sort()

Supported list types

Number list | Text list | Issue list | Use cases and examples

Number list

This function sorts a given **number list** in a specified order. Available orders are **ASC** (for ascending order) and **DESC** (for descending order).

Syntax
`sort(numberList, order) #Output: Number list`

Examples

Parser expression	Description
`sort([2, 4, 3, 1], ASC)`	This example returns: **[1, 2, 3, 4]**
`sort(fieldValue({issue.cf10110}, linkedIssues()), DESC)`	This example returns a **number list** with e.g. the **Story Points** of all linked issues in **descending order**. To achieve this, the following functions are used: • fieldValue() • linkedIssues() Note that **`{issue.cf10110}`** is the field code for **Story Points**. It might differ on your instance.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`numberList`	NUMBER LIST	Any given number list.
`order`	TEXT	Available orders are **ASC** (for ascending order) and **DESC** (for descending order).

Output

The function returns a NUMBER LIST

Text list

Variant for **text lists**.

Syntax
`sort(textList, order) #Output: Text list`

Examples

Parser expression	Description
`sort(["red", "blue", "green"], ASC)`	This example returns: **["blue", "green", "red"]**
`sort(fieldValue(%{issue.assignee}, subtasks()), ASC)`	This example returns a text list with all **sub-tasks's assignees** in ascending order. To achieve this, the following functions are used: • fieldValue() • subtasks()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`textList`	TEXT LIST	Any given text list.
`order`	TEXT	Available orders are **ASC** (for ascending order) and **DESC** (for descending order).

Output

The function returns a TEXT LIST

Issue list

Variant for **issue lists**.

In this case, a **field** has to be provided that should be used for the sort order. Available orders are **ASC** (for ascending order) and **DESC** (for descending order).

Syntax
`sort(issueList, field, order) #Output: Issue list`

Examples

Parser expression	Description
`sort(linkedIssues("is blocked by"), {issue.dueDate}, ASC)`	This example returns an **issue list** of issues **blocking** current issue, sorted in **ascending** order by **Due date**. To achieve this, the following functions are used: • linkedIssues()
`sort(subtasks(), %{issue.assignee}, ASC)`	This example returns an **issue list** of **sub-tasks,** sorted in **ascending** order by their **assignee**. To achieve this, the following functions are used: • subtasks()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
`issueList`	ISSUE LIST	Any given issue list. Usually this value is retrieved from a function (e.g. linkedIssues() or subtasks()).
`field`	TEXT	Any field code representing a number, text or selectable field.
`order`	TEXT	Available orders are **ASC** (for ascending order) and **DESC** (for descending order).

Output

The function returns an ISSUE LIST

Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Highest ranked custom field value among all linked issues			Text		first() sort() fieldValue() linkedIssues()