(define (a b)
    (string-append b "!"))
(a "hello")
; Intermediate step: copy the body of a, replace b with "hello".
(string-append "hello" "!")
; Final answer:
"hello!"

(define (f x y)
    (+ (string-length x) y))
(f "hello" 104)
; Intermediate step: copy the body of f, replace x with "hello",
; y with 104.
(+ (string-length "hello")
  104)
; Next step:
(+ 5 ; Notice: the double-quotes are not characters IN the string.
   104)
; Final answer:
109 ; No quotes: this is not a string.

(require picturing-programs)

(define (c d e)
    (beside e d e))
; This one is very worth drawing arrows from the parameter names
; to where they occur in the body.

(c ♣️ ♥️)
; Intermediate step: copy the body of c, replace d with ♣️ and
; e with ♥️.
(beside ♥️ ♣️ ♥️)
; Final answer:
♥️♣️♥️ ; No quotes: this is not a string.
(define (t i)
    (beside i (rotate 45 i)))
(t (square 10 "outline" "black"))
; Intermediate step: reduce (square 10 "outline" "black") to an
; atomic expression.
(t □)
; Next step: copy the body of t, replace i with □.
(beside □ (rotate 45 □))
; Next step: reduce (rotate 45 □) to an atomic expression.
(beside □ ◊)
; Final answer:

(define (u i)
    (rotate 45 (beside i i)))
(u (square 10 "solid" "black"))
; Intermediate step: reduce (square 10 "solid" "black") to an
; atomic expression.
(u □)
; Next step: copy the body of u, replace i with □.
(rotate 45 (beside □ □))
; Next step: reduce (beside □ □) to an atomic expression.
(rotate 45 □□)
; Final answer: