

Number lists

Overview

The **Number list data type** is an ordered list of numbers. This data type is returned, among others, by functions that return values of number fields in a selection of issues (**linked issues**, **sub-tasks**, and **subsets**).

Fixed values

A **number list** can also be written in literal form using the following format: *[number, number, ...]*.



Example

A number list with 5 elements: [1, -2, 3, 3.14, 2.71]

Number list functions

The following functions are intended to build expressions that return **number lists** or **numbers**.

Function	Input	Returned value
filterByCardinality (number list l , comparison operator operator , number n)	<div>NUMBER []</div> <div>NUMBER</div>	Returns a <div>NUMBER []</div> l whose cardinality (i.e., the number of times it appears in list l) satisfies the comparison cardinality operator n . Available comparison operators: =, !=, <, <=, > and >=. Example: <code>filterByCardinality([1, 1, 2, 3, 4, 4, 4, 5], >, 1)</code> returns the following number list: [1, 4] .
filterByValue (number list l , comparison operator operator , number n)	<div>NUMBER []</div> <div>NUMBER</div>	Returns a <div>NUMBER []</div> l satisfying the comparison number_in_list operator n . Example: <code>filterByValue([1, 2, 3, 10, 11, 25, 100], >, 10)</code> returns the list of numbers greater than 10 . i.e., [11, 25, 100] .
filterByPredicate (number list l , boolean expression predicate)	<div>NUMBER []</div> <div>BOOLEAN</div>	Returns a <div>NUMBER []</div> l that validates a predicate. Argument predicate is a boolean expression, where ^ is used for referencing numeric values in argument l . Example: <code>filterByPredicate([1, 2, 3, 4], ^ > 2)</code> returns values greater than 2 , i.e., [3, 4] . Example: <code>filterByPredicate([1, 2, 3, 4], remainder(^, 2) = 0)</code> returns even values, i.e., [2, 4] .
append (number list l , number list m)	<div>NUMBER []</div>	Returns a <div>NUMBER []</div> with all numbers in arguments l and m . Duplicated numbers may appear in output. Use function union(l, m) instead, if you want to avoid repetitions. Example: <code>append([1, 2, 3], [3, 4, 5])</code> returns [1, 2, 3, 3, 4, 5] . Example: <code>append(fieldValue({00025}, linkedIssues("is blocked by")), fieldValue({00025}, subtasks()))</code> returns a list of numbers with Total Time Spent (in minutes) in blocking issues and sub-tasks. This number list can be summed using function sum() .
union (number list l , number list m)	<div>NUMBER []</div>	Returns a <div>NUMBER []</div> with all numbers in argument l or in argument m without duplicated numbers. Example: <code>union([1, 2, 3], [3, 4, 5])</code> returns [1, 2, 3, 4, 5] .

On this page

- [Overview](#)
- [Fixed values](#)
- [Number list functions](#)

except (number list l , number list m)	NUMBER []	Returns a NUMBER [] with all numbers in argument l which are not in argument m . Duplicated numbers in l may appear in output. Use function distinct() to remove them if you need to. Example: except ([1, 2, 3, 4, 5], [2, 4]) returns [1, 3, 5].
intersect (number list l , number list m)	NUMBER []	Returns a NUMBER [] with all numbers in argument l and m simultaneously. Example: intersect ([1, 2, 3, 4, 5], [9, 7, 5, 3, 1]) returns [1, 3, 5].
distinct (number list l)	NUMBER []	Returns a NUMBER [] with all numbers in list l with out any duplication. Example: distinct ([1, 2, 1, 3, 4, 4, 5]) returns [1, 2, 3, 4, 5]. Example: distinct (fieldValue ({... duedate }, linkedIssues ("is cloned by")) returns a list of dates containing due dates of cloning issues, with only one occurrence per due date, although more than one issue may share the same due date.
count (number list l)	NUMBER []	Returns the NUMBER of numeric values in l . Example: count ([1, 1, 2, 2]) returns 4. Example: count (subtasks ()) - count (fieldValue ({... duedate }, subtasks ())) returns the number of sub-tasks with field "Due Date" unset.
count (number n , number list l)	NUMBER NUMBER []	Returns the NUMBER of times n appears in l . Example: count (1, [1, 1, 2, 2, 1, 0]) returns 3.
sum (number list l)	NUMBER []	Returns the sum of NUMBER values in l . Example: sum ([1, 2, 3, 4, 5]) returns 15. Example: sum (fieldValue ({00025}, subtasks ())) returns the total time spent in minutes in all sub-tasks of current issue.
avg (number list l)	NUMBER []	Returns the arithmetic mean of NUMBER values in l . Example: avg ([1, 2, 3, 4, 5]) returns 3. Example: avg (fieldValue ({00024}, linkedIssues ("is blocked by"))) returns the mean of remaining times in minutes among blocking issues.
max (number list l)	NUMBER []	Returns the maximum NUMBER value in l . Example: max ([1, 2, 5, 4, 3]) returns 5. Example: max (fieldValue ({00024}, linkedIssues ("is blocked by"))) returns the maximum remaining times in minutes among blocking issues.
min (number list l)	NUMBER []	Returns the minimum NUMBER value in l . Example: min ([2, 1, 5, 4, 3]) returns 1. Example: min (fieldValue ({00024}, linkedIssues ("is blocked by"))) returns the minimum remaining times in minutes among blocking issues.
first (number list l)	NUMBER []	Returns NUMBER of the first element in number list l , or null if l is an empty list. Example: first ([3, 2, 1, 0]) returns 3.
last (number list l)	NUMBER []	Returns NUMBER of the first element in number list l , or null if l is an empty list. Example: last ([3, 2, 1, 0]) returns 0.
nthElement (number list l , number n)	NUMBER NUMBER []	Returns NUMBER element at position n in number list l , where n >= 1 and n <= count(l) . Returns null if n is greater than the number of elements in l . Example: nthElement ([5, 6, 7, 8], 3) returns 7.

