If

# If#

Use the If node to split a workflow conditionally based on comparison operations.

## Add conditions#

Create comparison Conditions for your If node.

• Use the data type dropdown to select the data type and comparison operation type for your condition. For example, to filter for dates after a particular date, select Date & Time > is after.

• The fields and values to enter into the condition change based on the data type and comparison you select. Refer to Available data type comparisons for a full list of all comparisons by data type.

Select Add condition to create more conditions.

### Combining conditions#

You can choose to keep data:

• When it meets all conditions: Create two or more conditions and select AND in the dropdown between them.

• When it meets any of the conditions: Create two or more conditions and select OR in the dropdown between them.

## Templates and examples#

by Eduard

by Jonathan

by Jonathan

## Branch execution with If and Merge nodes#

0.236.0 and below

n8n removed this execution behavior in version 1.0. This section applies to workflows using the v0 (legacy) workflow execution order. By default, this is all workflows built before version 1.0. You can change the execution order in your workflow settings.

If you add a Merge node to a workflow containing an If node, it can result in both output data streams of the If node executing.

One data stream triggers the Merge node, which then goes and executes the other data stream.

For example, in the screenshot below there's a workflow containing an Edit Fields node, If node, and Merge node. The standard If node behavior is to execute one data stream (in the screenshot, this is the true output). However, due to the Merge node, both data streams execute, despite the If node not sending any data down the false data stream.

## Related resources#

Refer to Splitting with conditionals for more information on using conditionals to create complex logic in n8n.

If you need more than two conditional outputs, use the Switch node.

## Available data type comparisons#

### String#

String data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty

• is equal to

• is not equal to

• contains

• does not contain

• starts with

• does not start with

• ends with

• does not end with

• matches regex

• does not match regex

### Number#

Number data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty

• is equal to

• is not equal to

• is greater than

• is less than

• is greater than or equal to

• is less than or equal to

### Date & Time#

Date & Time data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty

• is equal to

• is not equal to

• is after

• is before

• is after or equal to

• is before or equal to

### Boolean#

Boolean data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty

• is true

• is false

• is equal to

• is not equal to

### Array#

Array data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty

• contains

• does not contain

• length equal to

• length not equal to

• length greater than

• length less than

• length greater than or equal to

• length less than or equal to

### Object#

Object data type supports these comparisons:

• exists

• does not exist

• is empty

• is not empty