

(λ x. x x) (λ x. x x)

A programmer's blog.

[Home](#) [About](#)

Fisher-Yates shuffle algorithm in Scheme

SEPTEMBER 24, 2008

type and press enter

September 2008

M T W T F S S

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

« Aug Nov »

RECENT POSTS

📖 [Moved to new location](#)

📖 [SQLite: A Lesson In Low-Defect Software](#)

📖 [Fermion – The Scheme Web Server](#)

📖 [John Backus Turing Award lecture](#)

📖 [Spark moves to GitHub](#)

META

📖 [Register](#)

📖 [Log in](#)

📖 [Entries RSS](#)

📖 [Comments RSS](#)

📖 [WordPress.com](#)

```

1 (define (shuffle a)
2   (let* ((len (length a))
3         (i len) (ret (list->vector a))
4         (r 0) (tmp null))
5     (let loop ()
6       (if (> i 1)
7         (begin
8           (set! r (random i))
9           (set! i (sub1 i))
10          (set! tmp (vector-ref ret i))
11          (vector-set! ret i (vector-ref ret r))
12          (vector-set! ret r tmp)
13          (loop)))
14     (vector->list ret)))

```

Here is some test code:

```

1 (printf "~a~n" (shuffle '(1 2 3 4 5 6)))
2 (printf "~a~n" (shuffle '(#\H #\E #\L #\L #\O #\, #\W #\O #\R #\L #\D)))
3
4 ;; Shuffle a whole deck of cards!
5 (printf "~a~n" (shuffle (list 'clubs-ace 'clubs-2 'clubs-3 'clubs-4
6
7   'clubs-5 'clubs-6 'clubs-7
8
9   'clubs-8 'clubs-9 'clubs-10
10
11  'clubs-jack 'clubs-queen 'clubs-king
12
13  'diamonds-ace 'diamonds-2 'diamonds-3 'diamonds-4
14
15  'diamonds-5 'diamonds-6 'diamonds-7
16
17  'diamonds-8 'diamonds-9 'diamonds-10
18
19  'diamonds-jack 'diamonds-queen 'diamonds-king
20
21  'hearts-ace 'hearts-2 'hearts-3 'hearts-4
22
23  'hearts-5 'hearts-6 'hearts-7
24
25  'hearts-8 'hearts-9 'hearts-10
26
27  'hearts-jack 'hearts-queen 'hearts-king
28
29  'spades-ace 'spades-2 'spades-3 'spades-4
30
31  'spades-5 'spades-6 'spades-7
32
33  'spades-8 'spades-9 'spades-10
34
35  'spades-jack 'spades-queen 'spades-king)))

```

About these ads



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...

[The Vigilance Theme.](#)

Blog at W 