

# randintlist-13

LaTeX3

Creating random integer number lists,  
with multiple numbers or not,  
sorted or not.

Version 0.1.5 - 27/08/2025

Cédric Pierquet

c pierquet - at - outlook . fr

<https://github.com/cpierquet/latex-packages/tree/main/randintlist>

10 numbers, between 1 and 100, without repetition:

2,29,14,66,51,1,29,32,21,19

The 5th value is:

51

10 numbers, between 1 and 100, without multiples of 5:

23,3,39,23,86,63,82,58,66,46

The 9th value is:

66

15 numbers, between 1 and 20, with repetition:

12,9,2,5,13,14,18,3,13,20,4,8,15,7,20

The last value is:

20

6 sorted numbers, between 1 and 51, without repetition:

ascending: 9,10,25,29,47,49

descending: 50 > 46 > 34 > 31 > 14 > 3

- 
1. The *luarandom* package do the same things, but with the obligation to compile with LuaLaTeX.
  2. The *tuple* package is so much better... take a look...
-

# Contents

<b>1 Loading, useful packages</b>	<b>3</b>
<b>2 The Macros</b>	<b>3</b>
2.1 Global usage . . . . .	3
2.2 Generate the list . . . . .	3
2.3 Accessing elements . . . . .	4
<b>3 History</b>	<b>5</b>
<b>4 The code</b>	<b>5</b>

# 1 Loading, useful packages

In order to load `randintlist-l3`, simply use:

```
\usepackage{randintlist-l3}
```

All code is written in  $\LaTeX$ 3, so no extra packages are needed.

## 2 The Macros

### 2.1 Global usage

Package `randintlist-l3` supports the creation of random integer number lists where a number will appear only once or multiple times. Generated lists can be used with `listofitems`.

💡 Macros are prefixed with `\rdl...` (for `randintlist`).

### 2.2 Generate the list

```
%generate list  
\rdlgenlist[keys]{\macro}
```

Available keys are:

- **min**: minimum value (default 1);
- **max**: maximum value (default 50);
- **nb**: number of values (default 6);
- **sort**: sorting options, within no/asc/dec (default no);
- **repeat**: boolean to authorize repeating values (default false);
- **excluded**: list of excluded values (default empty);
- **seed**: random seed value according to used packages (default -).

```
%default values  
\rdlgenlist{\mylistA}\mylistA  
20,49,9,62,73,14,65,41,22,65
```

```
%10 between 1 and 50, with ascending  
\rdlgenlist[sort=asc,min=1,max=50,nb=10]{\mylistB}\mylistB  
2,3,3,15,25,26,31,33,37,46
```

```
%15 between 1 and 50, with ascending and repetitions allowed  
\rdlgenlist[sort=asc,min=1,max=50,nb=15,repeat]{\mylistC}\mylistC  
7,9,9,10,11,15,15,20,29,33,38,44,46,46,48
```

```
%15 between 1 and 50, without multiples of 5  
\rdlgenlist[%  
  sort=asc,min=1,max=50,nb=15,repeat,%  
  excluded={5,10,15,20,25,30,35,40,45,50}]%  
  {\mylistC}\mylistC  
4,4,12,14,23,27,28,31,37,38,39,44,44,44,49
```

```
%list used with listofitems
\rdlgenlist{\mylistD}\mylistD\par
\readlist*\mylistused{\mylistD}\showitems{\mylistused}\par
\mylistused[1]; \mylistused[-1]
```

```
37,57,94,4,21,24,2,24,71,85


|    |    |    |   |    |    |   |    |    |    |
|----|----|----|---|----|----|---|----|----|----|
| 37 | 57 | 94 | 4 | 21 | 24 | 2 | 24 | 71 | 85 |
|----|----|----|---|----|----|---|----|----|----|


37; 85
```

```
%list, with alt sep, used with listofitems
\rdlgenlist<+>{\mylistZ}\mylistZ\par
\setsepchar{+}\readlist*\mylistused{\mylistZ}\showitems{\mylistused}\par
\mylistused[1]; \mylistused[-1]
```

```
66+2+68+61+70+29+19+97+67+78


|    |   |    |    |    |    |    |    |    |    |
|----|---|----|----|----|----|----|----|----|----|
| 66 | 2 | 68 | 61 | 70 | 29 | 19 | 97 | 67 | 78 |
|----|---|----|----|----|----|----|----|----|----|


66; 78
```

## 2.3 Accessing elements

```
%accessing item
\rdlgetitem(*){\macro}{index}[\macrores]
```

```
%with default keys
\rdlgenlist{\mylistE}raw list: \mylistE\par
items list:\par
\xintFor* #1 in {\xintSeq{1}{6}}\do{\rdlgetitem*\mylistE}{#1}\par
first element: \rdlgetitem*\mylistE}{1}\par
```

```
raw list: 98,94,32,81,84,81,99,33,27,66
items list:
98
94
32
81
84
81
first element: 98
```

```
\rdlgetitem{\mylistE}{3}[\myres]%
third element: \myres
third element: 32
```

