

randintlist

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:

72,85,2,64,20,16,69,68,63,95

The 5th value is:

20

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

74,34,56,52,48,59,53,63,1,43

The 9th value is:

1

15 numbers, between 1 and 20, with repetition:

17,1,9,1,10,13,12,17,7,14,7,8,20,15,10

The last value is:

10

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

ascending : 19,24,35,42,49,50

descending : 47>33>29>23>21>20

-
1. The *luarandom* package do the same things, but with the obligation to compile with *Lua^AT_EX*.
 2. The *tuple* package is so much better... but I keep *randintlist*, without new features...
-

Contents

1 Loading, useful packages	3
2 The Macros	3
2.1 Global usage	3
2.2 Generate the list	3
2.3 Accessing elements	4
2.4 Version française	5
3 Example	6
4 History	7
5 The code	7

1 Loading, useful packages

In order to load `randintlist`, simply use:

```
\usepackage{randintlist}
```

Loaded packages are ifthen, simplekv, listofitems, randomlist, xintexpr and xstring.

2 The Macros

2.1 Global usage

Package `randintlist` 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`.

All engines T_EX are compatible with this package.

2.2 Generate the list

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

Available keys are:

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

```
%default values  
\randintlist{\mylistA}\mylistA  
31,38,13,20,49,36
```

```
%10 between 1 and 50, with ascending  
\randintlist[sort=asc,min=1,max=50,nb=10]{\mylistB}\mylistB  
4,5,13,14,24,29,30,31,35,37
```

```
%15 between 1 and 50, with ascending and repetitions allowed  
\randintlist[sort=asc,min=1,max=50,nb=15,repeat]{\mylistC}\mylistC  
7,7,9,12,21,24,25,25,25,27,35,41,44,48,50
```

```
%15 between 1 and 50, without multiples of 5
\randintlist[%
  sort=asc,min=1,max=50,nb=15,repeat,%
  exclude={5,10,15,20,25,30,35,40,45,50}]%
{\mylistC}\mylistC
```

```
1,1,2,3,4,4,6,9,11,11,18,23,26,47,49
```

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

```
49,42,48,41,13,18
49 42 48 41 13 18
49; 18
```

2.3 Accessing elements

```
%accessing item
\getitemfromrandintlist[separator]{\macro}{index}[\macrores]
```

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

```
raw list: 36,13,4,31,35,8
items list:
36
13
4
31
35
8
first element: 36
```

```
\getitemfromrandintlist{\mylistE}{3}[\myres]%
third element: \myres
third element: 4
```

2.4 Version française

Voilà les commandes en version française, la syntaxe et les clés ne seront pas explicitées.

```
%obtenir la liste  
\ListeRandint[Min=.,Max=.,Nb=.,Repet=.,Graine=.,Tri=.,Sep=.,Exclure=..]{\macro}
```

```
%extraire un élément  
\ExtraireEltListeRandint[sep]{\macro}{position}[\macrores]
```

```
%liste  
\ListeRandint[Min=5,Max=15,Nb=7,Repet,Tri=croiss,Sep={/}]{\maliste}\maliste\  
%élément  
\ExtraireEltListeRandint[/]{\maliste}{4}
```

```
10/12/12/13/13/13/13  
13
```

```
%liste  
\ListeRandint[Min=50,Max=100,Nb=10,Repet,Tri=croiss]{\malisteB}\malisteB\  
%troisième élément  
\ExtraireEltListeRandint{\malisteB}{3}[\montroisieme]  
troisième élément : \montroisieme
```

```
50,50,65,69,72,75,75,83,90,93  
troisième élément : 65
```

3 Example

The following example uses TikZ, and comes from luarandom's documentation.

```
\begin{tikzpicture}[scale=0.75]
  \randintlist[min=1,max=100,nb=100]{\mylistsquare}
  \draw[thin,gray] (0,0) grid (10,10) ;
  \foreach \i in {1,...,100}{%
    \xdef\tmpnumber{\getitemfromrandintlist{\mylistsquare}{\i}}%
    \xdef\tmpnumberrow{\xinteval{\xintiiRem{\i-1}{10}}}%
    \xdef\tmpnumbercol{\xinteval{\xintiiQuo{\i-1}{10}}}%
    \draw ({0.5+\tmpnumbercol},{0.5+\tmpnumberrow}) node {\tmpnumber} ;
  }%
\end{tikzpicture}
```

