

COMMON FORMULAS

split

This formula divides a string around a specified character and returns an array of strings.

Syntax

String.split(char)

- String - An input string value. You can use a datapill or static value.
- char - (optional) The character at which to split the text. This is case sensitive. If no character is defined, then by default, strings are split by white spaces.

Sample usage

Formula	Result
"Ms-Jean-Marie".split("-")	["Ms", "Jean", "Marie"]
"Ms Jean Marie".split	["Ms", "Jean", "Marie"]
"Split string".split()	["Split", "string"]
"Split string".split("t")	["Split", " s", "ring"]
"01/23/2014".split("/")	["01", "23", "2014"]
"01/23/2014".split("/").join("-")	"01-23-2014"

How it works

This formula looks for the specific character in the input string. Every time it is found, the input will be split into a new string.

last

This formula divides a string around a specified character and returns an array of strings.

Syntax

List.last(**number**)

- List - An input list.
- **number** - (optional) The number of items to retrieve from the list. If not specified, the formula will return only *one* item.

Sample usage

Formula	Result
<code>["One","Two","Three","Four","Five"].last()</code>	"Five"
<code>["One","Two","Three","Four","Five"].last(2)</code>	["Four", "Five"]
<code>[1,2,3,4,5].last()</code>	5
<code>[1,2,3,4,5].last(3)</code>	[3,4,5]

How it works

This formula returns the last *n* items from a list. If *n* is greater than one, the output is formatted as a list.

present?

This formula will check the input and if there is a value present, it will return true. If the input is nil, boolean false, an empty string, or an empty list, the formula will return false.

Syntax

Input.present?

- Input - An input datapill. It can be a string, number, date, or list datatype.

Sample usage

Formula	Result
"Any Value".present?	true
123.present?	true
0.present?	true
"2017-04-02T12:30:00.000000-07:00".present?	true
nil.present?	false
"".present?	false
[],.present?	false

How it works

If the input is null, an empty string or an empty list, the formula will return false. For any other data, it returns true.

to_date

This formula converts the input data into a date. Returns the date formatted as YYYY-MM-DD.

Syntax

String.to_date(format:char)

- **String** - An input datetime or a string that describes a date or datetime.
- **format** - (optional) The date format of the input written as a string. If not specified, Workato will parse the input string automatically.

Sample usage

Formula	Result
<code>"23-01-2020 10:30 pm".to_date(format: "DD-MM-YYYY")</code>	"2020-01-23"
<code>"01-23-2020 10:30 pm".to_date(format: "MM-DD-YYYY")</code>	"2020-01-23"
<code>"2020/01/23".to_date(format: "YYYY/MM/DD")</code>	"2020-01-23"

How it works

Converts the input data into a date datatype.

to_s

Converts data to a string (text) datatype.

Syntax

`Input.to_s`

- `Input` - An input data. You can use number, array, object, or datetime datatypes.

Sample usage

Formula	Result
<code>-45.67.to_s</code>	"-45.67"
<code>"123".to_s</code>	"123"
<code>[1,2,3].to_s</code>	"[1,2,3]"
<code>{key: "Workato"}.to_s</code>	"{:key=>"Workato"}"
<code>""2020-06-05T17:13:27.000000-07:00"".to_s</code>	"2020-06-05T17:13:27.000000-07:00"
<code>"2020-06-05T17:13:27.000000-07:00".to_s(:short)</code>	"05 Jun 17:13"
<code>"2020-06-05T17:13:27.000000-07:00".to_s(:long)</code>	"June 05, 2020 17:13"

How it works

This formula returns a string representation of the input data.

first

This formula returns the first item in a list.

It can also be used to return the first n items in a list. In this case, the output will be formatted as a list.

Syntax

List.first(**number**)

- List - An input list.
- **number** - (optional) The number of items to retrieve from the list. If not specified, the formula will return only *one* item.

Sample usage

Formula	Result
<code>["One","Two","Three","Four","Five"].last()</code>	"One"
<code>["One","Two","Three","Four","Five"].last(2)</code>	["One", "Two"]
<code>[1,2,3,4,5].last()</code>	1
<code>[1,2,3,4,5].last(3)</code>	[1,2,3]

How it works

This formula returns the first n items from a list. If n is greater than one, the output is formatted as a list.

strftime

Returns a datetime input as a user-defined string.

Syntax

Date.strftime(**format**)

- **Date** - An input date or datetime.
- **format** - The format of the user-defined datetime written as a string.

Sample usage

Formula	Result
"2020-06-05T17:13:27.000000-07:00".strftime("%Y/%m/%d")	"2020/06/05"
"2020-06-05T17:13:27.000000-07:00".strftime("%Y-%m-%dT%H:%M:%S%z")	"2020-06-05T17:13:27-0700"
"2020-06-05T17:13:27.000000-07:00".strftime("%B %e, %l:%M%p")	"June 5, 5:13 pm"
"2020-06-05T17:13:27.000000-07:00".strftime("%A, %d %B %Y %k:%M")	"Friday, 05 June 2020 0:00"

Parameters

As shown above, each code (%B, %e, %l etc.) refers to a specific element of datetime. Static text and punctuation can also be added, such as commas (,), slashes (/), and colons (:). Refer to [Ruby documentation](#) for a full list of codes.

How it works

Allows the user to define a datetime format. Returns the datetime input in the specified format.

now

Returns the time and date at runtime in US Pacific Time Zone (PST).

Sample usage

Formula	Result
now	"2022-02-01T07:00:00.000000-08:00"
now + 8.hours	"2022-02-01T15:00:00.000000-08:00"
now +2.days	"2022-02-03T07:00:00.000000-08:00"

How it works

The formula calculates the timestamp when a job is being processed. Each step using this formula will return the timestamp at which the step runs.

gsub

Replace parts of a text string. Returns a new string with the replaced characters.

Syntax

String.gsub(**find**,**replace**)

- **String** - An input string. You can use a datapill or static value.
- **find** - The string to look for. You can use a `/pattern/` syntax.
- **replace** - The replacement string. You can define the replacement using a string or hash.

Sample usage

Formula	Result
"I have a blue house and a blue car".gsub("blue", "red")	"I have a red house and a red car"
"Jean Marie".gsub("J", "M")	"Mean Marie"
"Jean Marie".downcase.gsub("j", "M")	"Mean marie"

How it works

This formula works like find and replace. It takes two input parameters:

1. The first input is the string that you want to replace. This is case-sensitive, so make sure to type correctly in either uppercase or lowercase to find all occurrences that are an exact match.
2. The second input is the new string that will be used for replacing all occurrences of first input.

join

Combines all items in a list into a text string. A separator is placed between each item.

Syntax

List.join(separator)

- List - An input of list datatype.
- separator - The character to add between items when they are joined. If no separator is specified, the list items will be joined together.

Sample usage

Formula	Result
["Ms", "Jean", "Marie"].join("-")	"Ms-Jean-Marie"
[1,2,3].join("--")	"1--2--3"
["ab", "cd", "ef"].join	"abcdef"

How it works

The list items are combined into a single text string. The separator characters are added between each item.