

OPERATIONS

FIRST: RETURNS THE FIRST ELEMENT IN A LIST
↳ `(first '(1 2 3))` → 1

REST: RETURNS A LIST OF ALL THE ELEMENTS IN THE LIST, EXCLUDING THE FIRST
↳ `(rest '(1 2 3))` → '(2 3)

LIST?: RETURNS #t or #f DEPENDING IF SOMETHING IS A LIST OR NOT
↳ `(list? '(1 2 3))` → #t
↳ `(list? ())` → #f

NULL?: RETURNS #t or #f DEPENDING IF THE LIST IS EMPTY OR NOT.
↳ `(null? '())` → #t
↳ `(null? '(1 2))` → #f

MAP: GIVEN A FUNCTION AND LIST, MAP RETURNS A LIST WITH THE FUNCTION APPLIED TO ALL ELEMENTS
↳ `(define (sqr x) (* x x))`
`(map sqr '(1 2 3 4))` → '(1 4 9 16)

FOLD: GIVEN A FUNCTION, BASE, & LIST, FOLD RETURNS A NUMBER OF ALL THE LIST ELEMENTS + THE BASE COMBINED BY THE FUNCTION
↳ `(fold + 2 '(1 2 3))` → 1+2+3+2 → 8