Scheme and how it works
↳ built around lists

(1 2 3)

In Scheme data and code is (almost) interchangeable

notation	American	British
( )	paren	brackets
[ ]	brackets	sq. brackets
{ }	(curly) braces	curly brackets

(+ 1 2) → 3

making a function call

(* 3 5) → 15

In Scheme, I use a ' to indicate that a list should not be executed car - first item in a list

cdr - everything except first item

(car '(a b c))

↙

a   item

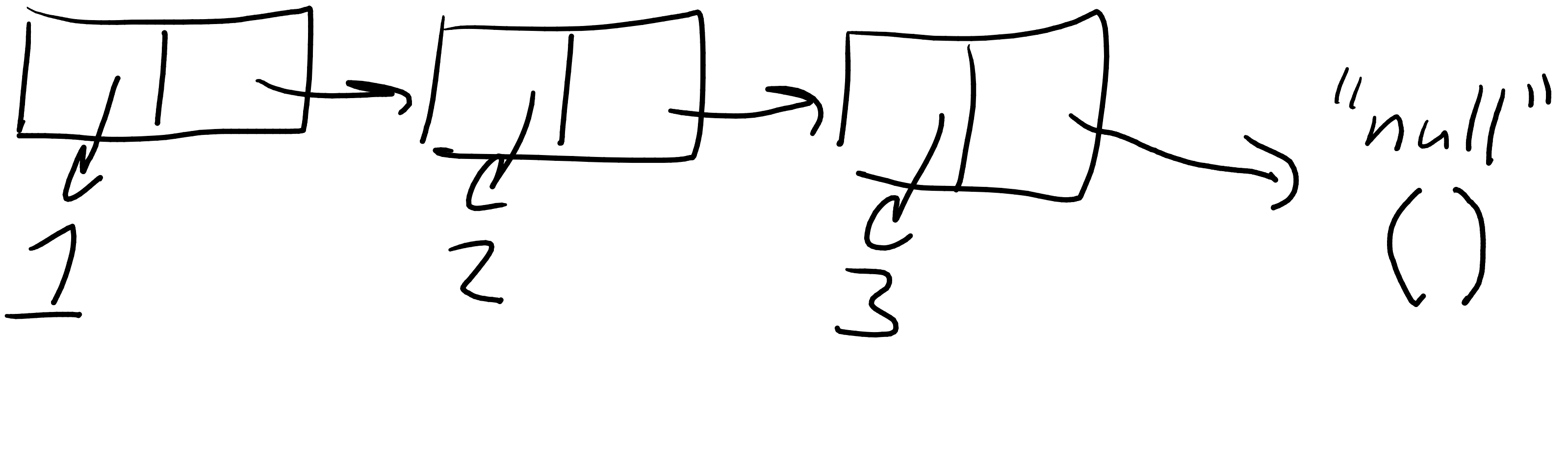
(cdr '(a b c))

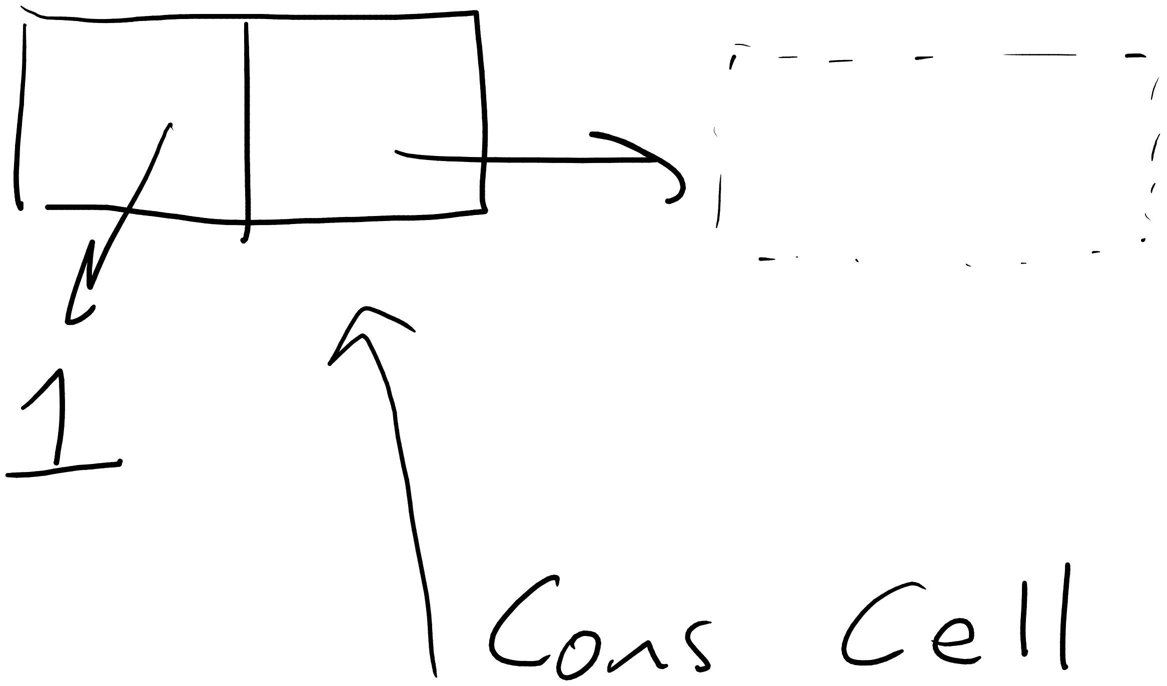
d

(b c)   list

Underneath, all lists in Scheme are linked lists

(1 2 3)

