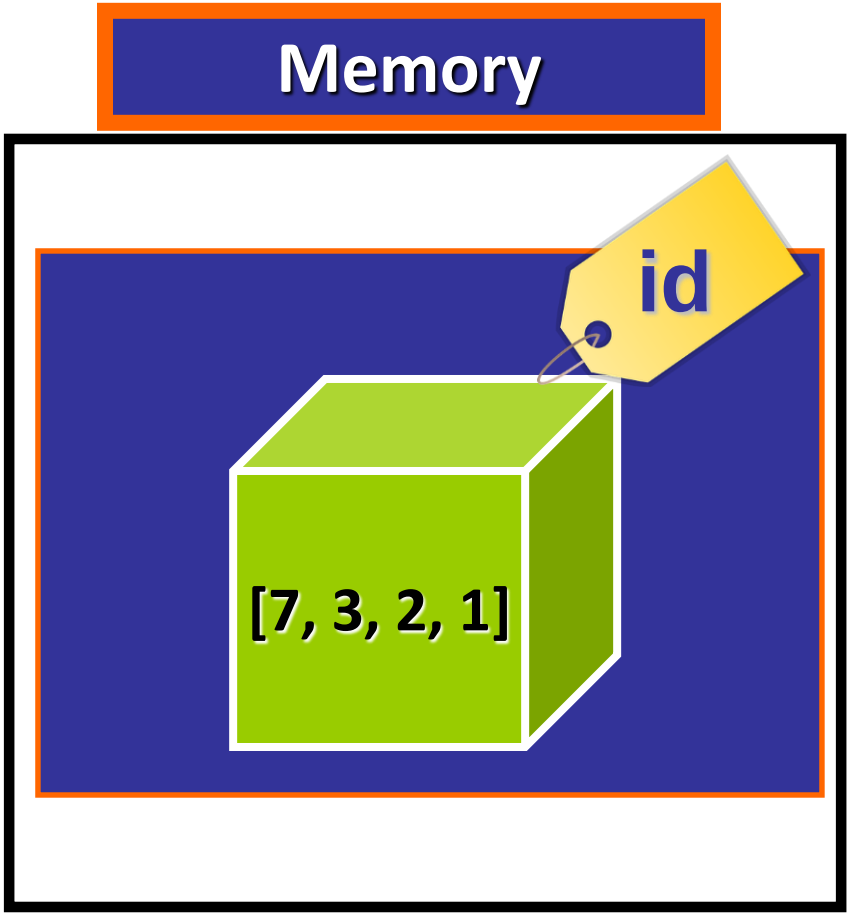
Advantages & Disadvantages

Mutable – Advantages

- **More Memory efficient**
  - Reuse existing objects instead of making new copies for every change.
- **Represent real-world objects that are mutable by nature**
  - e.g Player can change speed, location.

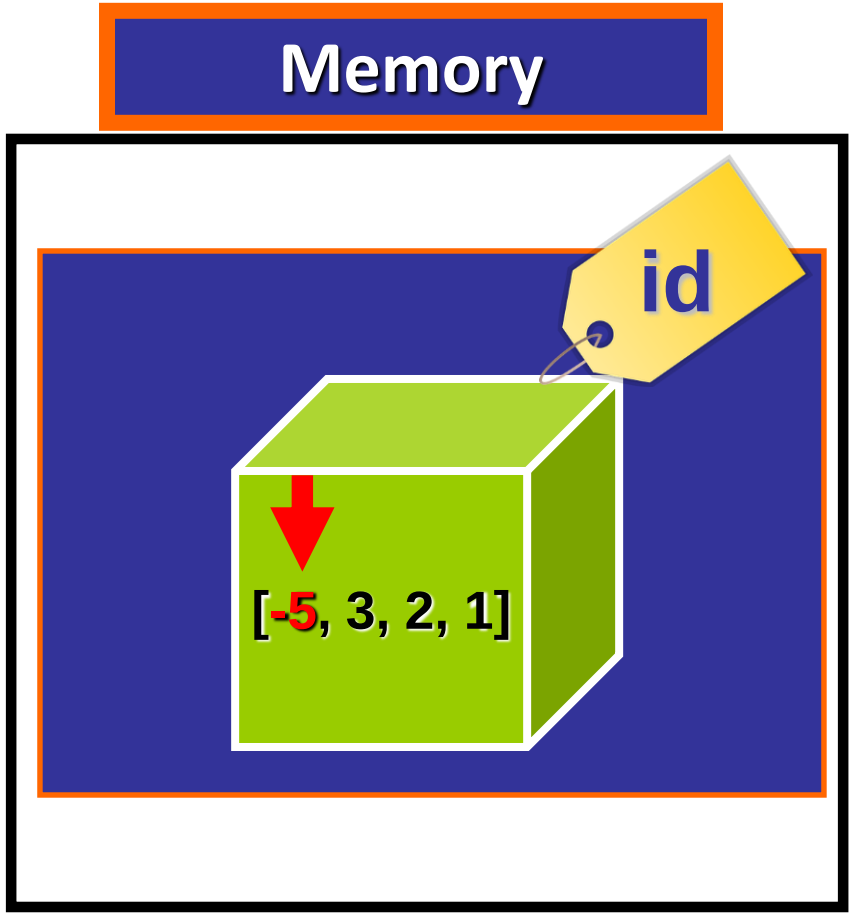
Mutable – Advantages

```
>>> a = [7, 3, 2, 1]
```



