Understanding Negative Logic Conditions

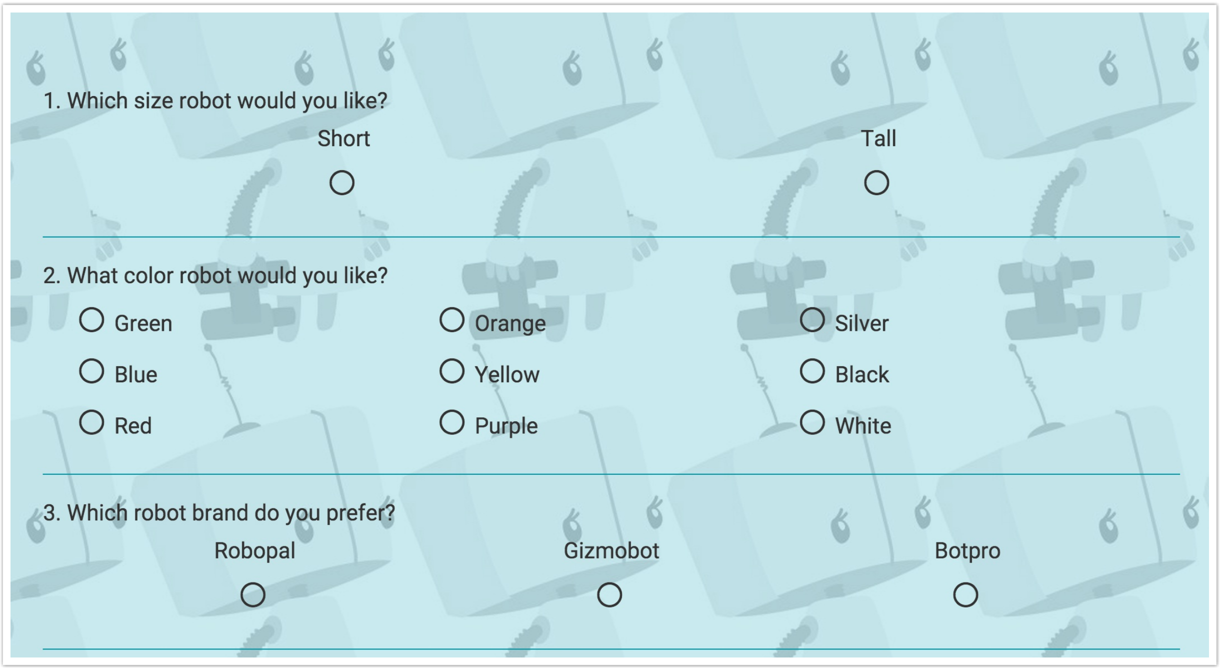
Using negative logic conditions, such as **is not exactly equal to**, **is not one of the following** or **is not answered**, can be very useful when building logic or filters. Negative logic conditions can save you time since, in many cases, negative conditions can accomplish the same result as positive conditions but with fewer steps.

That said, negative logic conditions are more difficult to think about and even harder to explain...On that note, let's give it a shot! Below is a primer for each condition listed above:

1. **Is not exactly Equal to:** Only show the question **IF** the answers a respondent provides does **NOT** equal the combination of answers provided. This operator works best in conjunction with questions that respondents can provide multiple answers to (Checkboxes).
2. **Is not one of the following:** Only show the question **IF** any of the answers set up in the logic conditions are **not** selected.
3. **Is not answered:** To fill this condition, an answer is not provided for the question that drives the logic condition. If that question is left blank, the question where logic conditions are being implemented will display.

The Efficiency of Negative Logic

Let's say, we ask customers to choose a size, color, and brand in our robot order form.



