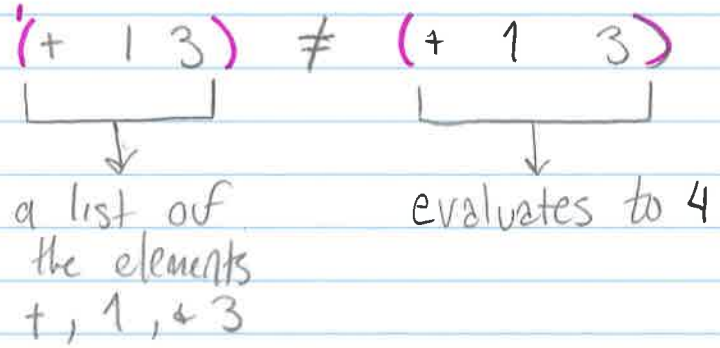
lists

- A SUPER IMPORTANT DATA STRUCTURE
- USEFUL FOR PROOFS & RECURSION
- 3 WAYS TO CONSTRUCT (at least in this class)
  - '()
  - cons
  - list

'()

- NORMALLY IN RACKET WE HAVE THE FOLLOWING SYNTAX
  - (function param0 param1 etc)
  - THIS APPLIES THE function TO THE params
- WHEN WE DON'T WANT TO EVALUATE WHATS IN PARENS, WE USE '() to create A list

EG: '(1 2 3) is a list of elements 1 2 & 3



- '() on its own is an empty list

CONS

- USED TO PUT TWO THINGS TOGETHER INTO A LIST OR PAIR
- (cons a B) makes the list '(a B) WHERE a IS AN element AND B IS A LIST

EXAMPLES:

(cons 1 2) → '(1 . 2) a pair

(cons 1 '(2)) → '(1 2) a list

(cons '(1) '2)) → '('(1) 2) a list

(cons 1 '()) → '(1) a list

LIST

- SIMILAR TO CONS AS ITS A KEYWORD TO MAKE A LIST
- DIFFERENT FROM CONS IN TWO WAYS
  - ↳ TAKES MORE THAN TWO ELEMENTS
  - ↳ DOESN'T DO ANY PAIR THINGS

Ex: (list 1 2 3) → '(1 2 3)