NAME

seq - print sequences of numbers

SYNOPSIS

seq [**-w**] [**-f** *format*] [**-s** *string*] [**-t** *string*] [*first* [*incr*]] *last*

DESCRIPTION

The **seq** utility prints a sequence of numbers, one per line (default), from *first* (default 1), to near *last* as possible, in increments of *incr* (default 1). When *first* is larger than *last*, the default *incr* is -1.

All numbers are interpreted as floating point.

Normally integer values are printed as decimal integers.

The **seq** utility accepts the following options:

-f *format*, **--format** *format*
: Use a printf(3) style *format* to print each number. Only the **A**, **a**, **E**, **e**, **F**, **f**, **G**, **g**, and **%** conversion characters are valid, along with any optional flags and an optional numeric minimum field width or precision. The *format* can contain character escape sequences in backslash notation as defined in ANSI X3.159-1989 ("ANSI C89"). The default is **%g**.

-s *string*, **--separator** *string*
: Use *string* to separate numbers. The *string* can contain character escape sequences in backslash notation as defined in ANSI X3.159-1989 ("ANSI C89"). The default is **\n**.

-t *string*, **--terminator** *string*
: Use *string* to terminate sequence of numbers. The *string* can contain character escape sequences in backslash notation as defined in ANSI X3.159-1989 ("ANSI C89"). This option is useful when the default separator does not contain a **\n**.

-w, **--fixed-width**
: Equalize the widths of all numbers by padding with zeros as necessary. This option has no effect with the **-f** option. If any sequence numbers will be printed in exponential notation, the default conversion is changed to **%e**.

EXIT STATUS

The **seq** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Generate a sequence from 1 to 3 (included) with a default increment of 1:

```
# seq 1 3
1
2
3
```

Generate a sequence from 3 to 1 (included) with a default increment of -1:

```
# seq 3 1
3
2
1
```

Generate a sequence from 0 to 0.1 (included) with an increment of 0.05 and padding with leading zeroes.

```
# seq -w 0 .05 .1
0.00
0.05
0.10
```

Generate a sequence from 1 to 3 (included) with a default increment of 1, a custom separator string and a custom terminator:

```
# seq -s "-->" -t "[end of list]\n" 1 3
1-->2-->3-->[end of list]
```

Generate a sequence from 1 to 2 (included) with an increment of 0.2 and print the results with two digits after the decimal point (using a printf(3) style format):

```
# seq -f %.2f 1 0.2 2
1.00
1.20
1.40
1.60
1.80
2.00
```

SEE ALSO

jot(1), printf(1), printf(3)

HISTORY

The **seq** command first appeared in Version 8 AT&T UNIX. A **seq** command appeared in NetBSD 3.0, and was ported to FreeBSD 9.0. This command was based on the command of the same name in Plan 9 from Bell Labs and the GNU core utilities. The GNU **seq** command first appeared in the 1.13 shell utilities release.

BUGS

The **-w** option does not handle the transition from pure floating point to exponent representation very well. The **seq** command is not bug for bug compatible with other implementations.