

Understanding Logic Operators

Choosing the wrong operator in a logic rule can cause you a lot of grief. In this tutorial we'll review each available operator and how it works. Later in the tutorial we will cover each Alchemer question type and its recommended operator/s.

Logic Operators Explained

There are 18 logic operators available to use in setting up logic conditions. Below we'll discuss each operator, how it is evaluated, and the question types it works best with.

is exactly equal to

This operator is the default for an open-text question type. This will evaluate as an exact match *including case*. Thus, the following two answers would **not** match:

- yes - used in logic rule
- Yes - answer provided by respondent

To better handle case-sensitivity in logic based on open-text questions you can use the **Capitalize each word** option on the **Validation** tab to increase consistency in text fields. Or better yet, using the **Auto Suggest** with the option to **Only allow answers from the auto suggest list**.

This operator can also be used for multi-select questions like the checkbox question type to set every option that must be given to trigger the action. Thus, if you have 2 answers checked and you are using *Is Exactly Equal To*, the respondent must check both of those answers and only those answers to trigger the logic action.

is not exactly equal to

The opposite of *is exactly equal to*, it is typically used for open-text question types to set the criteria that it should not match for the logic action to be triggered.

As with *is exactly equal to*, this will evaluate as an exact match including case. Thus, the following two answers would **not** match:

- yes - used in logic rule
- Yes - answer provided by respondent

To better handle case-sensitivity in logic based on open-text questions you can use the **Capitalize each word** option on the **Validation** tab to increase consistency in text fields. Or better yet, using the **Auto Suggest** feature with the option to **Only allow answers from the auto suggest list**.

is one of the following answers

One of the most versatile and powerful operators, it is used for single answer (radio button), multiple answer (checkbox), lists, and table question types. This operator is used to search through all the answers given for that particular question (or row in the case of a table) and trigger the logic action if **ANY** of the values selected in the logic rule are found.

A common use is to trigger the logic action if any of the selected options in a checkbox question are chosen. Not only if they choose those and only those values which are in the logic rule, but rather if **ANY** are present in their answer. If the rule has option1 and option2 selected, then a respondent answering option1 and option3 would qualify and trigger the logic action.

is not one of the following answers

The opposite of 'is one of the following answers', this operator is used to search through all the answers given for that particular question (or row in the case of a table) and trigger the logic action if **NONE** of the values selected in the logic rule are found.

A common use is to trigger the logic action if none of the selected options in a checkbox question are chosen. If the rule has option2 and option3 selected, then a respondent answering option1 and option4 would qualify and trigger the logic action.

is answered

This operator is very versatile. It is usually used for questions that are unrequired text fields. If the text field has an entry, then the rule is true.

is not answered

The opposite of is answered, this operator is usually used for unrequired text fields. If the question has **no** answer selected or in the case of a textbox has **no** answer given, then the rule is true.

> (greater than)

Used for any question where the expected response is a single answer and is numeric. This could include the Number question type or a Radio Button or a Dropdown Menu question with numeric reporting values. If the answer given is greater than the value set in the logic builder, then the logic action will trigger.

≥ (greater than or equal to)

Used for any question where the expected response is a single answer and is numeric. This could include the Number question type or a Radio button or a Dropdown Menu question with numeric reporting values. If the answer given is greater than or equal to the value set in the logic builder, then the logic action will trigger.

< (less than)

Used for any question where the expected response is a single answer and is numeric. This could include the Number question type or either a Radio button or a Dropdown Menu question with numeric reporting values. If the answer given is less than the value set in the logic builder, then the logic action will trigger.

≤ (less than or equal to)

Used for any question where the expected response is a single answer and is numeric. This could include the Number question type or a Radio button or a Dropdown Menu question with numeric reporting values. If the answer given is less than or equal to the value set in the logic builder, then the logic action will trigger.

Date ≥ (after or equal to)

Used for any question where the expected response is a single answer and is formatted as a date. The value can be formatted in many common formats (See some sample formats).

Date ≤ (before or equal to)

Used for any question where the expected result is a single answer and is formatted as a date. The value can be formatted in many common formats (See some sample formats).

matches regex pattern

[Advanced Users Only] Regular Expression matching is very powerful and allows you to build a complex logic magic based upon a specific Regular Expression. To see some examples of simple Regular Expression matches, View the Intro section (stop at the POSIX section) of this tutorial. You can practice regular expression matching using the this online real-time regular expression tester.

does not match pattern

[Advanced Users Only] Regular Expression matching is very powerful and allows you to build a complex logic magic based upon a specific Regular Expression. This function will trigger the logic action if the RegEx pattern is **not** found. To see some examples of simple Regular Expression matches, View the Intro section (stop at the POSIX section) of this tutorial. You can practice regular expression matching using the this online real-time regular expression tester.

In an open answer essay box, you may wish to skip over the next page of questions about the staff if they did not mention 'staff' anywhere in their answer. Choose the *does not match pattern* and set the value as: **staff**

is true/is false

The Is True/Is False operators are available to use *only* in combination with the Is Mobile (true/false) option in the left most dropdown. This evaluates whether the respondent is using a mobile browser. (This is the **only** operator you should use with Is Mobile.)