getMatchingValue (string key , string list key_list , number list value_list)	<div>STRING</div> <div>NUMBER []</div> <div>STRING []</div>	Returns <div>NUMBER</div> in value_list that is in the same position as string key is in key_list , or in case key doesn't exist in key_list and value_list has more elements than key_list , the element of value_list in position count(key_list) + 1 . Example: <code>getMatchingValue("Three", ["One", "Two", "Three", "Four", "Five"], [1, 1+1, 3*1, 4, 4+1])</code> returns 3.
getMatchingValue (string key , string list key_list , number list value_list)	<div>NUMBER []</div> <div>STRING []</div>	Returns <div>NUMBER</div> value in value_list that is in the same position as numeric key is in key_list , or in case key doesn't exist in key_list and value_list has more elements than key_list , the element of value_list in position count(key_list) + 1 . Example: <code>getMatchingValue(5, [1, 3, 5, 7, 9], [1, 1+1, 3*1, 4, 4+1])</code> returns 3.
sublist (number list l , number indexFrom , number indexTo)	<div>NUMBER []</div>	Returns a <div>NUMBER []</div> with elements in l from indexFrom index to indexTo index. Having indexFrom >= 1 and indexFrom <= count(l) and indexTo >= 1 and indexTo <= count(l) and indexFrom <= indexTo . Example: <code>sublist([1, 2, 3, 4, 5], 2, 4)</code> returns [2, 3, 4].
indexOf (number element, number list l)	<div>NUMBER</div> <div>NUMBER []</div>	Returns the index of <div>NUMBER</div> value element in number list l . Zero is returned when element is not found in l . Example: <code>indexOf(1, [5, 2, 1, 4, 1])</code> returns 3.
sort (number list l , order)	<div>NUMBER []</div>	Returns a <div>NUMBER []</div> with elements in l sorted in specified order. Available orders are ASC (for ascending order) and DESC (for descending order). Example: <code>sort([2, 4, 3, 1], ASC)</code> returns [1, 2, 3, 4].
textOnNumberList (number list numbers , string text_expression)	<div>NUMBER []</div> <div>STRING</div>	Returns a <div>STRING []</div> resulting of evaluating text_expression against each of the numeric values in argument numbers . Argument text_expression is an expression that returns a string , where ^ represents each numeric value in argument numbers . Example: <code>textOnNumberList([1, 2, 3, 4, 5], substring("smile", 0, ^))</code> returns string list ["s", "sm", "smi", "smil", "smile"].
mathOnNumberList (number list numbers , number math_time_expression)	<div>NUMBER []</div>	Returns a <div>NUMBER []</div> resulting of evaluating math_time_expression against each of the numeric values in argument numbers . Argument math_time_expression is a math/time expression, where ^ represents each numeric value in argument numbers . Example: <code>mathOnNumberList([1, 2, 3, 4, 5], ^ * 2)</code> returns number list [2, 4, 6, 8, 10].