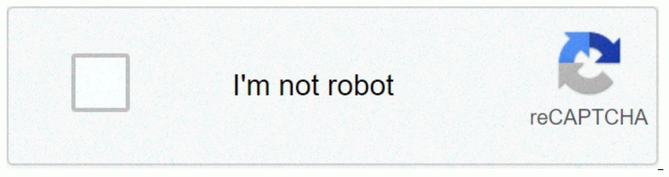

Continue

Google sheet function

The syntax is 'yes' (test, then_true, otherwise_value). There are three arguments in Google Sheets If: Test, Then_true and Otherwise-Value.In Google Sheets the If () entered by typing into a cell; a suggestion box seems to help. As with Excel's If () feature, the If () feature in Google Sheets makes it easy to make decisions that plug into a worksheet. Here's how to use google Sheets If ( ). The If () function tests whether a certain condition in a cell is true or false. If the condition is true, the function will perform an operation. If the condition is wrong, the function will perform a different operation. The initial true or false test, as well as the follow-up operations, are defined with the arguments of the function. Nest If statements () to test multiple conditions and perform different operations based on test results. The syntax of a function refers to the format in which the function should be indicated. It includes the name of the function, hooks, comma separators and arguments. The syntax for the If function ( ) is: 'if('if' (test, then_true, The three arguments of the function are: Test: a value or expression that is tested to see if it is true or falseThen_true: the operation that is performed if the test is trueOtherwise_value: the operation that is performed if the test is wrong The argument otherwise_value is optional , but you need to specify the first two arguments for the function to work properly. In row 3, the If() function returns various results such as: 'i This example: Tests to see if the value of the A2 cell is equal to 200 (the test argument)If this is the case, the function displays value 1 in cell B3 (the then_true argument)If A1 does not equate to 200, the function displays value 2 in cell B3 (the optional otherwise_value argument) if you refuse to enter into an argument of otherwise_value , Google Sheets will return the wrong logical value. Unlike Excel, Google Sheets does not use dialog boxes for function arguments. Instead, it has a self-suggesting box that appears that you type the name of the function into a cell. To enter the function: Click on the B3 cell to make it the active cell. Type in the equal sign followed by the name of the function though. When you type, the auto-suggest box appears with the names of functions that start with the letter I. When IF appears in the box, click on it to enter the function name and open the bracket or round holder in the B3 cell. Click on cell A2 in the worksheet to enter this cellular reference. After the cell reference, type in the equal symbol, followed by the number 200. Enter a comma to complete the test argument. Type 2 followed by a comma to enter this number as an argument Special. Type 1 to enter this number as a special otherwise_value argument. Don't get into a comma. Press Enter to insert a closing bracket) and complete the function. Value 1 should appear in cell B3, since the value in A2 is not equal to 200. If you click on cell B3, the appears in the formula bar above the worksheet. You're probably familiar with the basics of Google Sheets, but Google's spreadsheet offering has loads of features that aren't obvious at first glance. Here are some of our favorites. Of course, you probably already know some basic formulas, like SUM and AVERAGE. And it's likely you've gotten to know the toolbar pretty well, but it's pretty amazing how good it's going. I love spreadsheets, but even to this day, I'm still discovering new stuff in Google Sheets. Import Data Tables This sounds super boring, but it's actually really neat. If a website has a table or list of information that you want to keep track of, you can use The ImportHTML feature to basically scrape that data and paste it into a spreadsheet. From there, the data is updated automatically each time you open the spreadsheet (if changes to the original table have been made, of course). The feature would look like this: 'ImportHTML' (' URL', 'table', 'table', '0' is the web page where the data is located, 'table' is how the data is displayed on the web page (you can also use 'list' if it's a list),' and '0' represents which table you want to import if there are multiple tables on the web page (0 is the first , 1 is the second, and so on). An example of this would be to keep track of sports statistics for a fantastic league. You can import various statistics from a site like Baseball Reference into a spreadsheet. Of course, you can simply bookmark the site, but with ImportHTML, you can customize things like exactly what statistics appear (by adding Col1, Col4, etc. after the 0), as well as go and get data from other tables on a different web page and have everything appear in a single spreadsheet. RELATED: 10 tips and tricks for google docs reference data from other spreadsheets If you have multiple spreadsheets (or several sheets in a spreadsheet) that all relate to each other in some way, you may find yourself coming and going between them often. There's a way to make this a little easier. You can reference the cells of other sheets (or another spreadsheet entirely). For example, say that keep records of everything you spend on groceries in a sheet and that sheet also contains a total amount spent for the month. And, let's say you have another sheet that gives you a summary of what you spend each month on various categories. In your summary sheet, you can refer to this grocery sheet and the specific cell contains the total. Each time you update the original sheet, the value of the summary sheet automatically updates. The function would look like this: 'sheet1!' B5 Sheet1 would be the name of the sheet with the data you want to reference, and B5 is the cell you want to reference. The exclamation mark between the two. If you want to reference the data of an entirely different spreadsheet, you would use the IMPORTRANGE function, such as: 'IMPORTRANGE' (URL, 'sheet1! B5) The URL is the link to the other the other This connects the cell of this spreadsheet to the cell in which you enter the above formula. Each time the cell is updated with a different value, the other cell updates with it. As the function name suggests, you can also reference a range of cells, such as B5:C10. Conditional Fitness This feature is a little more well known than some of the others I mentioned, but I feel like it's still not as popular as it should be. Conditional