

rs.module

```
rs.module
```

regular sequences

rs.iseq

```
rs.iseq :from :to
```

Returns a list of integers.

Parameters

from: lower bound (included)

to: upper bound (included)

Examples

show rs.iseq 3 7

rs.rseq

```
rs.rseq :from :to :count
```

Returns outputs a list of equally spaced rational numbers.

Parameters

from: lower bound (included)

to: upper bound (included)

**count:* number of rational numbers numbers

Examples

show rs.rseq 3 5 5

rs.rand

```
rs.rand :a :b :times
```

Returns a list of random numbers

Parameters

a: lower bound (included)

b: upper bound (included)

times: number random numbers

Examples

show rs.rand 1 100 20

rs.pick

```
rs.pick :list :times
```

Returns a list of random element picked from a list

Parameters

a: lower bound (included)

b: upper bound (included)

times: number random numbers

Examples

show rs.pick ["f "m] 20

rs.rep

```
rs.rep :list [each 1] [times 1]
```

Replicates the values in a list

Parameters

list: list of element to be replicated

each: repeats each list element 'each' times

times: repeats the list 'times' times

Returns

A list.

Examples

```
show (rs.rep (list "a "b) 1 1)
```

```
show (rs.rep (list "a "b) 2 1)
```

```
show (rs.rep (list "a "b) 1 2)
```

```
show (rs.rep (list "a "b) 2 2)
```