9. As you just saw, ***lists are mutable*** – the value of the object itself (that is, its "insides") can change.

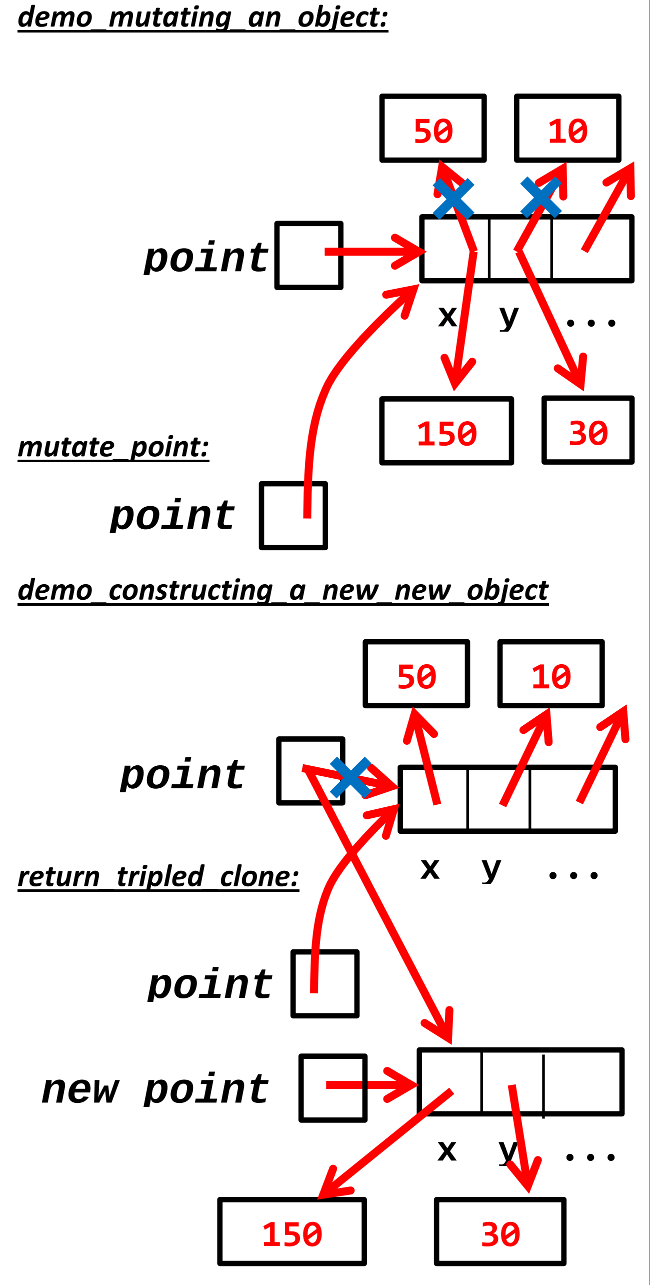
   Tuples are ***NOT mutable*** – that is their primary difference from lists. ***Strings*** are ***NOT mutable*** and ***numbers*** are ***NOT mutable***.

   Instances of user-defined classes (like the Zellegraphics objects) ***are, in general, mutable***.

   To see this, draw a Box and Pointer diagram that shows what happens when *main* (below) executes. Also show the output that is printed.

```
def main():
    demo_mutating_an_object()
    demo_constructing_a_new_object()

def demo_mutating_an_object():
    point = zg.Point(50, 10)
    mutate_point(point)
    print('A.', point)

def mutate_point(point):
    point.x = point.x * 3
    point.y = point.y * 3

def demo_constructing_a_new_object():
    point = zg.Point(50, 10)
    point = return_tripled_clone(point)
    print('B.', point)

def return_tripled_clone(point):
    new_point = zg.Point(point.x * 3,
                         point.y * 3)
    return new_point
```

Box and Pointer diagram:



Output:

A. Point(150, 30)

B. Point(150, 30)

mutate_point and *return_tripled_clone* both end up with a tripled point. ***Which one uses less storage?*** [✓ *mutate_point*] *return_tripled_clone* (circle your choice)