

```
;; cyclic-env :: id expr env -> env
;; Assumption: expr is a function expression
(define (cyclic-env id fun-expr env)
  (def fun-val-holder (box 'dummy)) ; Box intended to contain the
  ; value to which fun-expr shall reduce in the to-build environment
  ; Assuming it is already in the box, I can build the environment
  (def new-env (box-extend-env id fun-val-holder env))
  ; and use the environment to compute the value of fun-expr
  (def fun-val (eval fun-expr new-env))
  ; Now I set this value as the box content (establishing the circularity)
  (begin
    (set-box! fun-val-holder fun-val)
    ; and return the built environment
    new-env))
```