1. Which size robot would you like?

Short	Tall
○	○

2. What color robot would you like?

○ Green	○ Orange	○ Silver
○ Blue	○ Yellow	○ Black
○ Red	○ Purple	○ White

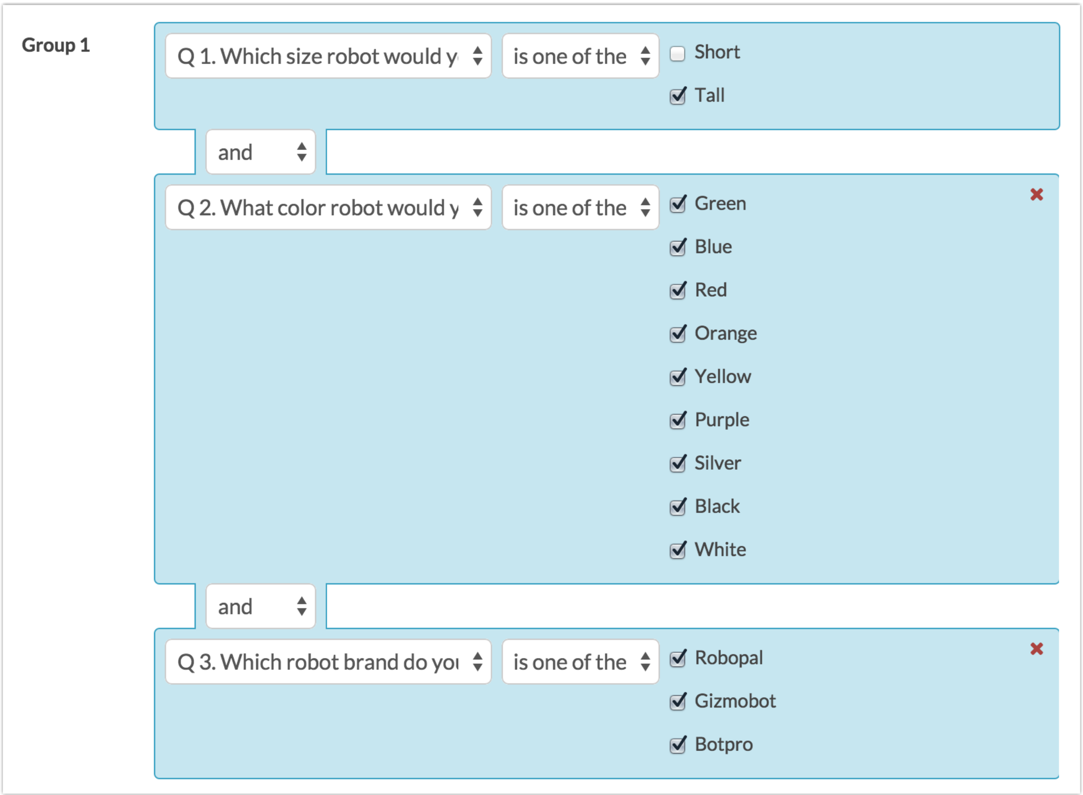
3. Which robot brand do you prefer?

Robopal	Gizmobot	Botpro
○	○	○

We have a coupon code that we want to show in a Text / Instruction Element on the next page.

The coupon applies to all combinations of robots except short, green, Robopal brand robots.

Below is the required set up to accomplish this using **positive logic**:



Group 1

Q 1. Which size robot would y | is one of the | □ Short ☑ Tall

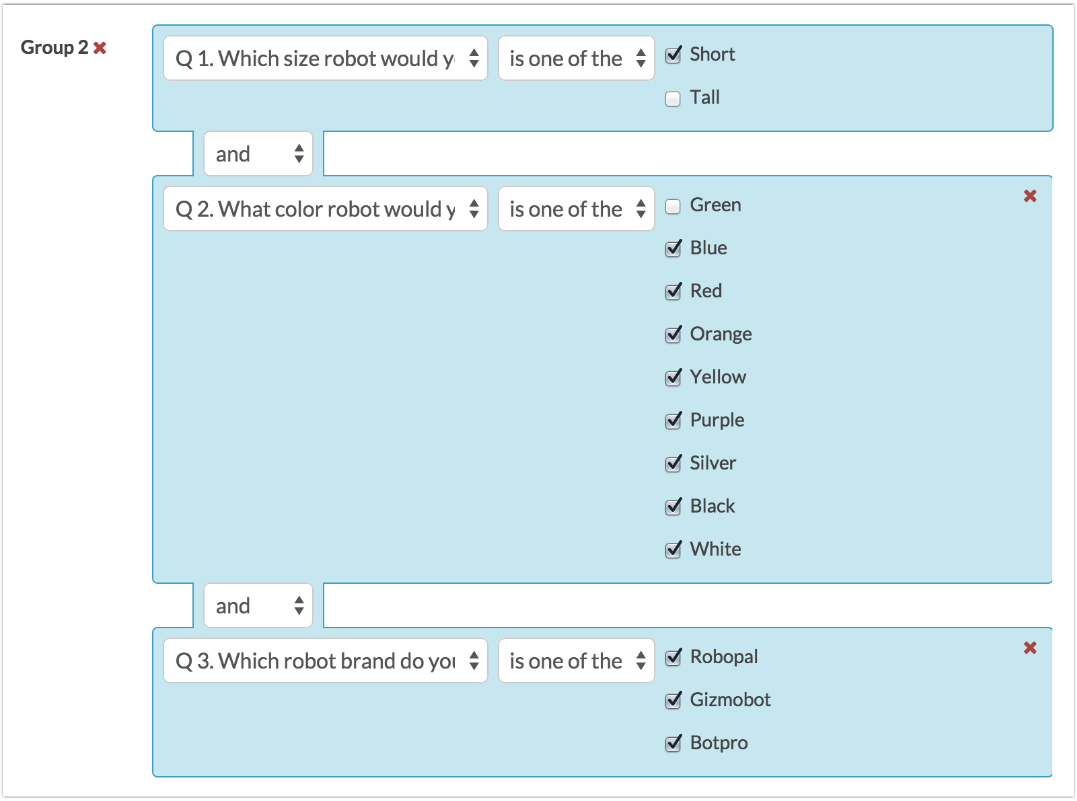
and

Q 2. What color robot would y | is one of the | ☑ Green ☑ Blue ☑ Red ☑ Orange ☑ Yellow ☑ Purple ☑ Silver ☑ Black ☑ White

and

Q 3. Which robot brand do yoı | is one of the | ☑ Robopal ☑ Gizmobot ☑ Botpro

OR



Group 2 ✖

Q 1. Which size robot would y — is one of the — ☑ Short □ Tall

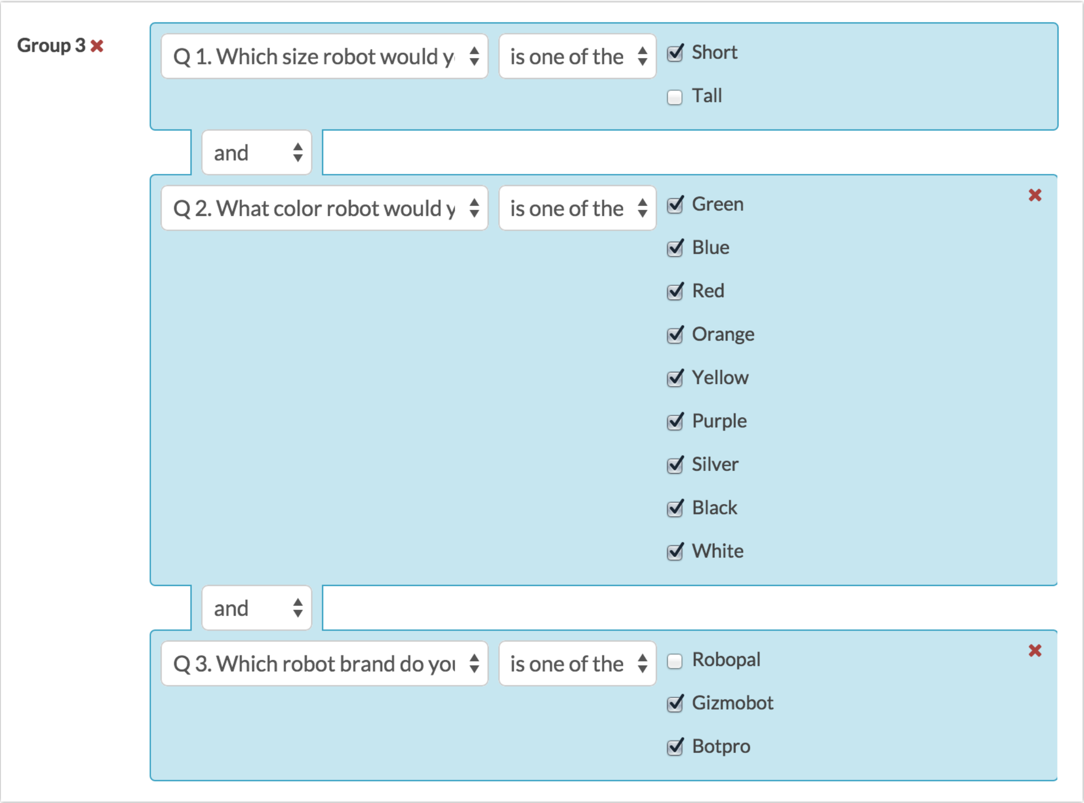
and

Q 2. What color robot would y — is one of the — □ Green ☑ Blue ☑ Red ☑ Orange ☑ Yellow ☑ Purple ☑ Silver ☑ Black ☑ White ✖

and

Q 3. Which robot brand do yoı — is one of the — ☑ Robopal ☑ Gizmobot ☑ Botpro ✖

OR



Group 3 ✖

Q 1. Which size robot would y — is one of the — ☑ Short □ Tall

and

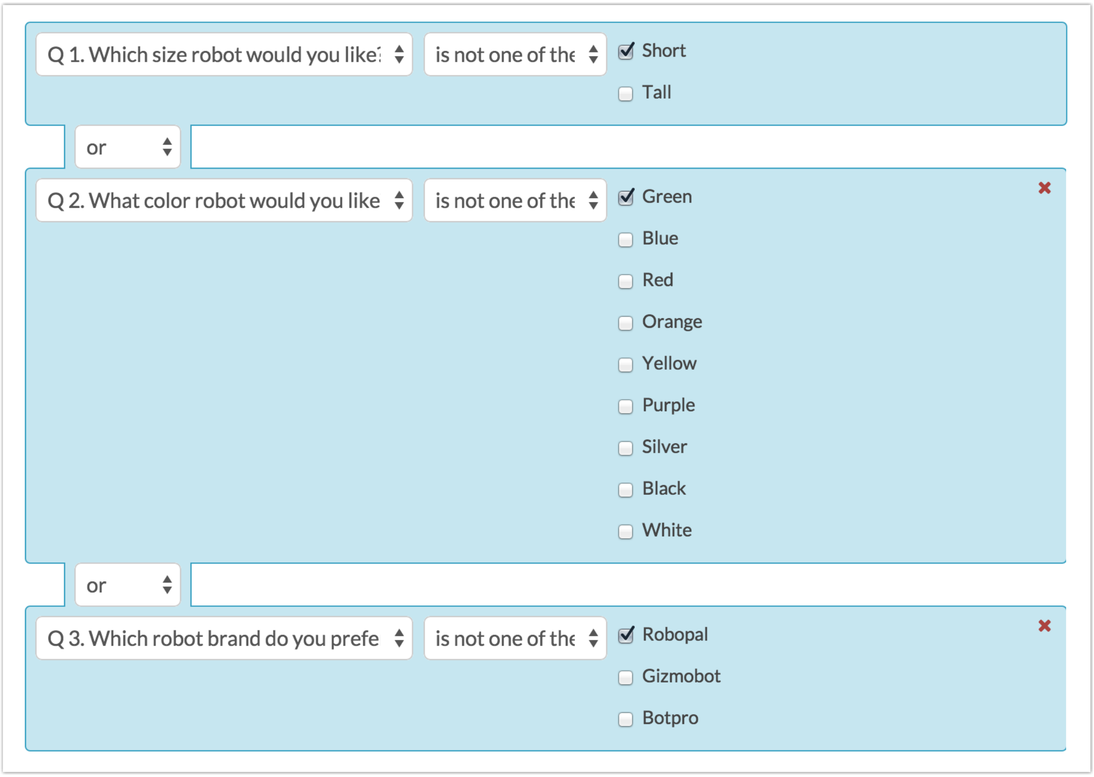
Q 2. What color robot would y — is one of the — ☑ Green ☑ Blue ☑ Red ☑ Orange ☑ Yellow ☑ Purple ☑ Silver ☑ Black ☑ White ✖

and

Q 3. Which robot brand do yoı — is one of the — □ Robopal ☑ Gizmobot ☑ Botpro ✖

Phew! As you can see, this requires 3 groups, each with 3 conditions and a lot of clicks!

Now, let's see how we can get the same result when using **negative logic**:



Q 1. Which size robot would you like? | is not one of the | ☑ Short ☐ Tall

or

Q 2. What color robot would you like | is not one of the | ☑ Green ☐ Blue ☐ Red ☐ Orange ☐ Yellow ☐ Purple ☐ Silver ☐ Black ☐ White

or

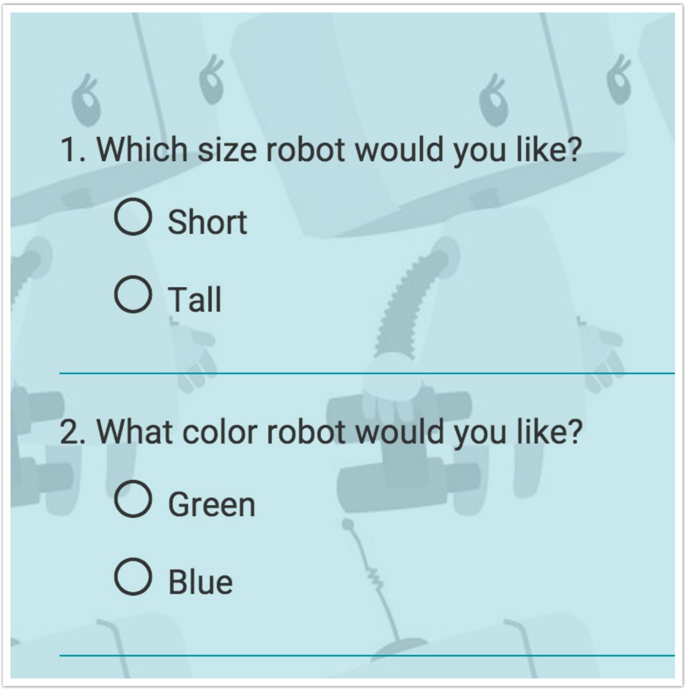
Q 3. Which robot brand do you prefe | is not one of the | ☑ Robopal ☐ Gizmobot ☐ Botpro

Note that negative logic requires a single grouping of logic with 3 conditions and fewer clicks. Consider how much time you could save!

And and Or Confusion

Notice in the above negative logic example the 3 conditions are joined by the **OR** operator. Were you thinking it should be AND? This is something almost all of us get wrong when setting up negative logic. Let's take a simpler example and examine why we used OR instead of AND to join our conditions.

Let's say, for example, after respondents answer the following two questions...



1. Which size robot would you like?

○ Short

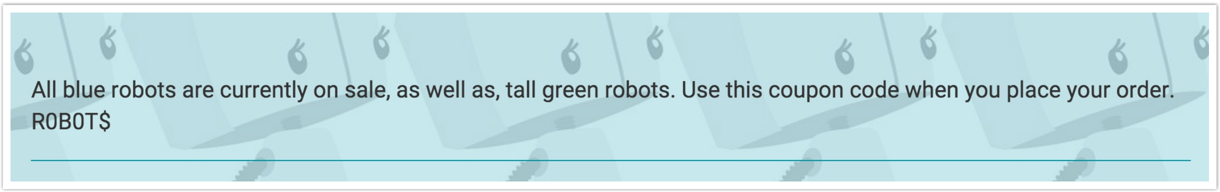
○ Tall

2. What color robot would you like?

○ Green

○ Blue

...we want to show the below Text / Instruction Element with a coupon code for all blue robots and tall, green robots.

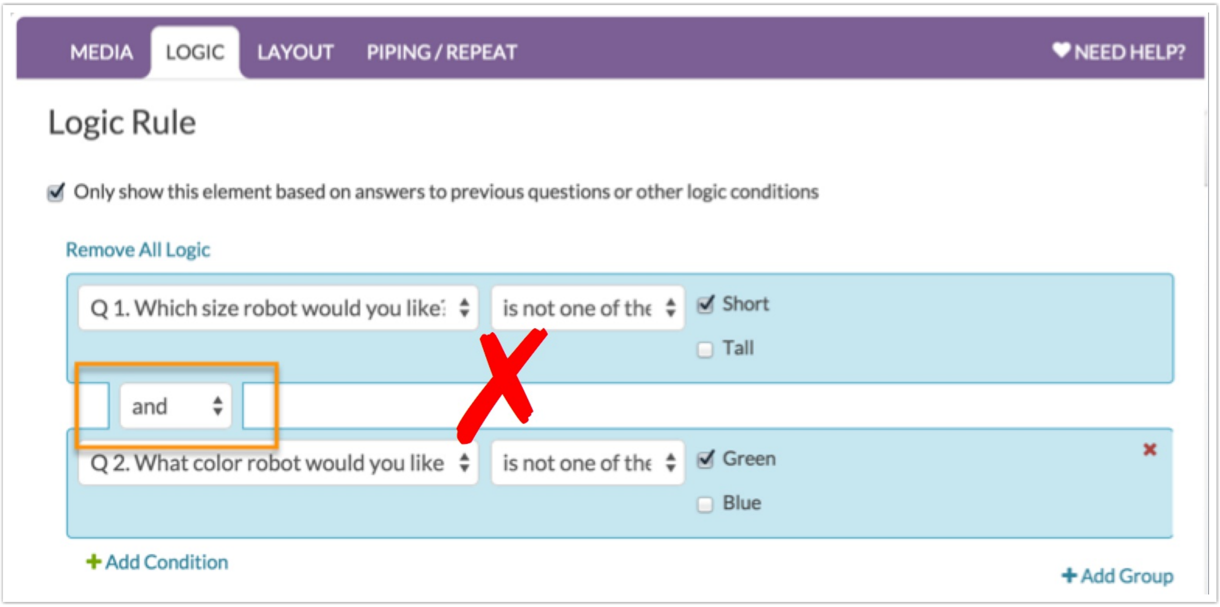


All blue robots are currently on sale, as well as, tall green robots. Use this coupon code when you place your order. R0B0T$

Because the group that we want to exclude is smaller than the group we want to include, it's more efficient to build the logic for the group we want to exclude.

We want to display the above coupon code for all blue robots and for tall green robots. In other words, we want to exclude short, green robots. More often than not we'll end up with the below logic rule; size is not short AND color is not green. Right? WRONG!

> Test out the use of the AND operator in a survey.



Logic Rule

☑ Only show this element based on answers to previous questions or other logic conditions

Remove All Logic

Q 1. Which size robot would you like? | is not one of the | ☑ Short □ Tall

and

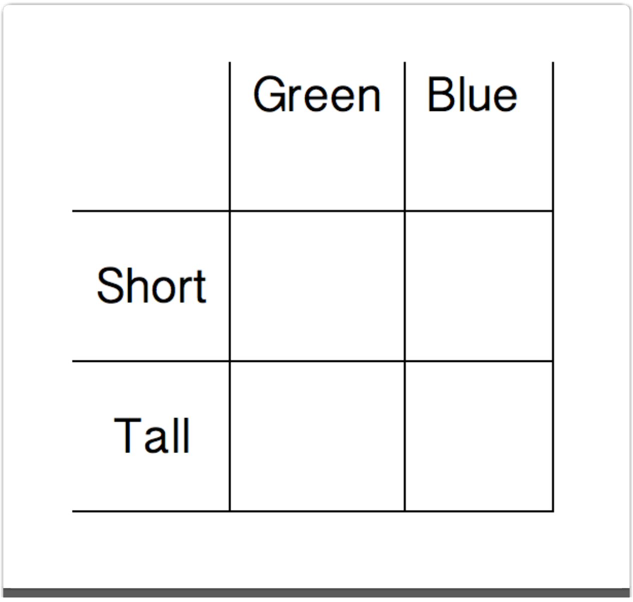
Q 2. What color robot would you like | is not one of the | ☑ Green □ Blue

If you contact our support team, they'll tell you to switch your AND to an OR. You'll switch them, test it out, see that it works and be on your merry way.