Mutable – Advantages

```
>>> a = [7, 3, 2, 1]
>>> a[0] = -5
>>> a
[-5, 3, 2, 1]
```



Mutable – Disadvantages

- **Bugs** 
  - Mutable objects are more likely to introduce bugs in a program.
  - You can unintentionally mutate the original object in a function or in a loop.

Mutable – Disadvantages

```
def sum_of_abs_value(lst):

    for i in range(len(lst)):
        lst[i] = abs(lst[i])

    return sum(lst)


numbers = [-3, -2, -6, 2, 5, 1]

print("Before:", numbers)

print("Sum of the absolute values:", sum_of_abs_value(numbers))

print("After:", numbers)
print("The list was mutated!")
```

Mutable – Disadvantages

```
def sum_of_abs_value(lst):

    for i in range(len(lst)):
        lst[i] = abs(lst[i])

    return sum(lst)


numbers = [-3, -2, -6, 2, 5, 1]

print("Before:", numbers)

print("Sum of the absolute values:", sum_of_abs_value(numbers))

print("After:", numbers)
print("The list was mutated!")
```

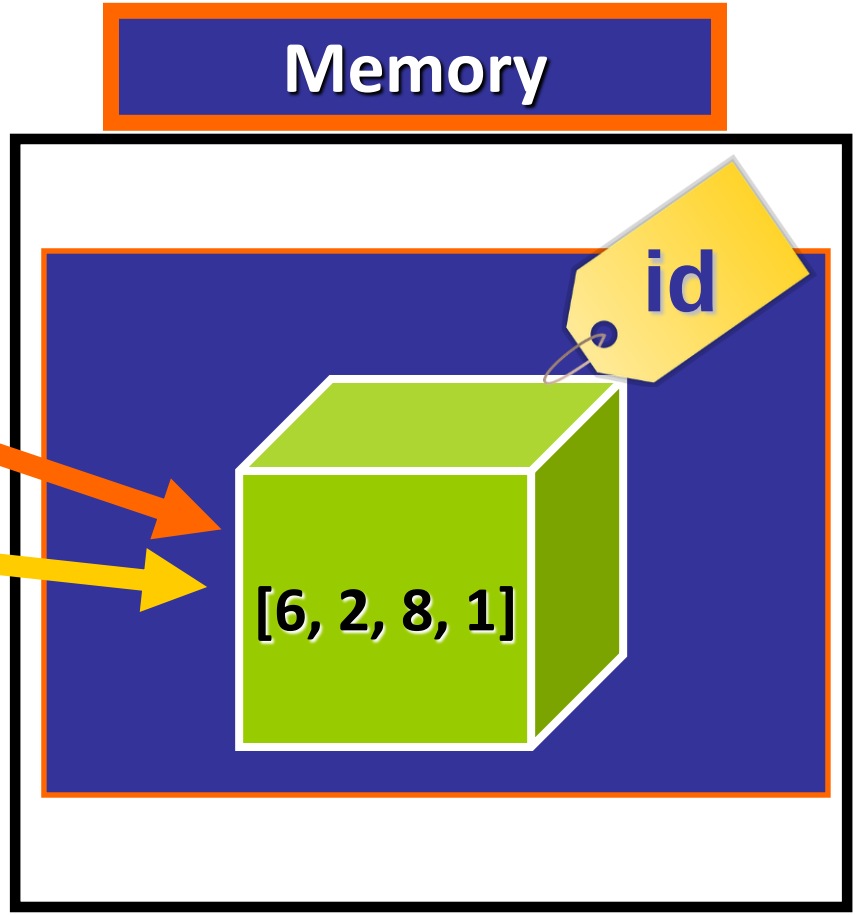
```
Before: [-3, -2, -6, 2, 5, 1]
Sum of the absolute values: 19
After: [3, 2, 6, 2, 5, 1]
The list was mutated!
```

