• Part A. Show all the following steps.
  • You do not need to include the "• Steps •", "\", nor "\"" punctuation that DrRacket shows when using step.
  • Include ALL the underlining of sub-expressions that will change.
  • In DrRacket, the step operation starts by copying the expression given to step so that it can add the underlining
    for that initial expression, but you may save some effort by adding the initial underlining directly to the original
    expression inside (step ...) rather than recopying that expression.

(step (map tall (list ○ ○ >)))

(step (map triangle (list 30 10 20)))

(step (map text-join (list "ant" "bear" "ox")))

(step (map function? (list "ant" #true list flip △)))

(step (map text? (list text-length "ant" scale △ 25)))

(step (map list? (list "ant" list #false - 25)))

(step (map binary? (list list? list length)))
(step (map - (map text-length (list "ant" "bear" "ox"))))

(step (map flip (map above (list ♂ ♂ ♂)))

(step (map clockwise (map text->image (list "ant" "bear" "ox"))))

(step (map + (map width (map wide (map solid-circle (list 30 10 20))))))
| (define (a b)          | (define (f c)        |
| (turn □ b))            | (oval c 15))        |
| (define (b c)         | (define (g e)       |
| (scale-width c 2))    | (+ e 10))           |
| (define (c b)         | (define (h f)       |
| (turn b 45))          | (+ f f))            |
| (define (d a)         | (define (i g)       |
| (scale-height Δ a))   | (text-join "=" g)) |
| (define (e h)         | (define (j g)       |
| (above ○ h Δ))        | (text-join g g "|"))  |

... and show all the following steps (with the same guidelines as described in Part A) ...

(step (a 90))

(step (b (c □)))

(step (map d (list 3 1 2)))

(step (map e (list □ ![fill]))
(step (map f (map g (list 0 2 10))))

(step (map square (map h (list 5 3 4))))

(step (map text-length (map i (map j (list "on" "xx"))))))