30	65	85	68	18	32	49	62	44	22
7	21	77	51	66	42	92	46	100	70
90	9	25	6	57	84	74	88	82	34
96	75	31	33	29	55	71	91	67	14
5	81	61	36	86	69	15	11	28	27
41	87	2	50	37	35	73	76	43	56
17	98	52	63	54	19	59	83	39	38
10	3	4	23	26	40	78	89	99	24
47	53	97	60	72	48	95	8	45	58
16	13	12	64	79	93	94	20	1	80

4 History

0.1.5: L^AT_EX3 version of the package
0.1.4: Bugfix
0.1.3: Bugfix
0.1.2: Changing name of internal macro
0.1.1: Possibility to exclude values
0.1.0: Initial version

5 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}
\ProvidesPackage{randintlist}[2025/08/27 0.1.5 Create a list of random numbers with or without multiple values]

%-----History
% 0.1.5 l3 version of the package
% 0.1.4 Bugfix
% 0.1.3 new usage of extracting element (with storing \macro)
% 0.1.2 Changing name of macro
% 0.1.1 Possibility to exclude values
% 0.1.0 Initial version

%-----Packages
\RequirePackage{simplekv}
\RequirePackage{listofitems}
\RequirePackage{randomlist}
\RequirePackage{xintexpr}
\RequirePackage{xstring}
\RequirePackage{ifthen}

%-----Macros (latex3) for sorting and seed
\ExplSyntaxOn
\cs_new_eq:NN \randintseed \sys_gset_rand_seed:n
\NewDocumentCommand\intascsortlist{m}
{
  \clist_sort:Nn #1
  {
    \fp_compare:nNnTF {##1} > {##2}
    { \sort_return_swapped: }
    { \sort_return_same: }
  }
}
\NewDocumentCommand\intdessortlist{m}
{
  \clist_sort:Nn #1
  {
    \fp_compare:nNnTF {##1} < {##2}
    { \sort_return_swapped: }
    { \sort_return_same: }
  }
}
\ExplSyntaxOff

%----Internal macro (latex2) for testing if element is in list
\newcommand\ifintvalueinlist[2]{\IfSubStr{,#2,}{,#1,}}

\newcommand\boolvalueinlist[2]{\IfSubStr{,#2,}{,#1,}}{\def\resisinlist{1}}{\def\resisinlist{0}}

\newcommand\testifintvalueinlist[4]{%
  \IfSubStr{,#2,}{,#1,}}{\xdef\RESTMPVALUE{1}}{\xdef\RESTMPVALUE{0}}%
  \xintifboolexpr{ \RESTMPVALUE == 1}{#3}{#4}%
}
```

```

%---Macro for generating
\defKV[randomlistintegers]{%
  min=\def\TAEEmin{#1},%
  max=\def\TAEEmax{#1},%
  nb=\def\TAEEnb{#1},%
  sep=\def\TAEEsep{#1},%
  sort=\def\TAEetri{#1},%
  seed=\def\TAESeed{#1},%
  exclude=\def\TAEExcluded{#1}
}

\setKVdefault[randomlistintegers]{%
  min=1,%
  max=50,%
  nb=6,%
  sep={,},%
  sort=no,%
  repeat=false,%
  seed={-},%
  exclude={}
}

\NewList{tmprandintlist}

\NewDocumentCommand\randintlist{ O{ } m }{%1=keys,2=listname
  \useKVdefault[randomlistintegers]%
  \setKV[randomlistintegers]{#1}%
  \ifboolKV[randomlistintegers]{repeat}%repeat or not
    {%repeat allowed
      \IfStrEq{\TAESeed}{-}%
        {}%
        {%
          \randintseed{\TAESeed}%
        }%
      %list creation of first element
      \def\resisinlist{1}%
      \whiledo{\resisinlist=1}{%
        \xdef\tmpresrandint{\fpeval{randint(\TAEEmin,\TAEEmax)}}%
        \boolvalueinlist{\tmpresrandint}{\TAEExcluded}%
      }%
      \xdef#2{\tmpresrandint}%
      %list creation of other elements
      \xintFor* ##1 in {\xintSeq{2}{\TAEEnb}}%
        \do{%
          \def\resisinlist{1}%
          \whiledo{\resisinlist=1}{%
            \xdef\tmpresrandint{\fpeval{randint(\TAEEmin,\TAEEmax)}}%
            \boolvalueinlist{\tmpresrandint}{\TAEExcluded}%
          }%
          \xdef#2{#2,\tmpresrandint}%
        }%
    }%
    {%no repeating
      %randomize numbers
      \IfStrEq{\TAESeed}{-}%
        {}%
        {%
          \RLsetrandomseed{\TAESeed}%
        }%
      \ClearList{tmprandintlist}%clearing the list
      \xintFor* ##1 in {\xintSeq{\TAEEmin}{\TAEEmax}}%
        \do{%
          \ifintvalueinlist{##1}{\TAEExcluded}%
            {}%
            {%
              \InsertRandomItem{tmprandintlist}{##1}%
            }%
        }%
      %list creation (first then other)
      \xdef#2{\tmprandintlist[0]}%
      \xintFor* ##1 in {\xintSeq{1}{\TAEEnb-1}}%
        \do{%

```



