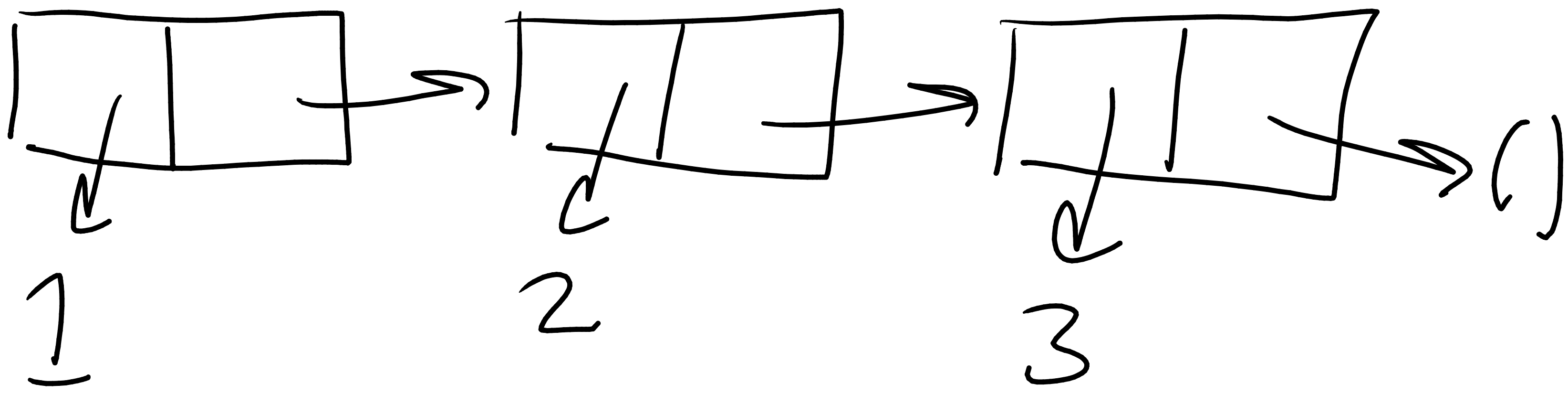
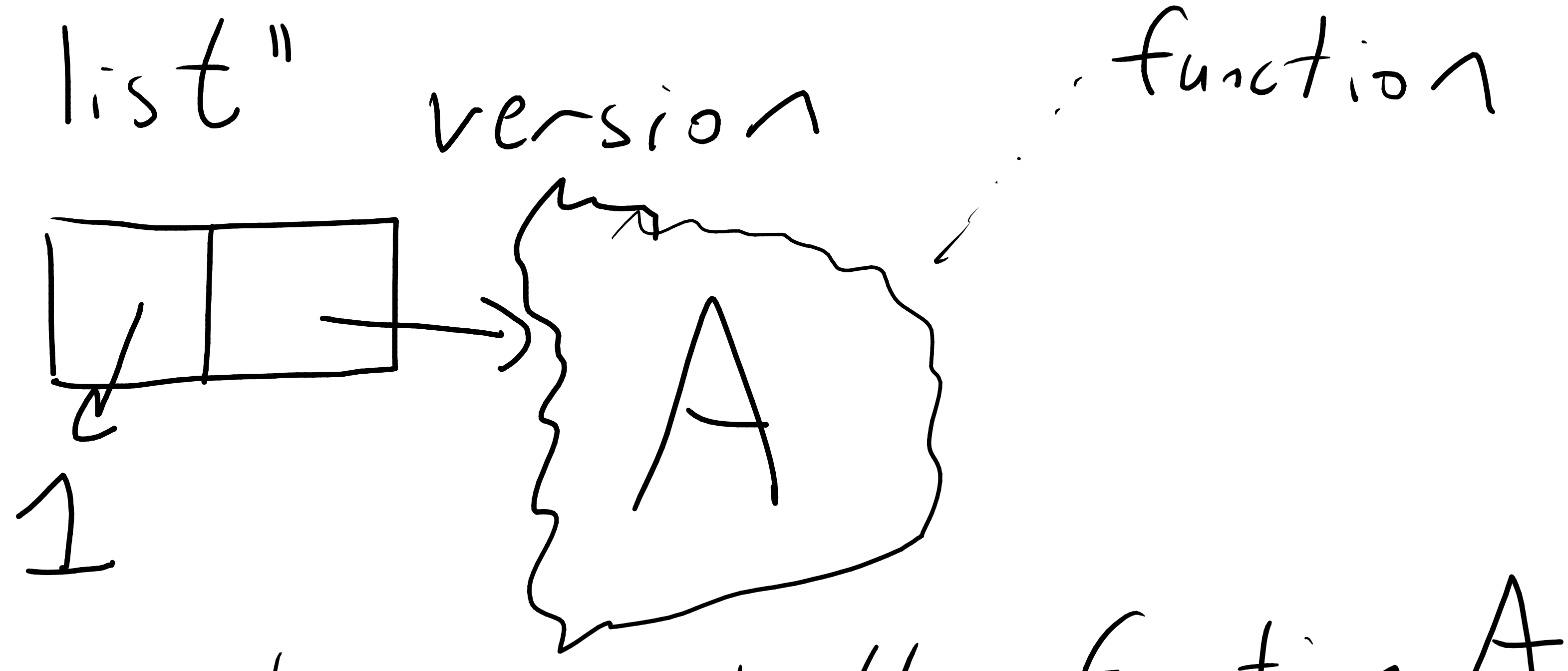
→Lazy lists (last big Scheme piece)