### 3 History

0.1.5: Initial version

### 4 The code

```
% Author      : C. Pierquet
% licence     : Released under the LaTeX Project Public License v1.3c or later, see
               http://www.latex-project.org/lppl.txt

\NeedsTeXFormat{LaTeX2e}
\ProvidesExplPackage{randintlist-13}{2025-08-27}{0.1.5}{Create a list of random numbers with or without multiple
  values}

%-----History
% 0.1.5 Initial version

%-----Main macro
%variables
\clist_new:N \l_randintlist_clist
\clist_new:N \l_randintlist_excluded_clist
\clist_new:N \l_randintlist_result_clist
\int_new:N \l_randintlist_min_int
\int_new:N \l_randintlist_max_int
\int_new:N \l_randintlist_nb_int
\int_new:N \l_randintlist_random_number
\int_new:N \l_randintlist_seed_int
\bool_new:N \l_randintlist_repeat_bool
\bool_new:N \l_randintlist_keepvalue_bool
\tl_new:N \l_randintlist_sep_tl
\tl_new:N \l_randintlist_sort_tl
\str_new:N \l_randintlist_sort_str

%keys
\keys_define:nn { randomlistintegers }
{
  min      .int_set:N = \l_randintlist_min_int,
  max      .int_set:N = \l_randintlist_max_int,
  nb       .int_set:N = \l_randintlist_nb_int,
  seed     .int_set:N = \l_randintlist_seed_int,
  excluded .clist_set:N = \l_randintlist_excluded_clist,
  repeat   .bool_set:N = \l_randintlist_repeat_bool,
  %sep     .tl_set:N   = \l_randintlist_sep_tl,
  sort     .tl_set:N   = \l_randintlist_sort_tl,
  min      .initial:n = 1,
  max      .initial:n = 100,
  nb       .initial:n = 10,
  seed     .initial:n = -1,
  excluded .initial:n = {},
  repeat   .initial:n = true,
  %sep     .initial:n = {,},
  sort     .initial:n = {none},
  min      .default:n = 1,
  max      .default:n = 100,
  nb       .default:n = 10,
  seed     .default:n = -1,
  excluded .default:n = {},
  repeat   .default:n = true,
  %sep     .default:n = {,},
  sort     .default:n = {none},
}

%macro
\NewDocumentCommand\rdlgenlist { 0{ } D<>{,} m } % #1=keys / #2=sep / #3=macrolist
{
  %\group_begin:
  % key init
  \keys_set:nn { randomlistintegers } { #1 }
}
```

```

% seed if necessary
\int_compare:nNnT { \l_randintlist_seed_int } > { -1 }
{
  \sys_rand_seed:n { \l_randintlist_seed_int }
}

% list init
\clist_clear:N \l_randintlist_result_clist

% repeating or not
\bool_if:NTF \l_randintlist_repeat_bool
{% w repeating
  \int_step_inline:nnnn { 1 } { 1 } { \l_randintlist_nb_int }
  {
    \bool_set_false:N \l_randintlist_keepvalue_bool

    \bool_until_do:Nn \l_randintlist_keepvalue_bool
    {
      \int_set:Nn \l_randintlist_random_number
      {
        \fp_eval:n { randint( \l_randintlist_min_int , \l_randintlist_max_int ) }
      }
      \tl_set:Nx \l_tmpa_tl { \int_use:N \l_randintlist_random_number }

      \clist_if_in:NeTF \l_randintlist_excluded_clist { \l_tmpa_tl }
      {
        \bool_set_false:N \l_randintlist_keepvalue_bool
      }
      {
        \bool_set_true:N \l_randintlist_keepvalue_bool
        \clist_put_right:Nx \l_randintlist_result_clist { \int_use:N \l_randintlist_random_number }
      }
    }
  }
}
{%w/o repetitions
  \seq_clear_new:N \l_tmpa_seq

  \int_step_inline:nnnn { \l_randintlist_min_int } { 1 } { \l_randintlist_max_int }
  {
    %creation of [min,...,max] list

    \clist_if_in:NnTF \l_randintlist_excluded_clist { ##1 }
    {
      \bool_set_false:N \l_randintlist_keepvalue_bool
    }
    {
      \bool_set_true:N \l_randintlist_keepvalue_bool
      \seq_put_right:Nn \l_tmpa_seq { ##1 }
    }
  }

  %shuffle
  \seq_shuffle:N \l_tmpa_seq

  %truncate
  \clist_clear_new:N \l_randintlist_result_clist
  \int_step_inline:nnnn { 1 } { 1 } { \l_randintlist_nb_int }
  {
    \seq_pop_left:NN \l_tmpa_seq \l_tmpa_tl
    \clist_put_right:Nx \l_randintlist_result_clist { \l_tmpa_tl }
  }
}

%sort or not
\str_set:Nx \l_randintlist_sort_str { \tl_use:N \l_randintlist_sort_tl }
\str_case:Nn { \l_randintlist_sort_str }
{
  {asc} { \clist_sort:Nn \l_randintlist_result_clist { \int_compare:nNnTF { ##1 } > { ##2 } {
\sort_return_swapped: } { \sort_return_same: } } }
  {des} { \clist_sort:Nn \l_randintlist_result_clist { \int_compare:nNnTF { ##1 } < { ##2 } {
\sort_return_swapped: } { \sort_return_same: } } }
}

```

```

}

%storing
\tl_gset:Nx #3 {
  \clist_use:Nn \l_randintlist_result_clist { #2 }
}
%\group_end:
}

\NewDocumentCommand\rdlgetitem{ s m m O{\resmyelt} }{%
  \clist_set:Nx \l_randintlist_clist {#2}
  \IfBooleanTF{#1}
  {
    \clist_item:Nn \l_randintlist_clist { #3 }
  }
  {
    \tl_gset:Nx #4{ \clist_item:Nn \l_randintlist_clist { #3 } }
  }
}

\endinput

```