formatting allows you to change the appearance of a cell based on the data in the cell. You can access the feature by clicking Format in the toolbar, and then selecting the Conditional Format command. In the shutter that opens on the right, you can set up your settings. For example, you can make a cell (or cells) green if the number they contain is greater than zero. There is also the IF function, which is not technically a part of the conditional fitness function, but it can take it to the next level in a way. This allows you to do things like add some value to a separate cell whenever the value in the active cell is a particular number: 'IF(B4'63350)So in this example, if the value of the B4 cell is 63 or more, you can automatically make the value of the current cell 35. And then show a 0 if not. Of course, this is just one example, because there is much more you can do with it. RELATED: How to highlight a line in Excel by using conditional formatting integration spreadsheets on a website If you've created a calendar or list in Google sheets that you want to share with others, you can simply share the actual document with them by sending them an email invitation to see it. However, if you need to supplement it with other information that is on your blog or website, you can actually integrate spreadsheets on web pages. All you need to do is browse to file publish on the web. From there, click on the Integrate tab and then choose to publish the entire spreadsheet or just a specific sheet. After that, just copy and paste the iFrame code into your web page. Play with scripts For anything That Google Sheets can't do out of the box, there's usually a Google Apps script that you can use next to your spreadsheet to make just about everything happen. We've already talked about Google Apps scripts, and you can do a lot with this kind of feature. You explore the additional modules available by going to Tools and Add-Ons, or you can write your own scripts in Script editor in the Tools menu. For example, I have a custom script that allows me to press a single button to instantly add values specific to existing values in a handful of cells. You can't do it with Google Sheets out of the box, so having the script editor here gives Google Sheets a good dose of steroids. A function is a pre-defined formula in Excel and Google Sheets that is intended to perform specific calculations in the cell in which it is located. The information in this article applies to Excel 2019, Excel 2016, Excel 2013 and Google Sheets. One Syntax refers to the layout of the function and includes the name of the function, hooks, comma separators, and arguments. Like all formulas, functions begin with the equal sign followed by the name of the function and its arguments: The name of the function tells Excel which calculations to perform. Arguments are contained within the brackets or round brackets and tell the function what data to use in these calculations. For example, one of the most commonly used features in Excel and Google Sheets is sum: SUM (D1: D6) In this example: The name tells Excel to add up the data in the selected cells. The argument function (D1:D6) adds the content of the D1 cellular range to D6. The usefulness of Excel's built-in functions can be expanded by nesting one or more functions within another function in a formula. The effect of nesting functions is to allow multiple calculations to take place in a single worksheet cell. To do this, the nested function acts as one of the arguments for the main or external function. For example, in the following formula, the SUM function is nested inside the ROUND function. ROUND (When evaluating nested functions, Excel first performs the deepest or most intimate function, then works its way outwards. As a result, the above formula will now be: Find the sum of the values in the D1 to D6 cells. Round this result to two decimals. Since Excel 2007, up to 64 levels of nested functions have been allowed. In earlier versions, seven levels of nested functions were allowed. There are two categories of functions in Excel and Google Sheets:custom worksheet functions or user-defined functions are integrated into the program, such as the SUM and ROUND functions mentioned above. Custom functions, on the other hand, are functions written, or defined, by the user. In Excel, custom features are written in the built-in programming language: Visual Basic for Applications or VBA for short. The features are created using the Visual Basic editor, which is installed with Excel. Google Sheets custom features are written in Apps Script, a form of JavaScript, and are created using the script editor under the Tools menu. Custom functions generally, but not always, accept some form of data input and return a result to the cell where it is located. Here's an example of a user-defined feature that calculates the buyer's discounts written in the VBA code. The functions defined by the original user, or UDFs, are published on Microsoft's website: Function reduction (quantity, price)If the amount Price - 0.1ElseDiscount - 0End IfDiscount - Application.Round (Discount, 2)End Function In Excel, user-defined functions cannot return values to the cell in which they are located. They cannot execute commands that change Excel's operating environment, such as changing content or formatting a cell. Microsoft's knowledge base lists the following limitations for user-defined functions: insertion, deletion, deletion, fitness cells in a worksheet. Changing the value of the data in another cell. Moving, changing names, deleting or adding sheets to a workbook. Changing environment options, such as computational mode or screen views. Adjusting properties or executing most methods. Although Google Sheets doesn't currently support them, in Excel, macros are a series of recorded steps that automate repetitive tasks on the worksheet. Tasks that can be automated include formatting data or copying and pasting. Although both use Microsoft's VBA programming language, they are different in two ways: UDFs perform calculations, while macros perform actions. As mentioned above, UDFs cannot perform operations that affect the program environment, while macros can do so. In the Visual Basic editor's window, the two can be differentiated because: UDFs start with a function statement and end with End Function.Macros starting with a Sub statement and ending with End Sub. Sub.