University of Colorado Logic Guide

Index:

Where in REDCap would you use logic?

Logic can be used in branching, calculated fields, conditional logic for automated survey invitations, alerts, data quality rules, the survey queue, advanced filters for reports, and project dashboards. The format for the logic will depend on where you are using it in the project, the type of field, and how the project is set up (classic, longitudinal, repeating instruments).

How do I add a variable to my logic?

You will need to add the variable name used in the project in square brackets []s (i.e. [variable]).

How to add a value to my logic?

The way you should format the value depends on if it is a 'string' or a pure number. A 'string' is any combination of text and numbers, such as a multiple-choice code in a radio or checkbox

field or an unvalidated text field. A pure number would be what you have in a field validated as number or an integer.

Any time you are referring to a 'string' you will need to wrap the value in quotes – e.g., [gender] = '1'. Anytime you are referring to a number you will leave off the quotes e.g., [weight] > 100.

You can also treat dates like numbers as statements in most functions. However, any use of a date as part of a function requires extra testing to make sure it is working as expected.

If all the options on a multiple-choice field are numbers, REDCap will treat them like numbers in its logic. However, if one or more of the options includes spaces, letters, or characters, all options will be treated like strings.

So, in:

1, Yes

0, No

99, N/A

All the code options will be treated like numbers. If the options are:

1, Yes

0, No

NA, N/A

All the code options will be treated like strings.

What operators can I use?

You may use mathematical operators (=, <, >, <=, >=, <>)

How do I combine statements in my logic?

REDCap uses Boolean logic to review statements. Boolean logic means:

AND: Includes BOTH items

OR: Includes EITHER item

For example:

If you have [variable1]= '1' AND [variable2]= '1', this will only be true if BOTH statements are true.

If you have [variable1]= '1' OR [variable2]= '1', this will be true if EITHER variable1 or variable2 equal 1.

You can combine statements with Boolean logic by using parenthesis. For example:

([variable1]= '1' OR [variable2]= '1') AND [variable3]= '2'

In this statement, variable3 must always be 2 for the statement to be true, but either variable1 or variable2 could be equal to 1.

How to indicate "no data" or "not equal to" in REDCap?

You use "" for "no data" or "null" and <> for "not equal to.'' It is important that there are no spaces in the "" statement; if there is a space, REDCap will look for a result that is just a space, not null.

For example, you may want to use branching logic so a field only shows up if the previous field has data. You could use [variable]<>'''' saying that the field does not equal no data.

You could also use this in an alert or automated invitation to send when a field has data in it.

If you are specifically looking for 0 and not null, set it as =''0'' in quotes to find the actual value of 0 and not risk pulling an empty value instead.

What is the difference between formatting logic for regular multiple-choice fields (radio, dropdown, yes/no fields) and checkbox fields?

Checkbox fields are a series of yes/no questions, with each answer choice being 1 (yes/checked) or 0 (no/unchecked).

The format for using them in logic is:

[variable_name(answer_choice)]='1' or [variable_name(answer_choice)]='0 '

For example, to reference race=4 in a checkbox field the logic would look like [race(4)] = '1'.

How can I use calculated fields to score multiple choice questions?

REDCap will use the coded value assigned to the answer choice when using radio or dropdown fields.

Checkbox fields will be scored based on if the answer choice is checked or unchecked. The format for this field type is:

[variable(answer choice)]= '1' (checked) or [variable(answer choice)]= '0' (unchecked)

Can fields from different events be used in logic (longitudinal projects)?

In longitudinal projects, if you do not tell REDCap what event the variable is in, REDCap will assume the variable is in the same event. To reference a different event, you need to prefix the event where REDCap should look in front of the variable. Event names can be found listed on the project's Define My Event's page on the right-hand side of the events table (i.e. [event_name_arm_#][variable]='1').

Can fields from different instances be used in logic (repeating instruments)?

You can tell REDCap to look at a variable in a specific instance of a repeating instrument by appending the instance number to the end of the variable. For example, [variable1][4] will look for variable1 in instance 4 of the repeating instrument.

Please note that repeating instruments are not fully compatible with Automated Survey Invitations, and you might need to check with a REDCap administrator if you are using or referencing repeating instruments with that feature.

What are smart variables and how can I use them in my logic?

