

Second Edition

Spice Up Your Sheet Life

100 tips & tricks for Google Sheets



by Google Developer Expert Ben Collins



Contents

Contents

Hello!

[#1. Keyboard Shortcuts in Google Sheets](#)

[#2. Notifications in Google Sheets](#)

[#3. Datetime Tip in Google Sheets](#)

[#4. Super Cool \(And Clever\) Way to Combine Text and Numbers in a Cell](#)

[#5. Replace Numerical IF Functions With MIN or MAX](#)

[#6. Quick Ways to Copy Data Down a Column](#)

[#7. Multiply an Entire Column With a Single MMULT Formula](#)

[#8. The OFFSET Function](#)

[#9. The ISEMAIL Function for Better Data Entry](#)

[#10. The CHAR Function to Insert Special Characters, Images and Emojis](#)

[#11. Group Rows or Columns so You Can Toggle Between Showing or Hiding Them](#)

[#12. Use the F4 Key to Quickly Repeat Your Previous Action in a New Cell](#)

[#13. Tag People in Comments in Google Sheets](#)

[#14. Conditional Formatting Across Entire Row](#)

[#15. Trick to Round Numbers to the Nearest Ten, Hundred, etc.](#)

[#16. Speed Up Your Data Entry](#)

[#17. Generate Random Letters With the CODE Function](#)

[#18. Create Heatmaps With Your Data](#)

[#19. Quickly Copy Pivot Tables](#)

[#20. Rename Pivot Table Column Headings](#)

[#21. Hyperlink to Different Parts of Your Sheet](#)

[#22. Repeat Number Groups With the ROW & MOD Functions](#)

[#23. Create Google Sheets Instantly](#)

[#24. Advanced Formula Workout Challenge](#)

[#25. Neat Trick With Freeze Panes](#)

[#26. Protect Individual Sheets](#)

[#27. Creating a New Sheet in the Folder You're In](#)

[#28. Automate your Sheets with Macros](#)

[#29. 5 Million Cells](#)

[#30. Collaboration & Filters](#)

[#31. Create Pictures With the CHAR Function](#)

[#32. Use the YEARFRAC Function to Calculate the % of the Year Left](#)

[#33. Format Individual Data Points In Charts](#)

[#34. Quickly Summarize Your Data](#)

[#35. How To Remove Duplicates](#)

[#36. Numbers Starting With Zero](#)

[#37. Vlookup Formula Challenge](#)

[#38. Text Trickery Part I](#)

[#39. Text Trickery Part II](#)

[#40. Text Trickery Part I](#)

[Formula Challenge #1](#)

[#41. ROW/COLUMN Functions](#)

[Formula Challenge #1 Solution](#)

[#42. The Endless Possibilities of Custom Functions](#)

[#43. The SUBTOTAL Function](#)

[#44. Year In Progress Sparkline](#)

[#45. Find the Most Frequent Word](#)

[#46. Google Sheets Features Roadmap](#)

[#47. Macro to Show/Hide Tabs](#)

[#48. Night Mode, Anyone?](#)

[#49. Create New Tabs Instantly](#)

[#50. Images in Cells](#)

[#51. Remove Duplicates](#)

[#52. Show Formulas](#)

[Formula Challenge #2](#)

[#53. Use the SPLIT Function to Separate Text Into an Array Output of Constituent Parts](#)

[Formula Challenge #2 Solution](#)

