

Operators DEF_LIST_REEL

1 Goal

To create a strictly increasing list of realities.

The list can be given “in extenso” by the user, or, it can be formed from under lists defined in “constant step”.

Product a structure of data of the type `listr8`.

2 Syntax

```
Lr      [listr8] = DEFI_LIST_REEL
      (
        /      ♦  VALE=      lr8      ,      [l_R]
        /      ♦  DEBUT=     debu     ,      [R]
        ♦  INTERVALLE=  (_F ( ♦  JUSQU_A =      r1, [R]
                               ♦  /  NUMBER =     n1, [I]
                               /  NOT =      r2, [R]
                               ), ),
        ♦  INFORMATION =      /  1      ,
        [DEFECT]
                               /  2      ,
        ♦  TITLE =      title     ,      [l_Kn]
      )
```

3 Operands

3.1 Operand VALE

VALE = lr8

List of realities which will form the structure of data `listr8` result.
This list can be built starting from a list Python.

3.2 Operand BEGINNING

♦ BEGINNING =

It is the first reality of the list of realities which one wants to build.

3.3 Operand INTERVAL

♦ INTERVAL =

♦ JUSQU_A = r1

It is the end of the interval which one will cut out with a constant step.

♦ / NUMBER = n1

It is the number of steps which one wants in the interval which ends in `r1`.

/ NOT = r2

It is the step of division interval.

3.4 Operand INFORMATION

♦ INFORMATION = I

Indicate the level of impression of the results of the operator.

- 1: no impression,
- 2: impression of the list of realities created

3.5 Operand TITLE

◇ TITLE = title

Title which the user wants to give to his list of realities.

4 Remarks

- when the keyword is used `NOT` it may be that the number of calculated step is not rigorously whole. One “will then adapt” the last interval to fall down exactly on the end value (`JUSQU_A`). So for that, one modifies the step value of more than 1/1000 one emits an alarm,
- caution: this order produces a structure of data `listr8` who can be used only in the orders expecting such structures of data and not in those which expect lists of realities (notation: `l_R`).

5 Examples

Example 1:

Let us imagine that one wants to create the list:

1. 3. 5. 10. 15. 20. 25. 26. 27. 28.

who is such as the step is:

2.	of	1.	with	5.
5.	of	5.	with	25.
1.	of	25.	with	28.

One can write:

```
Lr = DEFI_LIST_REEL (BEGINNING = 1. ,  
INTERVAL = (_F (JUSQU_A= 5. , NOMBRE= 2, ),  
_F (JUSQU_A= 25. , NOMBRE= 4, ),  
_F (JUSQU_A= 28. , PAS=  
1. ,),),  
)
```

Example 2:

To create the list: 1. 3. 12. 13.

One can write:

```
Lr = DEFI_LIST_REEL ( VALE = (1. , 3. , 12. , 13.), )
```

Example 3:

One can build a list Python in this manner.

```
Lr = DEFI_LIST_REEL ( VALE = [sqrt (I) for I in arranges (5)], )
```