

# Data types

Throughout the documentation we refer to **data types** that can be used in the **expression parser** and its functions and field codes.

Data types define what kind of data a function needs as **input** and what you should expect as **output**.

## Data types in JWT

Data type	Description
TEXT	This type represents any kind of text or character string including all kinds of select and multi-select fields
NUMBER	<p>This type represents numeric values, and is also used to store <b>date</b>, <b>time</b> and <b>date-time</b> values. Learn more about <a href="#">Dates, times and time zones</a></p> <p>If you need the output as <b>text</b>, simply add % in front of the field code or directly <b>insert</b> the field code as text.</p>
LIST	<ul style="list-style-type: none"><li>• <b>Number list</b>: This type represents a collection of <b>numeric</b> values returned by various functions. The size may vary from <b>0</b> to any number of <b>numeric</b> values.</li><li>• <b>Text list</b>: This type represents a collection of <b>text</b> values returned by various functions. The size may vary from <b>0</b> to any number of <b>text</b> values.</li><li>• <b>Issue list</b>: This type represents a collection of <b>issues</b>. The size may vary from <b>0</b> to any number of <b>issues</b>. It is returned by issue selection or filtering functions.</li></ul> <p>Learn more about <a href="#">Lists</a></p> <p>If you need to "flatten" the list, or simply convert it to a text, to be written into any field, comment, etc., make sure to have a look at the <a href="#">toString()</a> function!</p>
BOOLEAN	<p>Some <a href="#">operators</a>, workflow functions, <a href="#">parser functions</a>, and automation elements return a <b>logical</b>, or <b>boolean</b>, value of <b>true</b> or <b>false</b>.</p> <p>These examples include, but are not limited to:</p> <ul style="list-style-type: none"><li>• <a href="#">Logical condition</a></li><li>• <a href="#">Logical validator</a></li><li>• <a href="#">Logical mode</a></li><li>• <a href="#">Boolean condition</a></li></ul>

## Converting (casting) data types

**Converting** (or **casting**) data types is very important throughout all functions provided by JWT.

To be able to **set**, **transform** or **calculate** values it might be necessary to turn a **text** value to a **number**, a **number** or a **timestamp** to a **text**, and even individual **text** or **number** values to a **list**.

The following functions will help you to achieve that.

[Available functions](#)



Function	Short description	Output
<a href="#">issueKeysToIssueList()</a>	Converts a <b>text containing issue keys</b> into an <b>issue list</b> .	ISSUE LIST
<a href="#">toInteger()</a>	Converts <b>text</b> into an <b>integer</b> .	NUMBER
<a href="#">toNumber()</a>	Converts <b>text</b> into a <b>number</b> .	NUMBER
<a href="#">toNumberList()</a>	Converts <b>text containing numbers</b> into a <b>number list</b> .	NUMBER LIST
<a href="#">toString()</a>	Converts <b>numbers</b> and <b>lists</b> into plain <b>text</b> .	TEXT
<a href="#">toStringList()</a>	Converts <b>text</b> into a <b>text list</b> .	TEXT LIST

Whenever you write a numeric term at the right-hand side of **the + operator** or a **comparison operator** like = , and the left-hand side is occupied by a text term, JWT will automatically transform the right-hand side term into a text [toString\(\)](#).

Operator	Example
+	"His age is " + 30 is equivalent to "His age is " + <code>toString(30)</code>
= (any comparison operator)	"30" = 30 is equivalent to "30" = <code>toString(30)</code>

If you still have questions, feel free to refer to our [support](#) team.