This comes in very handy when creating surveys that you would like to optimize for mobile browsers without limiting the display and functionality for non-mobile respondents.

contains

The contains operator can be used to set up logic based on Textbox, Essay or Hidden Value open-text fields. The contains operator will parse the entire text of a text field response and will evaluate as an exact string match including case.

Question Types: Recommended Operators

Now that you understand the operators it's time to learn which operators are recommended with each question type. Below we'll cover the default and recommended operators for when using various question types as trigger questions.

Open-Text Questions (Textbox, Email, Essay)

Default Operator: *is exactly equal to*

If matching for a specific answer including case, *is exactly equal to* is a perfect fit.

This will evaluate as an exact match including case. Thus, the following two answers would **not** match:

- yes - used in logic rule
- Yes - answer provided by respondent

To better handle case-sensitivity in logic based on open-text questions you can use the **Capitalize each word** option on the **Validation** tab to increase consistency in text fields. Or better yet, using the **Auto Suggest** with the option to **Only allow answers from the auto suggest list**.

Other Recommended Operators: is Answered, is not answered, contains

One of the most common logic setups for text fields like these is to set up a question/page to show when the question is either answered or not answered.

Advanced Operator: *matches regex pattern*

If you are looking for the presence of a specific word anywhere in the answer, then using the *matches regex pattern operator* would allow you to drive logic off of the presence of a particular value anywhere in the answer. See above *matches regex pattern* for relevant links.

Numeric Open-Text Questions - (Number, Percent, Date, Currency)

Default Operator: *is exactly equal to*

Recommended Operators: *greater than, less than, greater than or equal to, less than or equal to*

If forcing a numeric value, such as Number or Percent, in the validation of your question, you can evaluate the rule with *is exactly equal to*, but most commonly you will want to compare the number to a certain other numeric threshold with the *greater than/less than* operators.

Single-Select Questions - (Radio Button, Dropdown Menu, Rating (Likert Scale), Image Select (single), Row in a Radio Button Grid, Row in a Dropdown Menu List)

Default Operator: *is one of the following*

The default operator will allow you to select **ANY** of the answers that will trigger the logic action. Therefore, if there are 5 possible answers to a multiple choice question, and 2 of the 5 should trigger the logic action, then you can check both items when using *is in list*, and either answer (if selected) would trigger the logic action. This is simpler and faster than adding multiple operators, one for each possible answer.

Multi-Select Question (Checkboxes, Row in a Checkbox Grid, Image Select (multiple))

Default Operator: *is one of the following*

The default operator allows you to select **ANY** of the answers that will trigger the logic action. If the respondent selects any of the answers in your rule's selected answers, even if they combine one of these answers with an answer you have not specified, the logic action

would be triggered. Use the below example:

Remove All Logic

Q 3. Checkbox is one of the following

- Asian
- Native Hawaiian or Other Pacific Islander
- Black/African-American
- White
- Hispanic/Latino
- American Indian/Alaska Native
- Other - Write In
- Prefer not to answer

+ Add Condition + Add Group

The checkbox based logic rule has both White and Hispanic/Latino checked. All of the following answers by a respondent would cause the logic action to trigger:

- White only
- Hispanic/Latino only
- White combined with any other answer not selected in your rule
- Hispanic/Latino combined with any other answer not selected in your rule
- Hispanic/Latino and White

Alternate Operator: *is exactly equal to*

If the logic rule should only trigger the logic action when the rule's selected values match **EXACTLY**, the operator *is exactly equal to* is the preferred choice. Using the below logic rule, only a respondent's answer of both **Caucasian** and **Hispanic** and *no other answers* would trigger the logic action:

Rows in Complex Tables (Star Rating)

Default Operator: *is exactly equal to*

Recommended Operator: *is one of the following*

In complex tables, each row represents multiple answers (a different answer for each column). *If there is only 1 column*, you can use *is exactly equal to*. However, most likely, you will want to drive logic off of the presence of a particular answer in any of the columns, such as a 1 star rating or an unsatisfied response in one of the dropdowns. In this case you would use *is one of the following answers*.

For example, in the below question you may want to trigger an action if the rating is a 1 star for any of the reps for any of the columns:

Star Rating Grid	Efficient	Knowledgeable	Helpful
Service Rep 1	✖ ★★★★★	✖ ★★★★★	✖ ★★★★★
Service Rep 2	✖ ★★★★★	✖ ★★★★★	✖ ★★★★★
Service Rep 3	✖ ★★★★★	✖ ★★★★★	✖ ★★★★★

The logic would look like this:

Remove All Logic

• Service Rep 1 ⌵ is one of the follc ⌵

or ⌵

• Service Rep 2 ⌵ is one of the follc ⌵ ✖

or ⌵

• Service Rep 3 ⌵ is one of the follc ⌵ ✖

[+ Add Condition](#) [+ Add Group](#)

Related Articles