- Let's consider:

```
(((lambda (x) (x 3)) (lambda (y) (lambda (x) (* x x)))) 4)
```

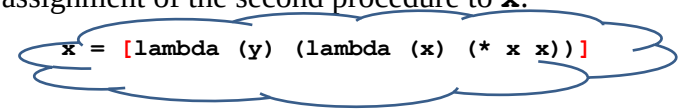
At the top level, we have two elements, indicated in green brackets and a green number:

```
([(lambda (x) (x 3)) (lambda (y) (lambda (x) (* x x)))] 4)
```

- Looking at the green bracketed expression, we see two lambda expressions indicated in red:

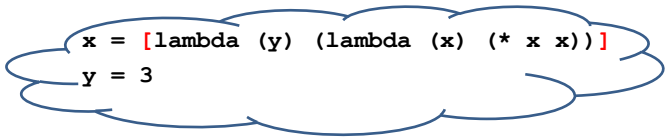
```
[[lambda (x) (x 3)] [lambda (y) (lambda (x) (* x x))]]
```

- Both evaluate to procedures.
- We then evaluate the application expression, resulting in an assignment of the second procedure to **x**:



- We then evaluate the body of the first procedure, i.e.: `(x 3)`

- This results in an assignment of 3 to y:



- We now evaluate the body of the second procedure, i.e. `(lambda (x) (* x x))`
- This value is a procedure and get's to be returned.
- The original expression now looks like this:

```
((lambda (x) (* x x)) 4)
```

- The rest is straight-forward.