Mutable – Disadvantages

Aliasing

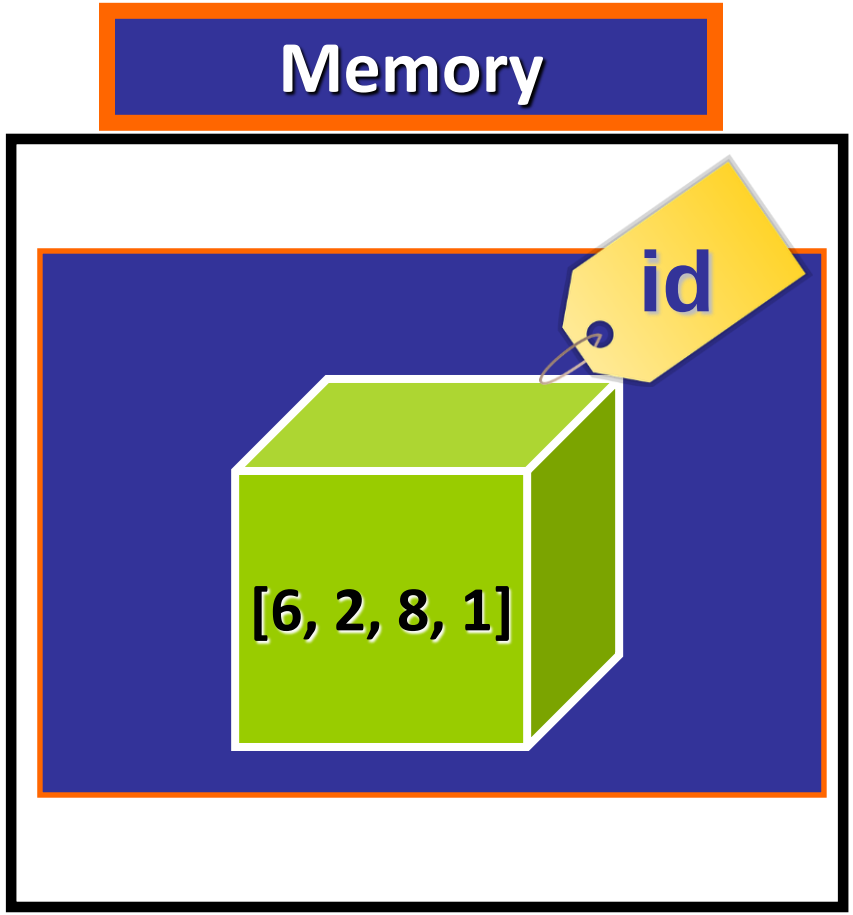
Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
```



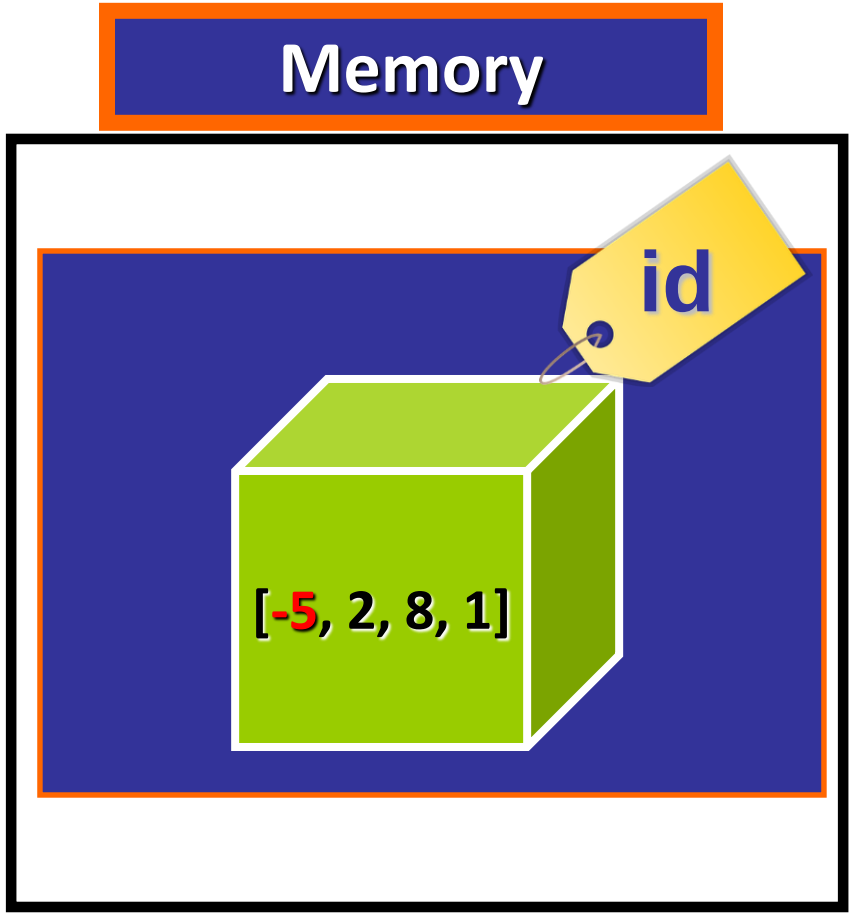
Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
>>> b[0] = -5
>>> b
[-5, 2, 8, 1]
```