Variable names always point to a data field in a project, however smart variables allow you to adapt to different situations. You may want to reference the first or previous event in a longitudinal project or the first or previous instance in a project with repeating instruments. Smart Variables are easily distinguishable from field variable names because they have dashes and colons whereas field variable names cannot (i.e. [first-event-name][variable]= '1' or [variable][first-instance]= '1').

You can find lists of smart variables anywhere you see the green "Smart Variables" button. Some places you can find this are on Project Setup --> Design your data collection instruments & enable your surveys; in the Online Designer on the form editing page; under the variable name when editing a variable; and at the bottom of the "Edit dashboard" page for project dashboards.

Can I calculate the difference between two dates in my project?

You can calculate the difference between two dates with the format:

datediff([date1],[date2],"unit")

The units are:

Unit	Definition	Additional information
"y"	years	1 year = 365.2425 days
"M"	months	1 month = 30.44 days
"d"	days	
"h"	hours	
"m"	minutes	
"s"	seconds	

Can I use "today" in my logic?

Yes, this can be used with date fields in the project e.g., - datediff("today",[visit_date],"d")

**Keep in mind that every time you save the form the calculation will run, so if you access the form a year later to make updates "today" will be updated to a year later. A work around would be to add a date field in the project and use the action tag @TODAY to capture today's date and use this date in your calculated field.

Can I use conditional logic in a calculated field?

Yes, you can use conditional logic with If/Then or If/Else statement in a calculated field. If statements have the following pattern:

if(statement = statement, value if true, value if false)

Example:

```
if([variable(1)]= '1', 1, 2)
```

If/Then or If/Else statements can also be used inside other IF/Then or IF/Else statements. You could nest them to get something like this:

```
if(statement = statement, value if true,
        if(statement = statement, value if true,
        if(statement = statement, value if true, value if all false)
        )
```

For example, if you have a question where Never is scored as 0, Rarely and Sometimes are scored as 1, and Often and Always are scored as 2, the coding might be

0, Never
1, Rarely
2, Sometimes
3, Often
4, Always

And the conditional logic for scoring would be:

```
if([variable]= '0', 0,
        if([variable]= '1', 1,
        if([variable]= '2', 1,
        if([variable]= '3', 2,
        if([variable]= '4', 2,)
        )))
)
```

You can also use conditional logic with calculated functions.

Example:

```
sum(
        if([variable1]='1', 1, ""),
        if([variable2]='1', 2, ""),
        if([variable3]='1', 3, ""),
        if([variable4]='1', 4, "")
        )
```
For each statement in the series of if statements it will provide either a value or a null depending on if the logic is true or false; and provide the sum of the results.

You can find a list of functions anywhere you see the blue "Special Functions" button. Some places you can find this are on Project Setup --> Design your data collection instruments & enable your surveys and in the Online Designer on the form editing page.

Can I use conditional logic with action tags?

You can use the @IF action tag to set these based on conditional logic in your project. The format is @IF (condition, value if true, value if false).

Example: @IF([yes_no]= '1', @HIDDEN, @HIDECHOICE= '3')

This will hide the field if the yes_no value is 1, otherwise it will hide the answer choice coded as 3.

You can also use a nesting if statements if you have multiple conditions.

Example:

@IF([location]='1',@DEFAULT="[variable1]",
@IF([location]='2',@DEFAULT="[variable2]",
@IF([location]='3',@DEFAULT="[variable3],"")))

**Keep in mind that, like all action tags, @IF will only work when a form initially loads so will not work in real time as data is entered on the page.

What is the difference between adding a series of variables and using the sum() function?

[variable1] + [variable2] will give you the sum if every answer has a value. If one of the values is blank, the entire function will not work.

If you use sum([variable1],[variable2]) and one of the values is blank, it will skip the missing value and give you the sum for all (non-blank) values.

For example, if [variable1] = 1, [variable2] = 4, and [variable3] is blank

[variable1] + [variable2] + [variable3] = Blank/Null

sum([variable1], [variable2], [variable3]) = 5

Which one you need will depend on the rules of what you are scoring.

Can a calculated field return a date instead of a number?

REDCap calculation fields can only return numeric values—they can't return words, strings, or dates unless you use the @CALCDATE or @CALCTEXT action tags. These can be added to a text field in the Action Tag/Field Annotation section.

