
07-Lists and Functions

Unknown Author

August 20, 2013

1 Lists and Functions

It's not really clear that this warrants a mention as a separate topic. But, it is a great integration of two important topics. Below, we just point out one gotcha.

We're used to functions having no effect on the variables that we pass in:

```
In [1]: x = 3

def add_3(val):
    val = val + 3
    return val

print add_3(x)
# Above, the function only modified it's own variable, so x stays the same
print x

6
3
```

But once we start using **mutable** data types like lists, things become tricky:

```
In [2]: x = [1, 2, 3, 5]

def add_3(val):
    val[2] = val[2] + 3
    return val

print add_3(x)
# Now, our function is modifying the contents of the list, and both variables still po
# So the list x points to *is* modified
print x

[1, 2, 6, 5]
[1, 2, 6, 5]
```

So, the issue here is our function is no longer changing `val` so that it points at a new "thing." Instead, we're taking the list that `val` points to (the same list `x` points to) and modifying it.

Tricky, but important!