if/let assignment → individual lambdas in other languages
globals in Python

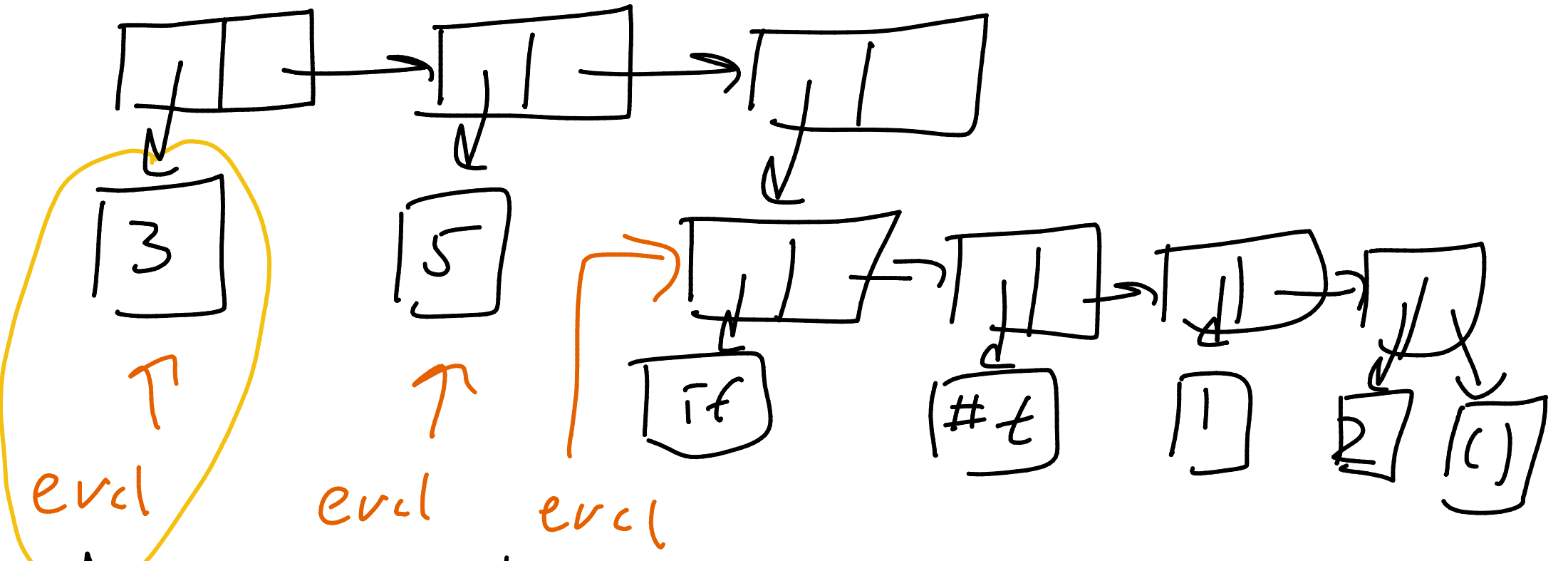
(if #t 3 5) ⇒ 3

```
(let ((x 3) (y 5))
  x)
```

3

5

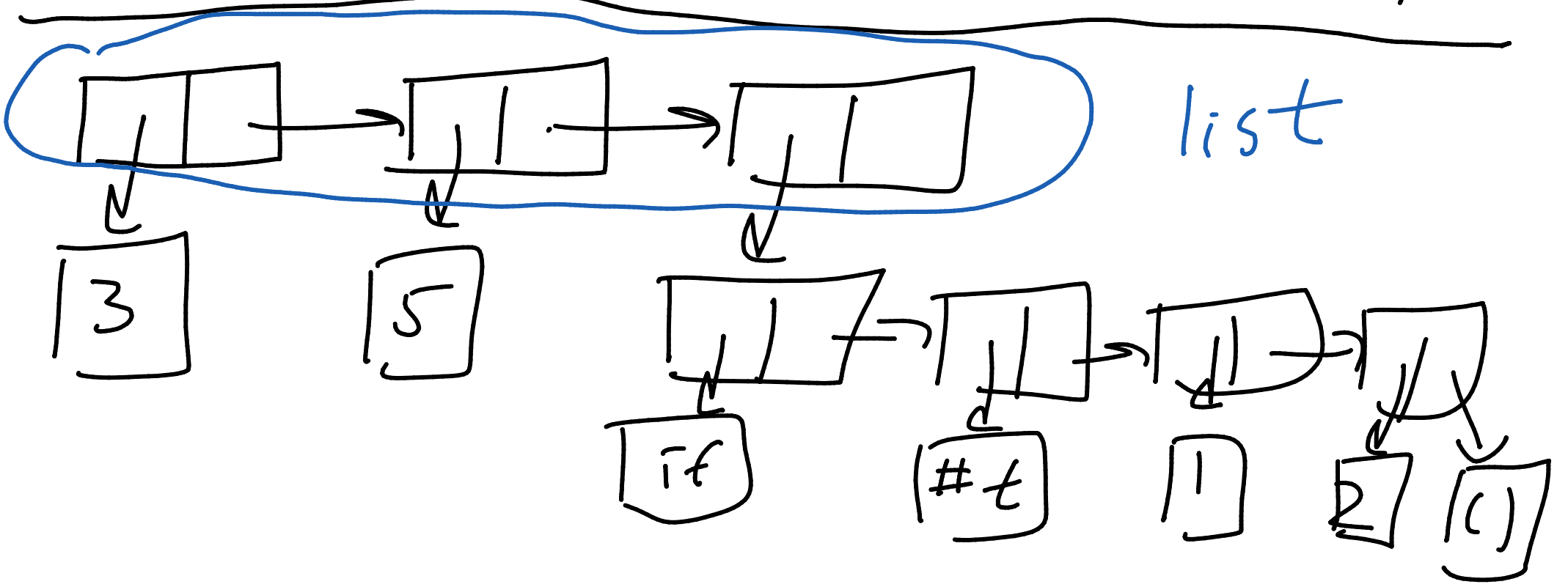
(if #t 1 2)