> Test out the use of the OR operator in a survey.

But wait! This is such a great learning opportunity! WHY does the switch of AND for OR fix this? We said we want to exclude short *and* robots; this seems like the correct operator! Let's dig a little deeper to understand why it is not.

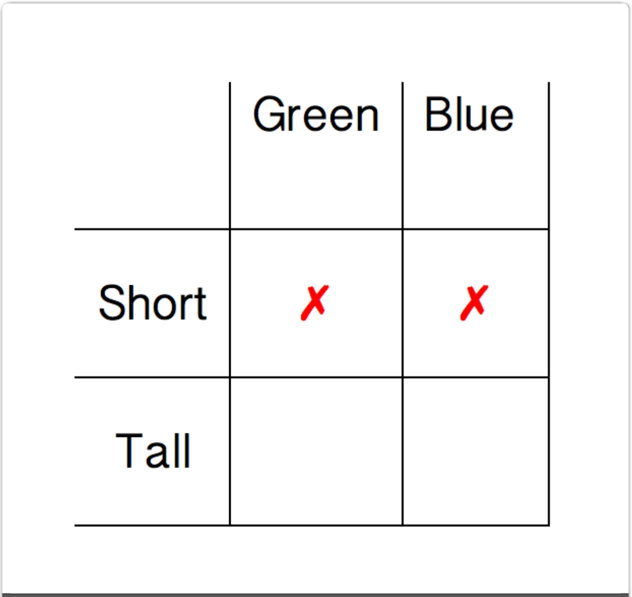
When setting up negative logic, we recommend setting up a logic chart like the below. This way you can check that your logic is to set up correctly. In our example, we have both short and tall green robots, as well as, short and tall blue robots so we have 4 possible combinations show here in our logic chart.



	Green	Blue
Short		
Tall		

Set up a chart to double check your negative logic

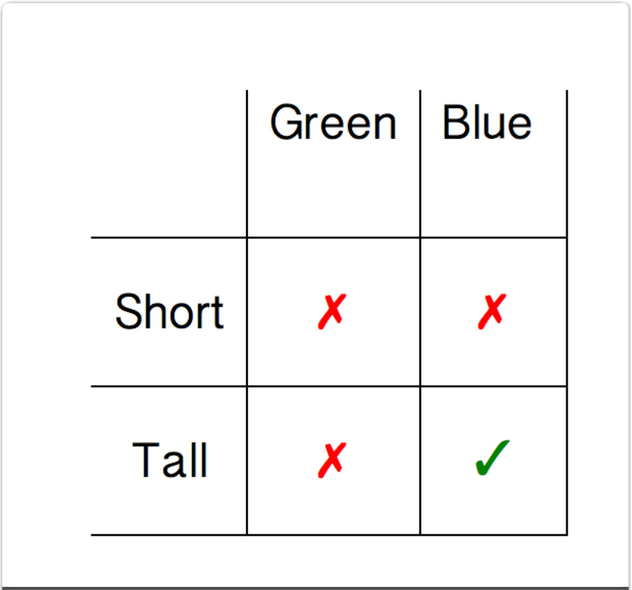
Let's apply the first logic condition we built above. **"Which size robot would you like?" is not Short.** This will display the question for all tall size responses. So far so good, the question will display for all tall robots.