```

\edef#2{#2,\tmprandintlist[#1]}%
}%
}%
%sorting
\IfStrEq{\TAEetri}{asc}%if ascending
{\intascsortlist{#2}}%
}%
\IfStrEq{\TAEetri}{des}%if descending
{\intdessortlist{#2}}%
}%
\StrSubstitute{#2}{,}{\TAEesep}[#2]%swipping separator if necessary
}

%-----Macro for extracting
\NewDocumentCommand\getitemfromrandintlist{ O{,} m m o }{%
\IfStrEq{#1}{/}%
{%useful
\setsepchar[.]{#1}%
}%
{%
\setsepchar{#1}%
}%
\readlist*\TMPLISTRANDINT{#2}%
\IfNoValueTF{#4}{\TMPLISTRANDINT[#3]}{\itemtomacro\TMPLISTRANDINT[#3]#4}%
}

%-----french version
\defKV[randomlisteentiers]{%
Min=\def\TAEemin{#1},%
Max=\def\TAEemax{#1},%
Nb=\def\TAEenb{#1},%
Sep=\def\TAEesep{#1},%
Tri=\def\TAEetri{#1},%
Graine=\def\TAEeseed{#1},%
Exclure=\def\TAEexclused{#1}
}

\setKVdefault[randomlisteentiers]{%
Min=1,%
Max=50,%
Nb=6,%
Sep={,},%
Tri=non,%
Repet=false,%
Graine={-},%
Exclure={}
}

\NewDocumentCommand\ListeRandint{ O{ } m }{%1=keys,2=listname
\useKVdefault[randomlisteentiers]%
\setKV[randomlisteentiers]{#1}%
\ifboolKV[randomlisteentiers]{Repet}%repeat or not
{%repeat allowed
\IfStrEq{\TAEeseed}{-}%
{%
}%
\randintseed{\TAEeseed}%
}%
%list creation of first element
\def\resisinlist{1}%
\whiledo{\resisinlist=1}{%
\edef\tmpresrandint{\fpeval{randint(\TAEemin,\TAEemax)}}%
\boolvalueinlist{\tmpresrandint}{\TAEexclused}%
}%
\edef#2{\tmpresrandint}%
%list creation of other elements
\xintFor* ##1 in {\xintSeq{2}{\TAEenb}}%
\do{%
\def\resisinlist{1}%
\whiledo{\resisinlist=1}{%
\edef\tmpresrandint{\fpeval{randint(\TAEemin,\TAEemax)}}%
\boolvalueinlist{\tmpresrandint}{\TAEexclused}%
}
}
}

```

```

    }%
    \xdef#2{#2,\tmpresrandint}%
  }%
}%
{%no repeating
  %randomize numbers
  \IfStrEq{\TAESeed}{-}%
  {}%
  {%
    \RLsetrandomseed{\TAESeed}%
  }%
  \ClearList{tmprandintlist}%clearing the list
  \xintFor* ##1 in {\xintSeq{\TAEEmin}{\TAEEmax}}%
  \do{%
    \ifintvalueinlist{##1}{\TAEExcluded}%
    {}%
    {%
      \InsertRandomItem{tmprandintlist}{##1}%
    }%
  }%
  %list creation (first then other)
  \xdef#2{\tmprandintlist[0]}%
  \xintFor* ##1 in {\xintSeq{1}{\TAEEnb-1}}%
  \do{%
    \xdef#2{#2,\tmprandintlist[##1]}%
  }%
}%
%sorting
\IfStrEq{\TAEETri}{croiss}%if ascending
  {\intascsortlist{#2}}%
  {}%
\IfStrEq{\TAEETri}{decroiss}%if descending
  {\intdessortlist{#2}}%
  {}%
\StrSubstitute{#2}{,}{\TAESep}[#2]%swipping separator if necessary
}

%-----Macro for extracting
\NewDocumentCommand\ExtraireEltListeRandint{ O{,} m m o }{%
  \IfStrEq{#1}{/}%
  {%useful
    \setsepchar[.]{#1}%
  }%
  {%
    \setsepchar{#1}%
  }%
}%
\readlist*\TMPLISTRANDINT{#2}%

```