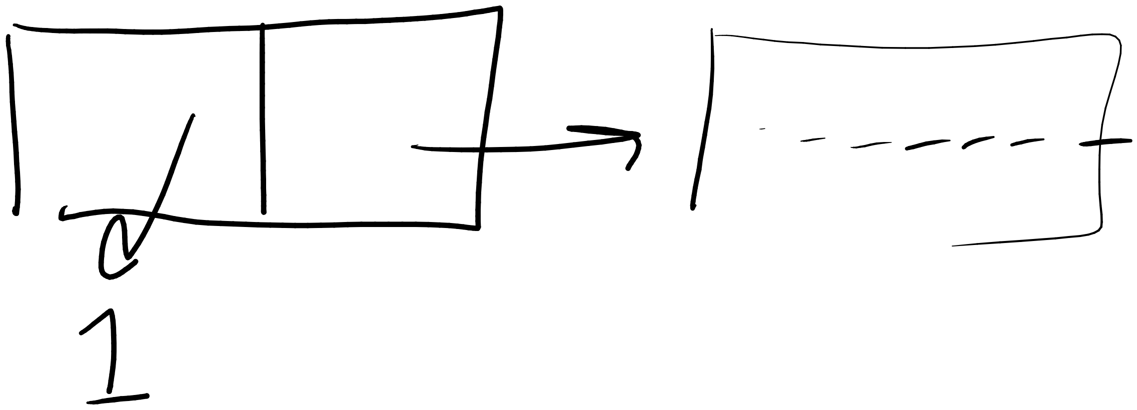




Cons Cell

"construct"

A list is a chain of cons cells
(pairs)

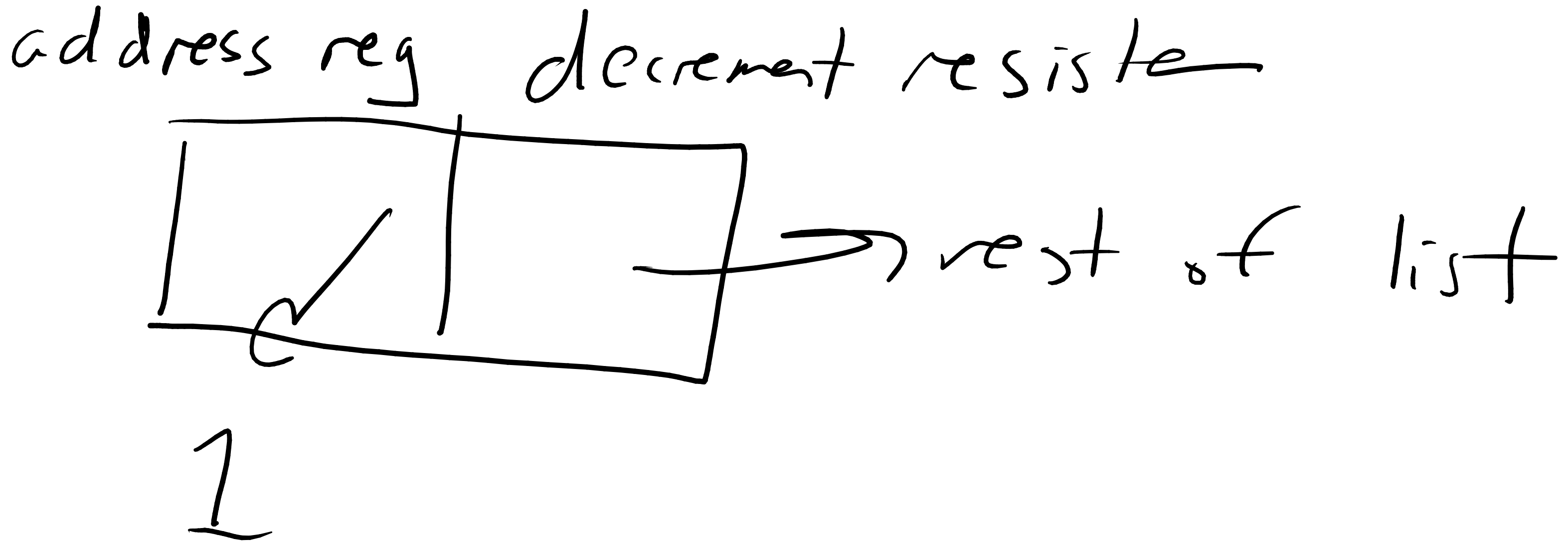


In 1957 (John McCarthy)

He used CPU registers to make things faster

-shoehorned the first half into
"address register"

- - - - second half - - -
"decrement register"



first item of list:

contats of address resister

car

rest of list:

contats of decrement register

cdr

How to create new functions

Built in "function" that is used to make functions

[Python: def]

Scheme: lambda - creates functions

Make a function to add two numbers

```
(lambda (x y)   (+  x  y))
```

```
(lambda (x y)
   (+  x  y))
```

✓

```
(lambda  (x  y)
    ( + x  y)
 )
```

```
(define  add-nums
   (lambda  (x   y)
       (+   x   y)))
```