

Examples of Math-Time Expressions

Mathematical calculations

| Expression | Returned Value |
|---|--|
| <code>max(count(subtasks(%{...parentIssueKey})) - 1, 0)</code> or <code>count(siblingSubtasks())</code> | For a sub-task, the number of sibling sub-tasks. Function <code>max(x, y)</code> is used to avoid returning -1 when used with non-sub-task issues. |
| <code>{...cf10000} = null ? 1 : {...cf10000} + 1</code> | Formula to increment a numeric custom field, setting it to 1 if it's initially unset. |
| <code>{...cf10000} + {...cf10001} + {...cf10003}</code> | Formula for summing 3 numeric custom fields when we are certain that all 3 the fields are initialized . In case any of these fields is not initialized, an error is raised and any of the following 2 expression examples should be used. |
| <code>((...cf10000) = null ? 0 : {...cf10000}) + ((...cf10001) = null ? 0 : {...cf10001}) + ((...cf10003) = null ? 0 : {...cf10003})</code> | Formula for summing 3 numeric custom fields when some of them may be uninitialized . When any of this fields is not initialized a zero value is assumed. |
| <code>sum([...cf10000], {...cf10001}, {...cf10003})</code> | A more compact syntax for summing three numeric custom fields when some of them may be uninitialized . |

On this page

- [Mathematical calculations](#)
- [Date-Time calculations](#)
- [Showing Time Durations in different formats](#)

Date-Time calculations

| Expression | Returned Value |
|--|--|
| <code>{...duedate} - 6 * {DAY}</code> | Calculates a date 6 natural days earlier than Due Date |
| <code>addTimeSkippingWeekends({...created}, 36*{HOUR} + 45*{MINUTE}, LOCAL)</code> | Returns a date-time value equivalent to adding 36 hour and 45 minutes to date and time of issue creation, skipping the periods of time which correspond to weekend. |
| <code>addTimeSkippingWeekends({...created}, 36*{HOUR} + 45*{MINUTE}, LOCAL, {FRIDAY}, {SATURDAY})</code> | Same as previous expression, but using Israeli weekend. Israeli weekend is on Friday and Saturday. |
| <code>addDaysSkippingWeekends({...duedate}, -6, LOCAL)</code> | Calculates a date 6 work days earlier than Due Date for Jira Server's local timezone. Work days depend on timezone, since certain time moment maybe Sunday in certain time zones, and Monday in another ones. |

| | |
|---|---|
| <code>subtractDatesSkippingWeekends({...duedate}, {...currentDateTime}, LOCAL) / {DAY}</code> | Returns the number of working days from Current Date and Time to Due Date, i.e., skipping weekends in Jira server's timezone. |
| <code>round((({...currentDateTime} - {...created}) / {HOUR}))</code> | Number of hours since issue creation. Function round() approximates the number of hours to the nearer integer. |
| <code>floor((({...duedate} - {...currentDateTime}) / {DAY}))</code> | Number of days to Due Date |
| <code>datePart({...currentDateTime}, LOCAL) + (dayOfTheWeek({...currentDateTime}, LOCAL) = 7 ? 6 : 6 - dayOfTheWeek({...currentDateTime}, LOCAL)) * {DAY}</code> | Returns a date value for next Friday , or for today if it's Friday |
| <code>datePart({...currentDateTime}, LOCAL) + (dayOfTheWeek({...currentDateTime}, LOCAL) = 6 ? 7 : (dayOfTheWeek({...currentDateTime}, LOCAL) = 7 ? 6 : 6 - dayOfTheWeek({...currentDateTime}, LOCAL))) * {DAY}</code> | Returns a date value for next Friday , even if today is Friday. |
| <code>floor(subtractDatesSkippingWeekends({...currentDateTime}, {...created}, LOCAL) / {DAY}) + " days " + floor(modulus(subtractDatesSkippingWeekends({...currentDateTime}, {...created}, LOCAL), {DAY}) / {HOUR}) + " hours " + round(modulus(subtractDatesSkippingWeekends({...currentDateTime}, {...created}, LOCAL), {HOUR}) / {MINUTE}) + " minutes"</code> | Calculates the time since issue creation skipping weekends, and shows it as a text like this: 12 days 6 hours 34 minutes . |
| <code>floor((({...currentDateTime} - {...created}) / {DAY}) + " days " + floor(modulus((({...currentDateTime} - {...created}), {DAY}) / {HOUR}) + " hours " + round(modulus((({...currentDateTime} - {...created}), {HOUR}) / {MINUTE}) + " minutes"</code> | Calculates the time since issue creation, and shows it as a text like this: 12 days 6 hours 34 minutes . |

On this page

- [Mathematical calculations](#)
- [Date-Time calculations](#)
- [Showing Time Durations in different formats](#)

Showing Time Durations in different formats

The following examples are string expressions in **advanced parsing mode**.

| Expression | Returned Value | Notes |
|---|--|---|
| <code>formatDuration({...currentDateTime} - {...created})</code> | Calculates the time since issue creation, and shows it as a text using whole words like: 12 days 6 hours 34 minutes . | |
| <code>shortFormatDuration({...currentDateTime} - {...created})</code> | Calculates the time since issue creation, and shows it as a text using abbreviations like: 12 d 6 h 34 m . | |
| <code>formatDuration(subtractDatesSkippingWeekends({...currentDateTime}, {...created}, LOCAL))</code> | Calculates the time since issue creation skipping weekends, and shows it as a text like: 12 days 6 hours 34 minutes . | |
| <code>formatWorkDuration({...currentDateTime} - {...created})</code> | Calculates the time since issue creation, and shows it as text, but using the workday and workweek defined at time tracking configuration instead of 24 hours per day and 7 days per week. | Example: <code>formatWorkDuration(24 * {HOUR} + 5 * {MINUTE})</code> returns " 3 days 5 minutes " when we use 8 hours per workday . |

| | | |
|---|---|---|
| <code>shortFormatWorkDuration({...currentDateTime} - {...created})</code> | Similar to the previous expression but shows the result using abbreviations instead of whole words. | Example: <code>shortFormatWorkDuration(24 * {HOUR} + 5 * {MINUTE})</code> returns " 3d 5m " when we use 8 hours per workday . |
|---|---|---|

On this page

- [Mathematical calculations](#)
- [Date-Time calculations](#)
- [Showing Time Durations in different formats](#)