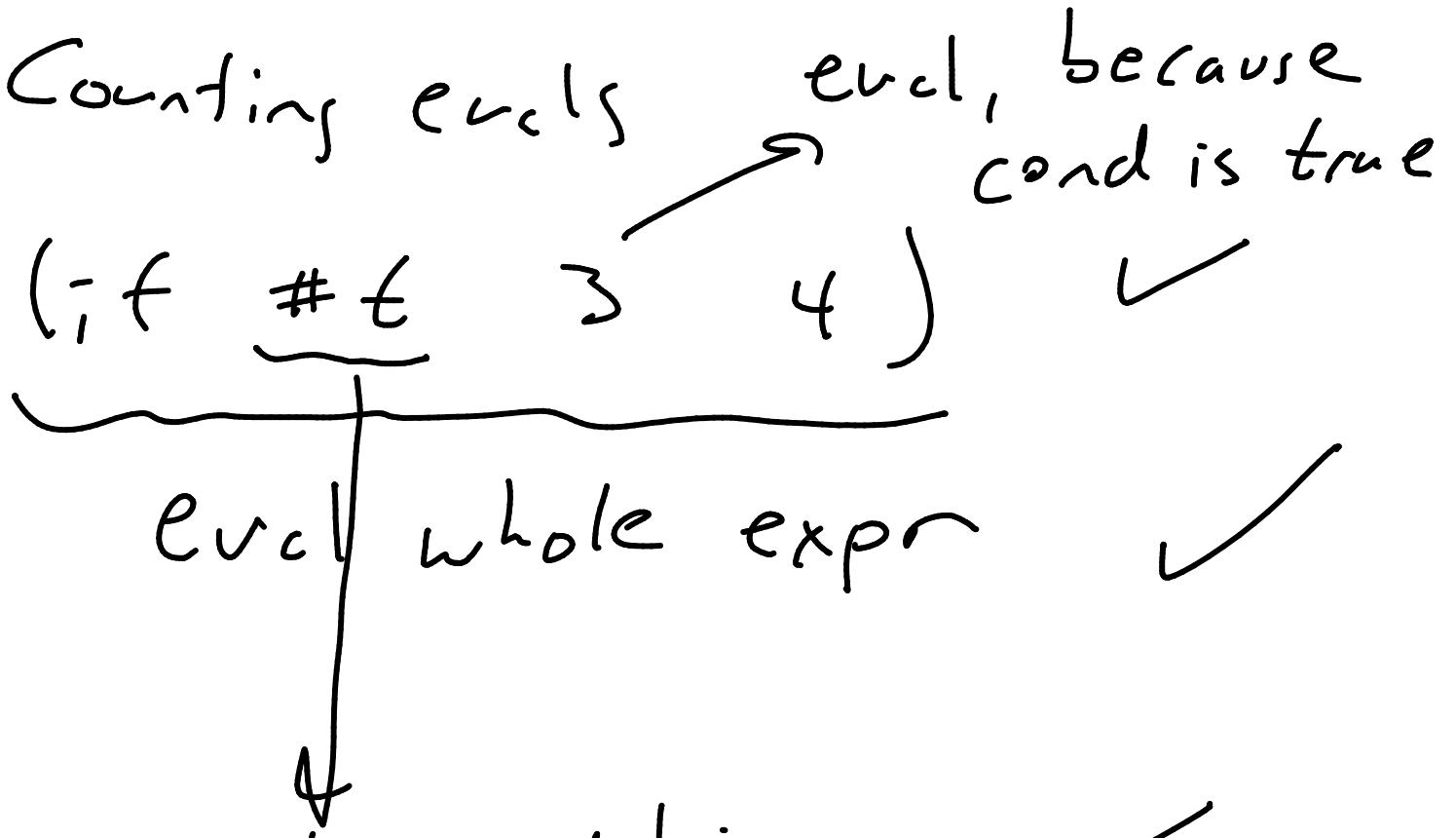
loop over top list {

eval each expr inside

}



Counting evals



if does "short-circuit" evaluation
- doesn't evcl both exprs that follow

when the cond is #t,
evaluate the first following
- - - - - #t
- - - - - second

⇝ evals

Q: evcl If gets a list that looks like

↙ just a list (instead of parms

Q: don't I eval the cons type on top?

