(λ x. x x) (λ x. x x)

A programmer's blog.

Home About

Fisher-Yates shuffle algorithm in Scheme

SEPTEMBER 24, 2008

```
(define (shuffle a)
  (let* ((len (length a))
     (i len) (ret (list->vector a))
     (r 0) (tmp null))
    (let loop ()
      (if (> i 1)
      (begin
        (set! r (random i))
        (set! i (sub1 i))
        (set! tmp (vector-ref ret i))
        (vector-set! ret i (vector-ref ret r))
        (vector-set! ret r tmp)
        (loop))))
    (vector->list ret)))
```

Here is some test code:

```
(printf "~a~n" (shuffle '(1 2 3 4 5 6)))
(printf "~a~n" (shuffle '(#\H #\E #\L #\L #\O #\, #\W #\O #\R #\L #\D)))

;; Shuffle a whole deck of cards!
(printf "~a~n" (shuffle (list 'clubs-ace 'clubs-2 'clubs-3 'clubs-4

                    'clubs-5 'clubs-6 'clubs-7

                    'clubs-8 'clubs-9 'clubs-10

                    'clubs-jack 'clubs-queen 'clubs-king

                    'diamonds-ace 'diamonds-2 'diamonds-3 'diamonds-4

                    'diamonds-5 'diamonds-6 'diamonds-7

                    'diamonds-8 'diamonds-9 'diamonds-10

                    'diamonds-jack 'diamonds-queen 'diamonds-king

                    'hearts-ace 'hearts-2 'hearts-3 'hearts-4

                    'hearts-5 'hearts-6 'hearts-7

                    'hearts-8 'hearts-9 'hearts-10

                    'hearts-jack 'hearts-queen 'hearts-king

                    'spades-ace 'spades-2 'spades-3 'spades-4

                    'spades-5 'spades-6 'spades-7

                    'spades-8 'spades-9 'spades-10

                    'spades-jack 'spades-queen 'spades-king)))
```

Related

Soundex in Scheme
In "Computer Science"

Bloom filter
In "Computer Science"

Yaaec (Yet another attempt to explain continuations)!
In "Lisp"

from →Computer Science, Lisp, Scheme, Spark

←Soundex in Scheme

Bloom filter →

No comments yet

Leave a Reply

Enter your comment here...