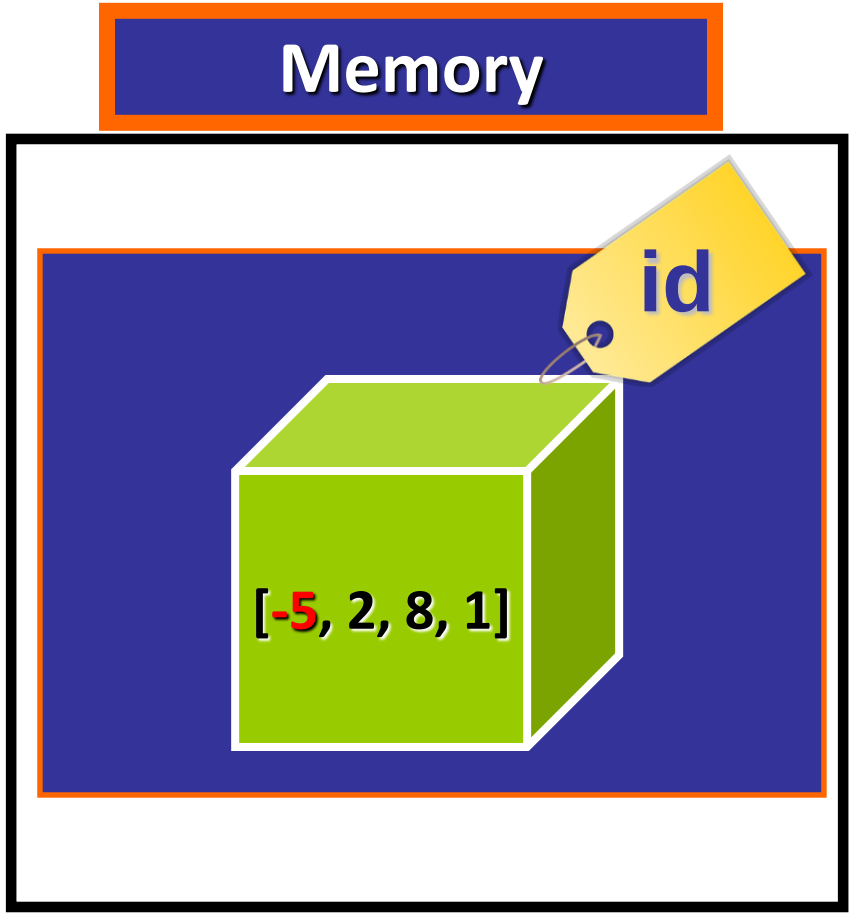
Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
>>> b[0] = -5
>>> b
[-5, 2, 8, 1]
```



Mutable – Disadvantages

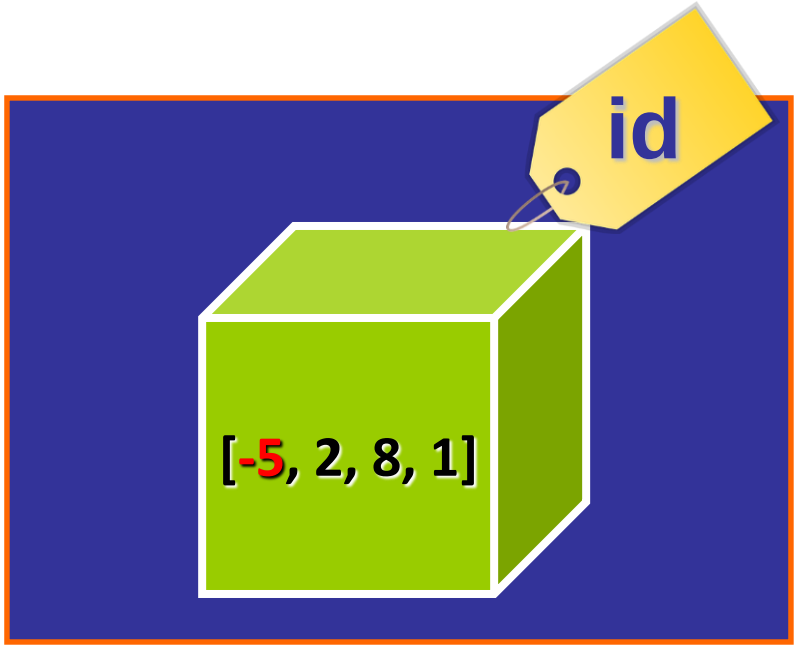
```
>>> a = [6, 2, 8, 1]
>>> b = a
>>> b[0] = -5
>>> b
[-5, 2, 8, 1]
```



Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
>>>
>>> a = [6, 2, 8, 1]
>>> b = a
>>> b[0] = -5
>>> b
[-5, 2, 8, 1]
>>> a
[-5, 2, 8, 1]
```