No, because I am not treating that as a Scheme expr of its own to be evaluated

If I did, that's evaluating

(#t 3 4)

→ wrong type to apply error

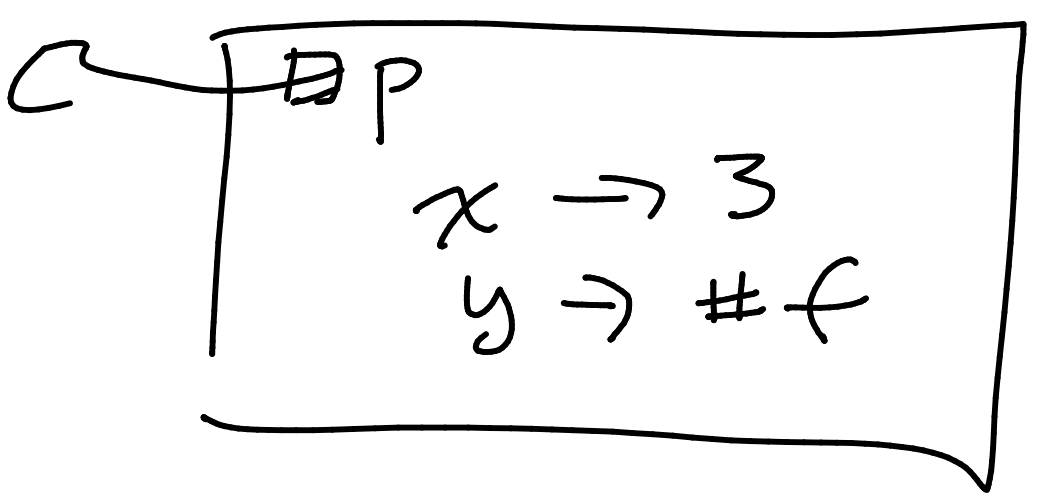
```
(let ((x 3) (y #f))
  (if y x x))
```

(2) (3) (5) (6) (4) (1)

let evals it

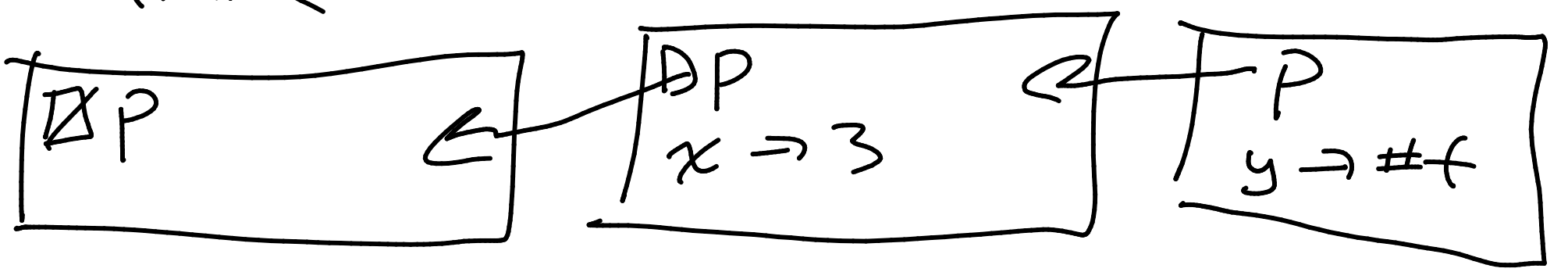
eval whole expr

let creates a frame

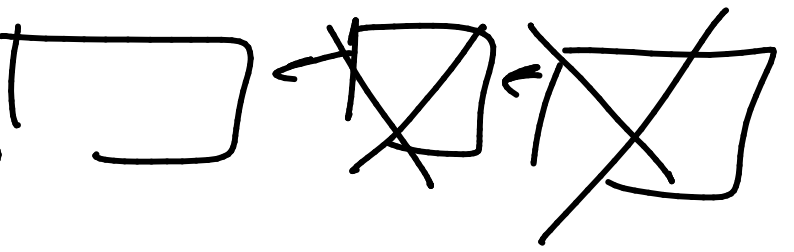


```
(let ((x 3))
  (let ((y #f))
    (if y x x)))
```

global frame



```
(let ((x 3))
  (let ((y #f))
    (if y x x)))
```



```
(let ((z 9))
  (if #t z 0))
```

```
z => 9
```

2 frames active in second if

Q: multiple exprs?

A: yes

```
(let ((x 3))
  (set! x 7)
  (+ x 3))
```