You can perform a date calculation to add or subtract a specified amount of time from a specified date or datetime field and then provide the result as a date or datetime value using the @CALCDATE action tag - e.g., @CALCDATE([visit_date], 7, 'd'). The first parameter inside the @CALCDATE() function should be a text field with date, datetime, or datetime_seconds validation, in which you may specify (if needed) the event and repeating instance – e.g, @CALCDATE([baseline_event][visit_date], 7, 'd'). The second parameter represents the offset number amount that should be added or subtracted. The third parameter represents the unit of time using the same abbreviations as datediff calculations.

Can a calculated field return text instead of a number?

REDCap calculation fields can only return numeric values—they can't return words, strings, or dates unless you use the @CALCDATE or @CALCTEXT action tags. These can be added to a text field in the Action Tag/Field Annotation section.

You can return text instead of numerical values using the @CALCTEXT action tag. This can be added to a text field in the Action Tag/Field Annotation section.

This typically works with if/then or if/else statements – @CALCTEXT(if([gender]='1', 'male, 'female')).

You can also use special functions with the @CALCTEXT action tag:

Function	Definition	Example
left (text, number of characters)	Returns the leftmost characters from a text value	left(dessert], 3) would return 'Cup' if the value of [dessert] is 'Cupcake'
right (text, number of characters)	Returns the rightmost characters from a text value	right([last_name], 4) would return 'cake' if the value of [dessert] is 'Cupcake'
length (text)	Returns the number of characters in a text string	length([dessert]) would return '7' if the value of [dessert] is 'Cupcake'
find (needle, haystack)	Finds one text value within another; Is case insensitive	The "needle" may be one or more characters long. For example, find('p', [dessert]) would return '3' if the value of [dessert] is 'Cupcake'. The value '0' will be returned if "needle" is not found within "haystack"
mid (text, start position, number of characters)	Returns a specific number of characters from a text string starting at the position you specify. The second parameter denotes the starting position, in which the beginning of the text value would be '1'. The third parameter represents how many characters to return	For example, mid([dessert], 2, 3) would return 'upc' if the value of [dessert] is 'Cupcake'
concat (text,text,...)	Combines/concatenates the text from multiple text strings into a single text value	concat([order_type], ': ', [dessert]) would return something like 'Delivery: Cupcake' if the value of [dessert] is 'Cupcake'. Each item inside the function must be separated by commas.

		Each item might be static text (wrapped in single quotes or double quotes), a field variable, or a Smart Variable
upper (text)	Converts text to uppercase	upper([dessert]) will return 'CUPCAKE' if the value of [dessert] is 'Cupcake'
lower (text)	Converts text to lowercase	lower([dessert]) will return 'cupcake' if the value of [dessert] is 'Cupcake'
trim (text)	Removes any spaces from both the beginning and end of a text value	trim(' Sentence with spaces on end. ') will return 'Sentence with spaces on end.'

Why do I see an error when I add branching logic to my calculated field?

When you have a calculated field, it will always run the calculation when the page loads. If you have branching logic on the field so it should not show up, it creates a problem—there is data in the calculated field, but the branching logic says it should be hidden with no data. This is why you get an error. You will need to update the calculation to take into account if you want the field to be hidden. An easy way to do this is to change the calculation to incorporate the branching logic.

Example: if([variable]='1', sum(), "").

This reads, "If [variable] equals 1, do this sum() calculation. If not, leave this field null. A null value will not trigger the branching logic, so you will not get the error.

I updated the logic in my calculated field or automated survey invitation. Why don't I see this applied to the records in my project?

When you add or change a calculation, it does not automatically go through and re-save all the data in that field for you. REDCap's default is to never change your data without explicit permission. You will see how the calculation works if you look at the page, but you need to save it to update the data on the server. Any field with the red bar next to it means that there is a discrepancy between what is showing up on the webpage and what is saved on the server—usually, an unsaved calculation. To update them all at once, you can go to Data Quality under Applications on the sidebar and run Rule H. It will show all fields where the calculated values do not match what is saved and give you the option to update them all.

If an automated survey invitation has been modified after data has already been entered in the project, you may click the "Re-evaluate Auto Invitations" button in the Online Designer, which will re-evaluate selected invitations for all records to ensure that invitations get properly sent or scheduled based on the new logic.