→ functional programming in general

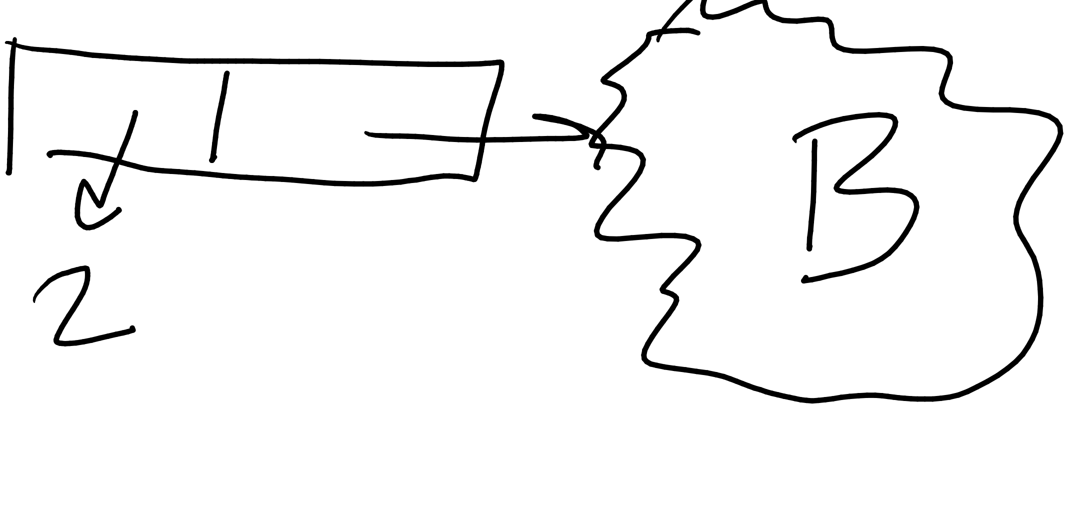
Regular Scheme list (1 2 3)



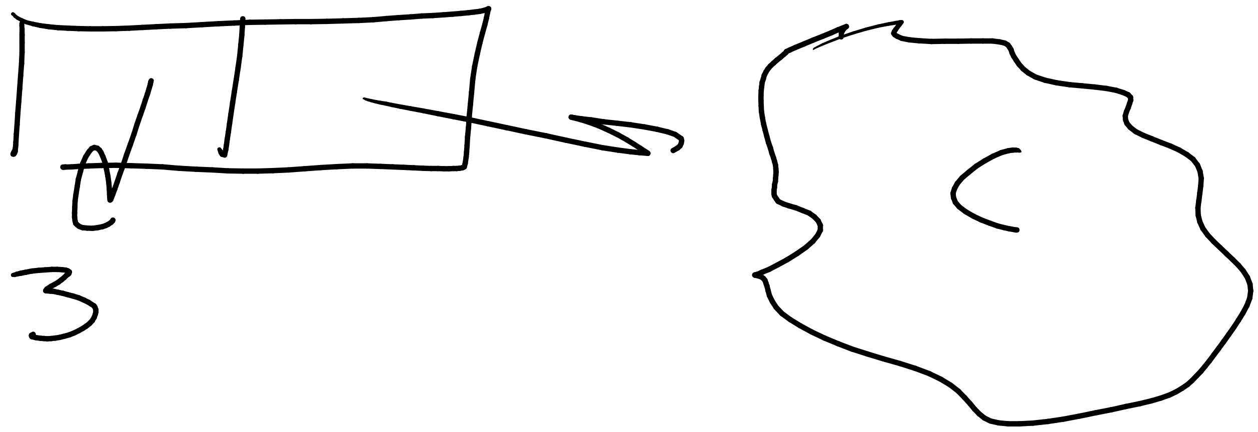
"Lazy list" version



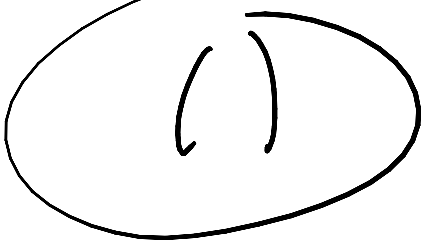
when I call this function, A it returns



When I call B, I get back



When I call C, I get back



Programming paradigms
4 most popular / famous

(1) Imperative

- Program is a series of steps
- each step modifies "state" (memory, etc) in some way

(most non obj-oriented programming you've done)

(2) Object-oriented

(some people consider this a superset of (1)

- obj orientation.
- objects storing, communicating, etc

(3) Logical

- Program is a series of statements of logic, and then the computer does logical deduction (Prolog)

④ Functional

Program is a series of function calls, which take parameters and return values **

Which one of these is Python?
Multiple answers people give.

? ②.

? ②, but ④, ③

I prefer that the question was meaningless.

Instead, we should say

"Python supports multiple programming paradigms, (Dominantly OO.....)"

One import aspect of functional paradigm is the desire to avoid side effects.

A function call has a side effect if it makes any kind of observable change or interaction except for the return value.

$x = 3$ side effect (???)

print("hello") side effect (???)

Why does it matter?

You can modify the order (in some cases) that work is done if you have no side effects, so long as return values are the same