	Green	Blue
Short	✗	✗
Tall		

"Which size robot would you like?" is not Short

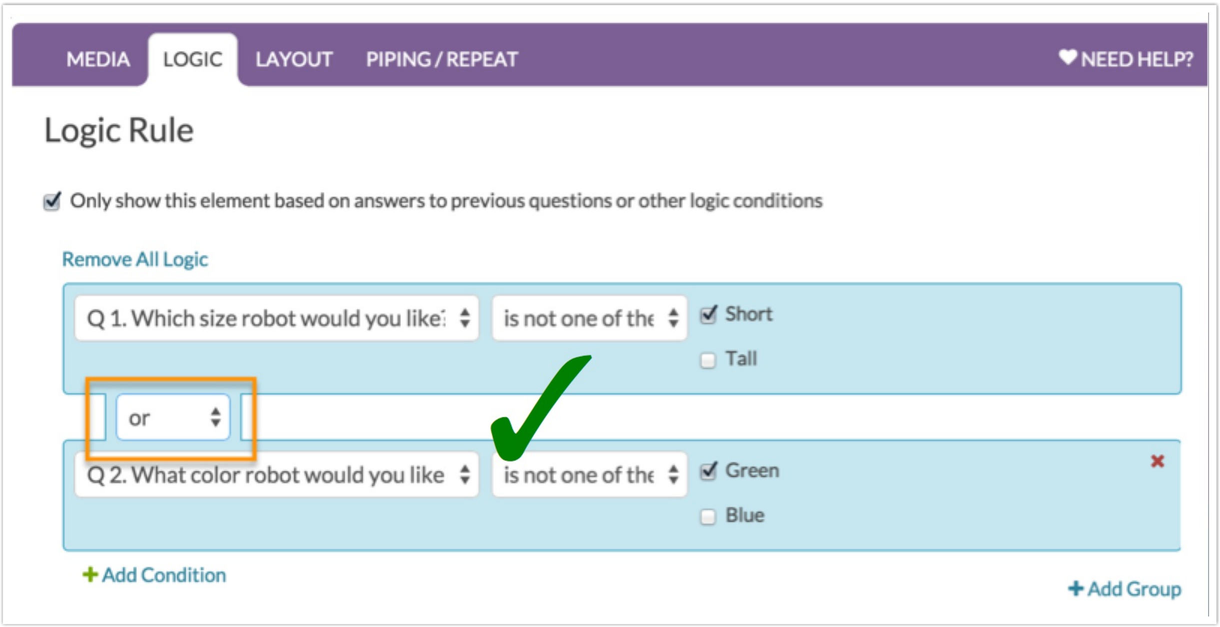
Let's apply the second logic condition we built. **AND "Which color robot would you like?" is not Green.** Because we are using the **AND** operator, both rules need to be true. So, we add a second red x to the tall x green cell. This means the coupon code will only display for tall, blue robots. This is not at all what we wanted; we just wanted to exclude short, green robots and we ended up excluding everyone but tall, blue robots.



	Green	Blue
Short	✗	✗
Tall	✗	✓

"Which size robot would you like?" is not Short AND "Which color robot would you like?" is not Green

So, let's switch the AND joining our two conditions to an OR like the support hero told us and examine why this works!



Logic Rule

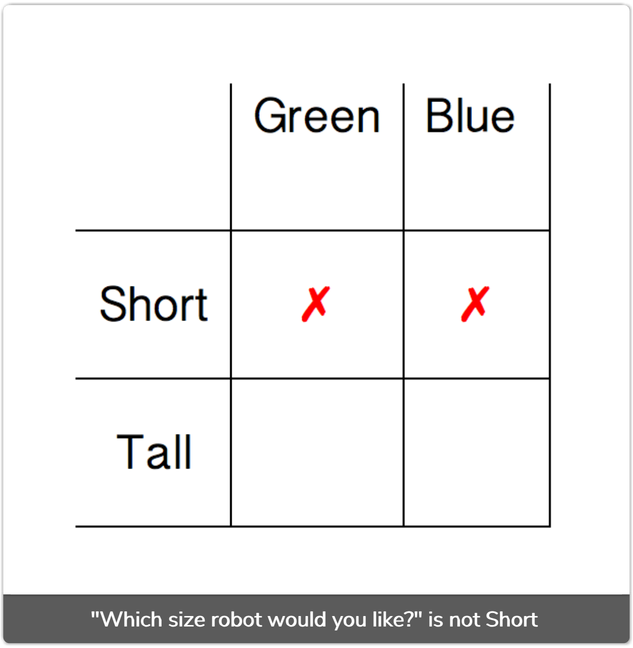
☑ Only show this element based on answers to previous questions or other logic conditions

Q 1. Which size robot would you like? | is not one of the | ☑ Short □ Tall

or

Q 2. What color robot would you like | is not one of the | ☑ Green □ Blue

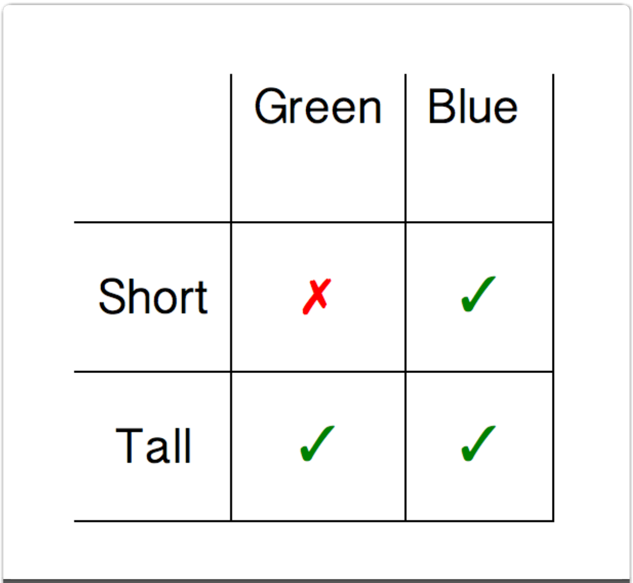
Let's apply the first logic condition we built above. **"Which size robot would you like?" is not Short.** This will then prevent the question from displaying for respondents that select short. Which is a-ok. Showing the coupon code for both green and blue tall robots is what we want.