Memory

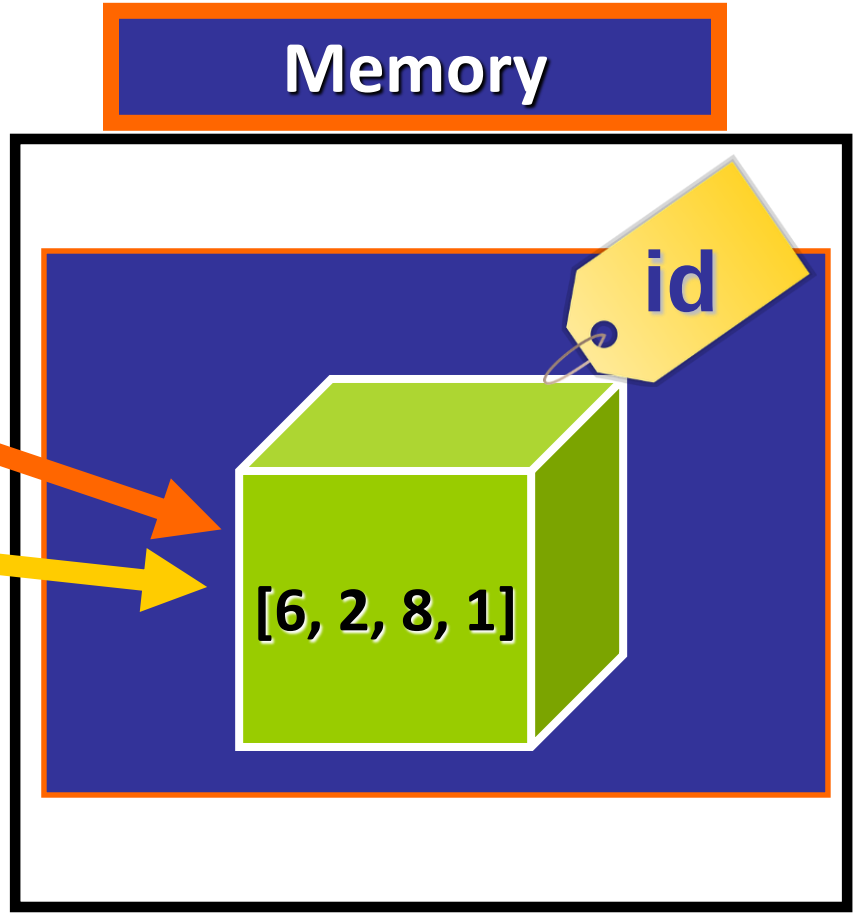


Mutable – Disadvantages

a and b are aliases

Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
```



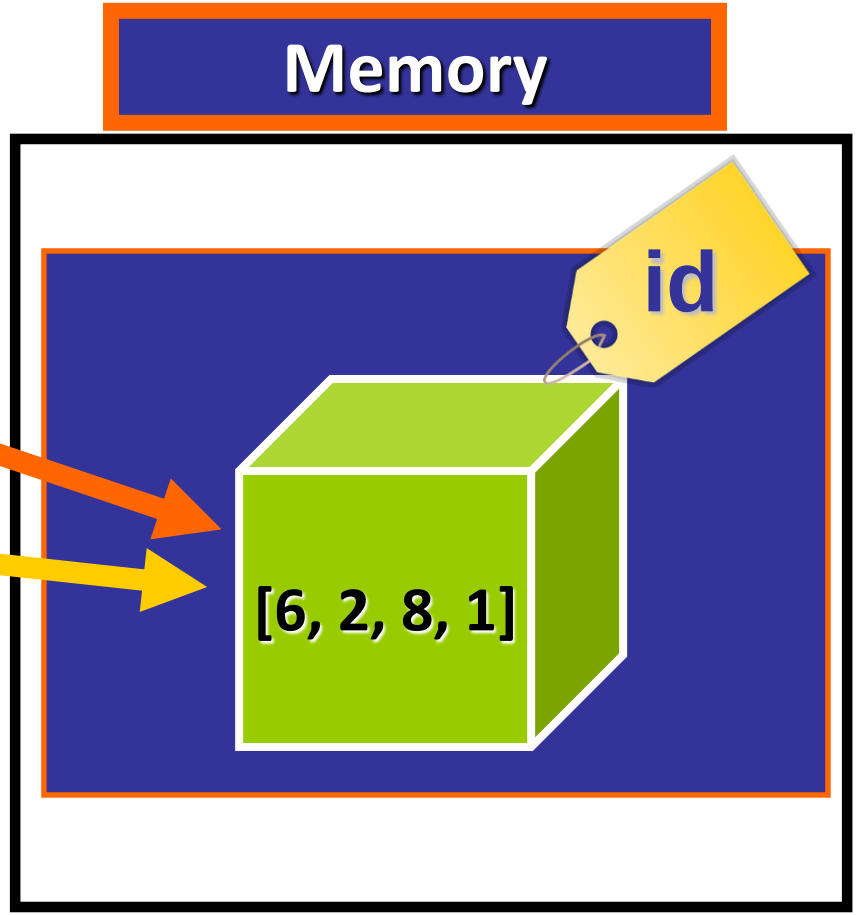
Mutable – Disadvantages

Changing one

Changes the other

Mutable – Disadvantages

```
>>> a = [6, 2, 8, 1]
>>> b = a
```



Immutable – Advantages

- **Safer from Bugs**
