

count()

Supported list types

[Number list](#) | [Text list](#) | [Issue list](#) | [Use cases and examples](#)

1

2

3

Number list

This function returns the **number** of elements in a **number list**.

Syntax

count(numberList) #Output: Number

Examples

Parser expression	Description
count([1, 1, 2, 2])	This example returns 4
count(subtasks()) - count(fieldValue({00012}, subtasks()))	This example returns the number of sub-tasks with an empty "Due Date" . To achieve this, the following functions are used: <ul style="list-style-type: none">• count()• subtasks()• fieldValue()

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
numberList	<div>NUMBER LIST</div>	Any given number list.

Output

This function returns a

NUMBER

This variant counts how often a **given number** appears in a **number list**.

Syntax

```
count(number, numberList) #Output: Number
```

Examples

Parser expression	Description
<pre>count(1, [1, 1, 2, 2, 1, 0])</pre>	This example returns 3

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
number	<input type="text" value="NUMBER"/>	Any given number.
numberList	<input type="text" value="NUMBER LIST"/>	Any given number list.

Output

This function returns a



Text list

This function returns the **number** of elements in a **text list**.

Syntax

```
count(textList) #Output: Number
```

Examples

Parser expression	Description
<pre>count(["blue", "red", "blue", "black"])</pre>	This example returns 4

```
count(distinct(fieldValue(%{00094},
subtasks()))))
```

This example returns the **number** of **components** selected among all **subtasks**.

To achieve this, the following functions are used:

- [distinct\(\)](#)
- [fieldValue\(\)](#)
- [subtasks\(\)](#)

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
textList	<input type="text" value="TEXT LIST"/>	Any given text list.

Output

This function returns a

This variant **counts** how often a given **text** appears in a **text list**.

Syntax

```
count(text, textList) #Output: Number
```

Examples

Parser expression	Description
<pre>count("blue", ["blue", "blue", "red", "red", "blue", "green"])</pre>	This example returns 3

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
text	<input type="text" value="TEXT"/>	Any given text.
textList	<input type="text" value="TEXT LIST"/>	Any given text list.

Output

This function returns a NUMBER

Issue list

This functions returns the **number** of elements in a given **issue list**.

Syntax

```
count(issueList) #Output: Number
```

Examples

Parser expression	Description
<pre>count(subtasks())</pre>	<p>This example returns the number of all subtasks.</p> <p>To achieve this, the following function is used:</p> <ul style="list-style-type: none">• subtasks()
<pre>count(filterByIssueType(subtasks(), "Technical task"))</pre>	<p>This example returns the number of all subtasks that are Technical tasks.</p> <p>To achieve this, the following functions are used:</p> <ul style="list-style-type: none">• filterByIssueType()• subtasks()
<pre>count(linkedIssues("is blocked by"))</pre>	<p>This example returns the number of all linked blocking issues.</p> <p>To achieve this, the following functions are used:</p> <ul style="list-style-type: none">• linkedIssues()
<pre>count(filterByResolution(linkedIssues("is blocked by"), ""))</pre>	<p>This example returns the number of unresolved blocking issues.</p> <p>To achieve this, the following functions are used:</p> <ul style="list-style-type: none">• filterByResolution()• linkedIssues()

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()).







Output

This function returns a NUMBER



Use cases and examples

Use case	JWT feature	Workflow function	Field type	Automated action	Parser functions
Prevent a transition depending of the number of components	 	Logical validator Logical condition			numberOfSelectedItems()
Number of issues in the same version			Number		count() issuesFromJQL()
Block a transition until all sub-tasks have certain fields populated	 	Logical validator Logical condition			count() filterByPredicate() subtasks()
Validate only issue links created in transition screen		Logical validator			count() transitionLinkedIssues() filterByProject() filterByIssueType()
Close epic when stories are done				Transition issue action	count() filterByStatus() issuesUnderEpic()
Automatically close parent when all sub-tasks are done				Transition issue action	count() filterByStatus() siblingSubtasks()
Create several issues combining fields		Create issue			nthElement() toStringList() modulus() count() ceil()
Number of open subtasks			Number		count() filterByPredicate() subtasks()
Number of unresolved blocking issues			Number		count() filterByResolution() linkedIssues()
Number of times a resolved issue has been rejected			Number		count() timesOfTransition()
Number of times a custom field has been changed			Number		count() fieldChangeTimes()
Reject duplicated file names in attachments	 	Logical validator Logical condition			count() toStringList() distinct()

Prevent issue creation with the same field value (Boolean)		Logical validator Validation based on JQL query	count() issuesFromJQL()
Ensure that all issues linked with a certain issue link type have "Due Date" field set	 	Logical validator Logical condition	count() linkedIssues() fieldValue()
Block an Epic's transition until all the issues under that Epic are resolved		Logical validator Logical condition	count() issuesUnderEpic() filterByResolution()
Block an Epic's transition depending on linked issues status and due date	 	Logical validator Logical condition	count() filterByPredicate() linkedIssues()