	Green	Blue
Short	✗	✗
Tall		

"Which size robot would you like?" is not Short

Next, let's apply our second logic condition. **OR "Which color robot would you like?" is not Green .** Because both rules do not need to be true at once, we can include all not green robots allowing us to remove the red x from the blue column so we are only excluding short green robots!



	Green	Blue
Short	✗	✓
Tall	✓	✓

"Which size robot would you like?" is not Short OR "Which color robot would you like?" is not Green

Limitations

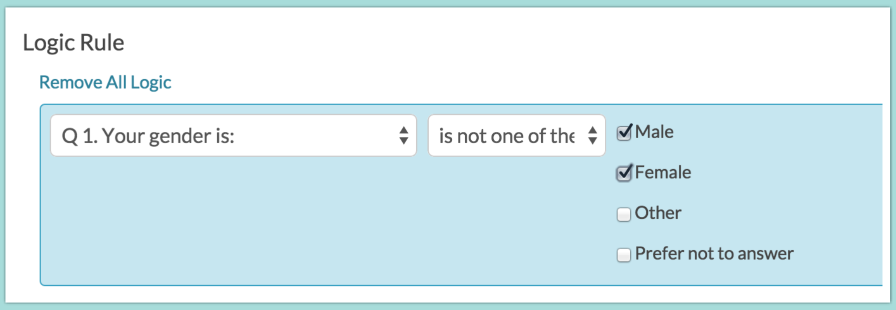
- Negative logic conditions are not available for questions on the same page. This means when using negative conditions for logic, the source question you are basing the logic on must be on a page prior to the question you are triggering.

Best Practice Tips

Because negative logic is often much more efficient, it is very tempting to use. As a general rule, we discourage the use of negative logic unless you are 100% sure you know what you are doing or are willing to *rigorously* test. Below are two best practice tips that will save you a lot of heartache when using negative logic in Alchemer.

Require Questions Used in Negative Logic!

This can be a very painful lesson to learn, particularly when using negative logic in segmented quotas. Say, for example, you set up the below logic rule.



Logic Rule

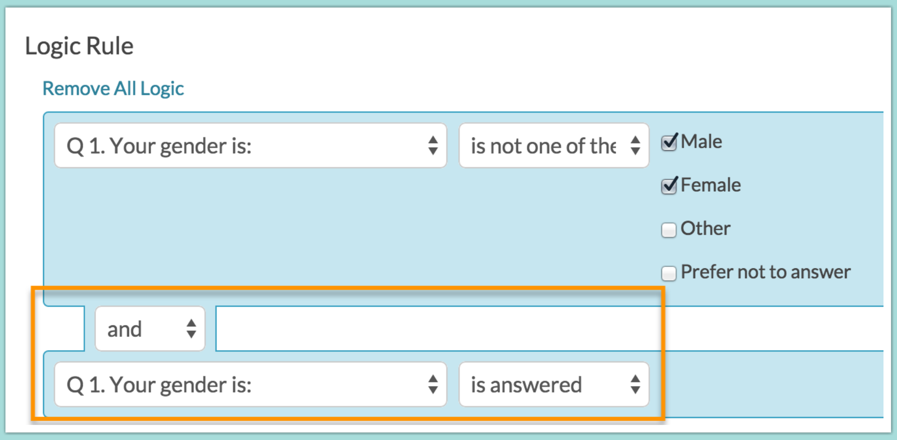
Remove All Logic

Q 1. Your gender is: | is not one of the

☑ Male
☑ Female
□ Other
□ Prefer not to answer

If the gender question is unrequired and a respondent chooses not to respond, the unanswered question will meet the above condition.

Alternatively, you can pair all logic conditions with an additional **is answered** condition to be extra careful!



Logic Rule

Remove All Logic

Q 1. Your gender is: | is not one of the | ☑ Male ☑ Female □ Other □ Prefer not to answer

and

Q 1. Your gender is: | is answered

When Using Negative Logic in Disqualify Logic or Quotas Test, Test, Test, and Test Again!

Because negative logic is so difficult to conceptualize, use extra care when applying it to your surveys. This is particularly true when you are using negative logic to disqualify respondents or set up segmented quotas, as this often translates to money!

Be sure to require questions used in your disqualify/quota logic or pair all conditions with an **is answered** condition as discussed above.

You should also test all possible variations of your logic conditions to ensure that respondents that should be disqualified are and quotas are being incremented properly.

When testing quotas we recommend creating a copy of your testing survey to reset all quotas before collecting live responses.

Related Articles