- **Easier to Understand**
  - Know clearly what your program is doing.
  - No “hidden” changes to the objects.

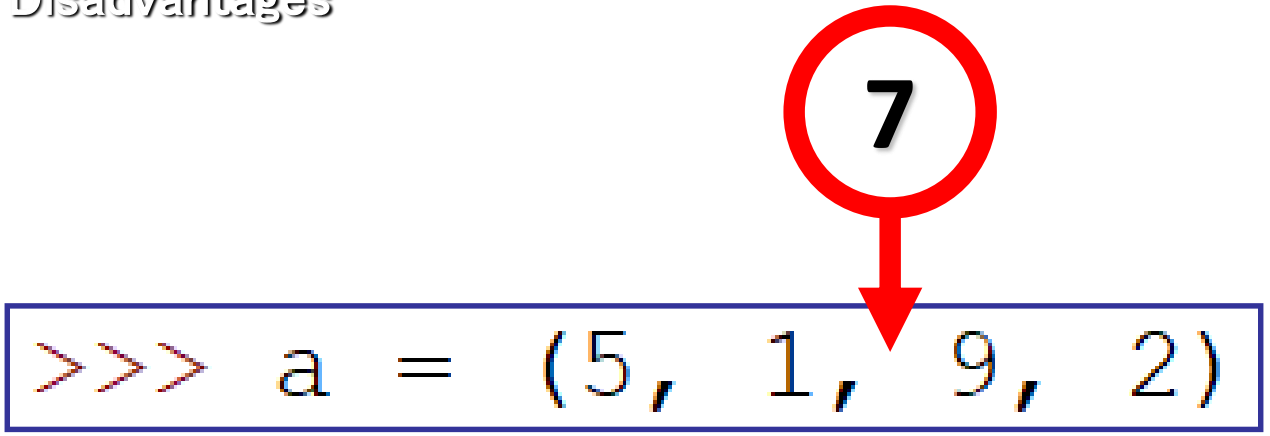
Immutable – Disadvantages

- **Can be less memory efficient**
  - You need to create a new object for every change.
  - Costly when the object is large or complex.

Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
```

