

Discussion 5: Mutability

What are the values of the script variables x and y after the given script finishes running?

a.

```

script variables x y
set x to 123
set y to list 1 2 3
set x to six
set y to six
    
```

```

+set+ input +to+ six+
set input to 6
    
```

x: 123
 y: [1,2,3]

b.

```

script variables x y
set x to 123
set y to list 1 2 3
add six to x
add six to y
    
```

```

+add+ six+ to+ input+
if is input a list?
add 6 to input
else
change input by 6
    
```

x: 123
 y: [1,2,3,6]


Challenge: Errors Galore

The following questions are based off this block:

```

+square+ numbers+ in+ list+
script variables new
set new to list
for i = 1 to length of new
replace item i of new with
item i of new x item i of new
    
```

Inside: Error reporter didn't report

a. When running this block, we get the following error:  What caused this error, and how can we fix it?

The error is due to not reporting at the end of the block definition. We can append [report (new)]

after the for loop and the block will work.

b. Now, that we've fixed it, we try to test our code, but we get another error:



A Scratch test block with the following structure: `test` block containing `square numbers in []` with a dropdown menu, `w/inputs` block containing `list` block with `1` and `2`, and `expecting output` block containing `list` block with `1` and `4`. A speech bubble on the right says "Inside: TypeError list.length is not a function".

How can we fix this?

The input list (`list [1] [2]`) must be nested in a list.

c. Now, we try running the following test, but it doesn't work as expected:



Scratch code blocks: `set` block with `my list` dropdown and `to` block containing `list` block with `1` and `2`. Below it is a `report` block containing a `test` block with `square numbers in []` dropdown, `w/inputs` block with `list` block containing `my list` and `list` block containing `my list`, and `expecting output` block with `list` block containing `1` and `4`, and another `list` block containing `1` and `4`. A speech bubble on the right shows a list with two items: `1` with a `true` toggle and `2` with a `false` toggle.

Why does it report this, and how can we fix it?

The first time we call (`square numbers in []`) on (`my list`) we are mutating (`my list`) so it becomes (`list [1] [4]`).

When we call (`square numbers in []`) on (`my list`) again we then get (`list [1] [16]`) which is not equal to the expected output (`list [1] [4]`). To fix this inside the (`square numbers in []`) block we can set (`new`) to (`copy of (list)`).
