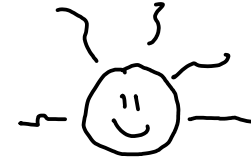


(+ (input "first")  
(input "second"))

add.com



"first"



STATELESS

(+  
(input "second"))

4

Now



Later

(input "first")

(lam (n1)

(+ n1 (input "second"))))

input-then :: String \* (Num -> Nothing) -> Nothing

(input-then "first" (lam (n1) (+ n1 (input "second"))))

prog = (lam (answer-goes-here)

(input-then "first"

(lam (n1)

(input-then "second"

(lam (n2)

(answer-goes-here (+ n1 n2))))))

  = part of every + expression's CPS

(+ • (input "second"))

Nothing

Nothing

Callbacks

(+ n1 •)

Callback Passing Style

(soon to be Continuation passing style)

(prog send-num-response-to-user)

# CPS Transformation

$(\text{lam } (\overset{k}{\text{answer-here}} : (\text{Value} \rightarrow \text{Nothing}))) \dots$

- possible  $k$ :
- print value
  - save value
  - send-user-response

$\text{cps } \llbracket n \rrbracket \quad (\text{lam } (k) \quad (k \ n))$

$\text{cps } \llbracket (+ \ e_1 \ e_2) \rrbracket \quad (\text{lam } (k)$   
 $\quad (\text{cps } \llbracket e_1 \rrbracket$   
 $\quad (\text{lam } (n_1)$

$\text{cps } \llbracket (+ \ 1 \ 2) \rrbracket \quad (\text{cps } \llbracket e_2 \rrbracket$   
 $\quad (\text{lam } (n_2)$   
 $\quad (k \ (+ \ n_1 \ n_2))))))$

$(\text{lam } (k)$   
 $\quad ((\text{lam } (k_2) \ (k_2 \ 1))$   
 $\quad (\text{lam } (n_1)$   
 $\quad \quad ((\text{lam } (k_3) \ (k_3 \ 2))$   
 $\quad \quad (\text{lam } (n_2)$   
 $\quad \quad \quad (k \ (+ \ n_1 \ n_2))))))$