Immutable – Disadvantages



Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

Immutable – Disadvantages

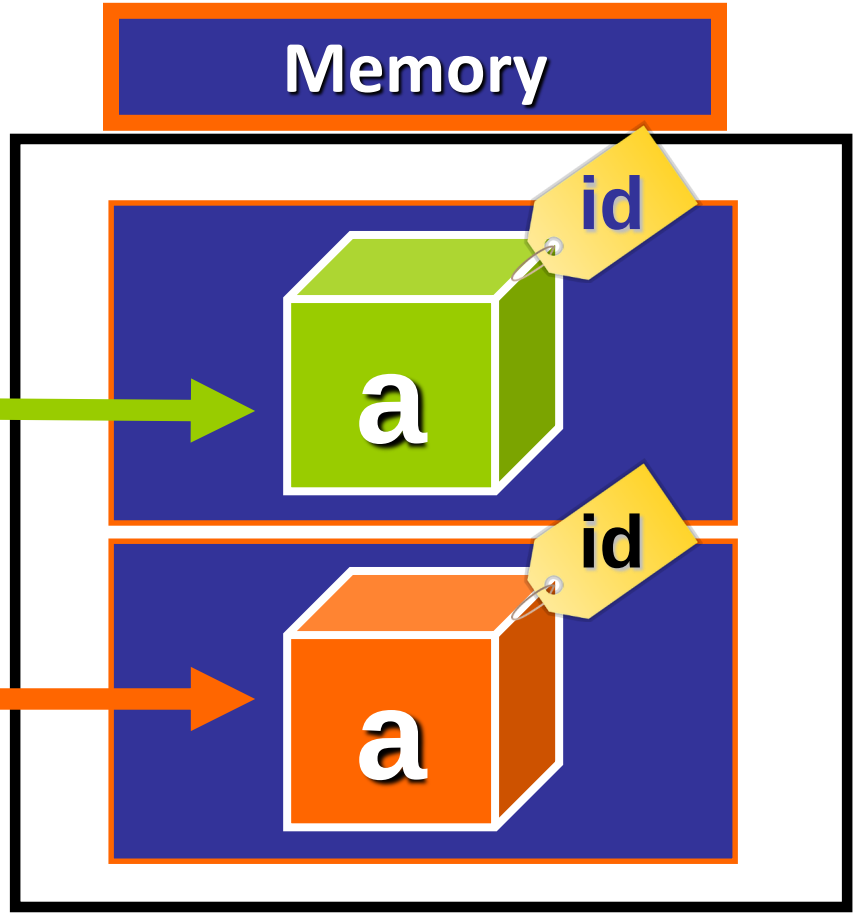
```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```

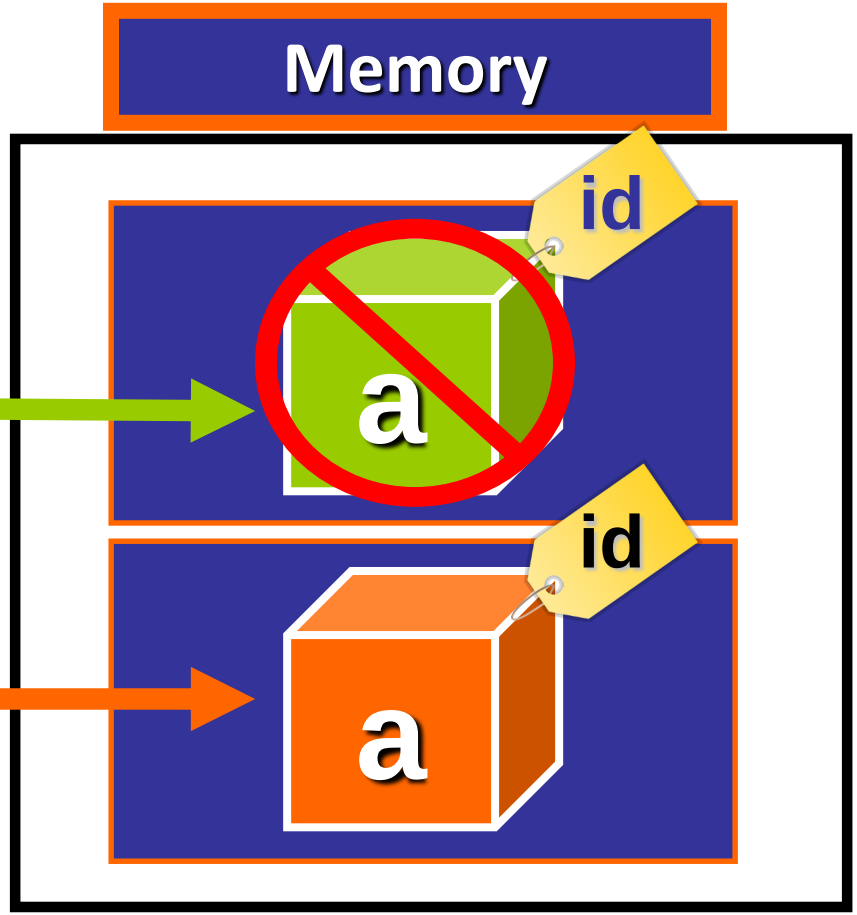
Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```



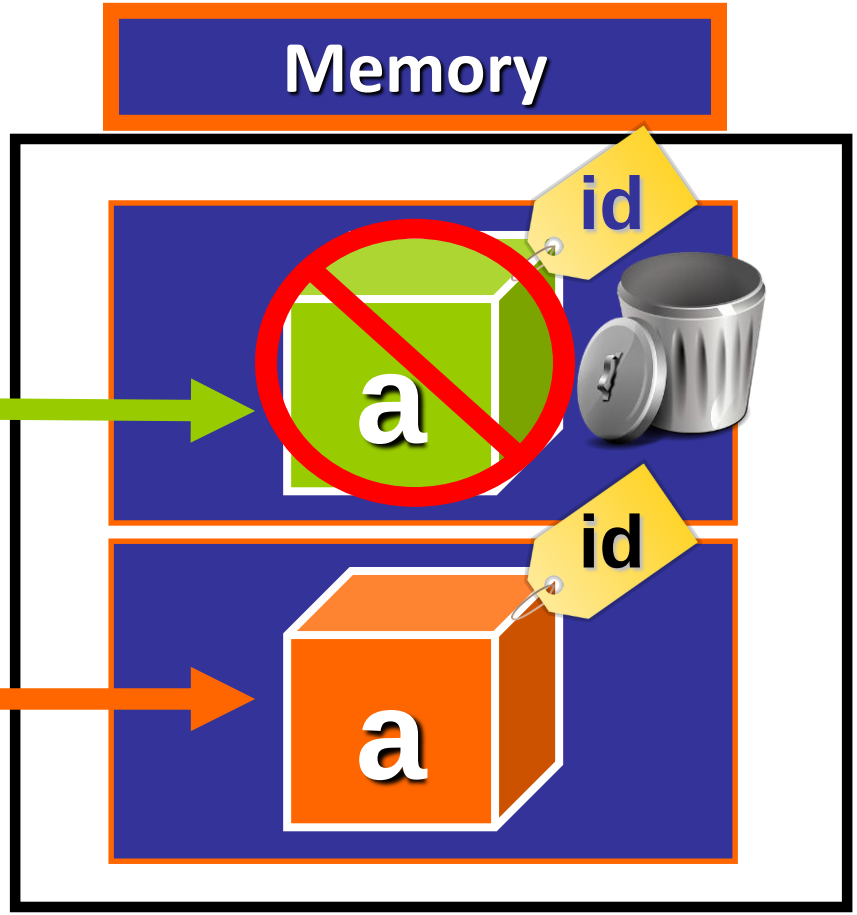
Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```



Immutable – Disadvantages

```
>>> a = (5, 1, 9, 2)
>>> id(a)
54119912
>>> a = a[:2] + (7,) + a[2:]
>>> a
(5, 1, 7, 9, 2)
>>> id(a)
57463024
```



Now… An Example