- [#54. Colorful Sheets!](#)
- [#55. Use the MID Function to Return a Segment of a String](#)
- [#56. Quick Entry & Exit With Formulas](#)
- [#57. Discover New Formulas Today](#)
- [#58. Make Conditional Formatting Permanent](#)
- [#59. Use Custom Number Formatting to Pluralize Data](#)
- [#60. Be Creative With Your Google Sheets!](#)
- [#61. Date Validation](#)
- [#62. Add Star Rating System](#)
- [#63. Create QR Codes With Formulas](#)
- [#64. Apps Script Button](#)
- [#65. QA Checklists](#)
- [#66. Use the Parser Functions to Format Array Data](#)
- [#67. Manage Apps Script Projects From the Dashboard](#)
- [#68. Brilliant Apps Script Keyboard Shortcuts](#)
- [#69. Video Sidebar With Apps Script](#)
- [#70. Extract Hyperlink URLs With a Formula](#)
- [#71. Annotate Line Charts](#)
- [#72. Build Numbered Lists With the SEQUENCE Function](#)
- [#73. Double Click Dates to Use the Date Picker](#)
- [#74. Use the RANK Function to Find the Position of a Value in a Dataset](#)
- [#75. A Fun Use of the FACT Function](#)
- [#76. Control Who Can Edit Tabs](#)
- [#77. Custom Formulas Data Validation](#)
- [#78. Line Chart Trick](#)
- [#79. Find Every Formula](#)
- [#80. Simple Formula Clock](#)
- [#81. Hyperlinked Image](#)
- [#82. Dates Are Really Just Numbers](#)
- [#83. Working With Excel Files](#)
- [#84. Colored Checkboxes](#)

[#85. TRANSPOSE Function: Part I](#)

[Formula Challenge #3](#)

[#86. TRANSPOSE Function: Part II](#)

[Formula Challenge #3 Solution](#)

[#87. Create Data With Randarray](#)

[#88. SWITCH Function](#)

[#89. Upgrade Your Apps Script!](#)

[#90. Rolling Average Formula](#)

[#91. Conditional Sparklines](#)

[#92. Trace Precedents With F2](#)

[#93. Group Columns Trick](#)

[#94. Organize Tabs](#)

[#95. Pivot Date Trick](#)

[#96. Pivot Custom Text Groups](#)

[#97. Pivot Table Bins](#)

[#98. Use the Combo chart to create bullet charts](#)

[#99. Use Array Formulas to check whether cells are blank](#)

[#100. Use the Hyperlink function to create a link to the next blank row](#)

[Closing Comments](#)

[Contact Details](#)

Hello!

Did you know that you can build a language translation tool in a Google sheet?

Or create a chart of stock prices in under 30 seconds?

How about writing a formula which inserts images or logos into your spreadsheet? (Try typing =CHAR(128202) into your worksheet.)

Check, check and check. All possible and covered in this book.

These tips come from my [Google Sheets Tips email series](#), a weekly series sharing a single, actionable Google Sheets or Apps Script tip via email every Monday.

Think of these tips as your Monday morning espresso, in spreadsheet form.

Each week features a single bite-size idea you can apply in your work immediately.

This book contains the first 100 tips, including tips that are practical and esoteric, easy and difficult, short and long, and everything in between.

You may have seen some of them before but my hope is that you'll find some magic in these pages, to inspire you or show you new ways to use spreadsheets you didn't know were possible.

Feedback is welcomed! If you find any errors or feel anything merits further clarification, please don't hesitate to contact me.

Likewise, if you have your own favorite, unusual or neat way of using spreadsheets, and would like to share, then I'd love to hear from you. I'll credit anyone who contributes.

Happy filtering, sorting, summing, pivoting, charting and coding!

Ben Collins
May 2020

#1. Keyboard Shortcuts in Google Sheets

Let's kick off this series with one of the all-time top Google Sheet productivity tips.

Here are my top 5 keyboard shortcuts:

1. Clear All Formatting in a cell or range

Mac: ⌘ + \

PC/Chromebook: Ctrl + \

2. Insert the current date in a cell

Mac: ⌘ + ;

PC/Chromebook: Ctrl + ;

3. Select all the data in a table

Mac: ⌘ + A

PC/Chromebook: Ctrl + A

4. Find and Replace

Mac: ⌘ + Shift + H

PC/Chromebook: Ctrl + H

5. Open the drop-down menu on filtered cell

Mac: Ctrl + ⌘ + R

PC/Chromebook: Ctrl + Alt + R

Today, I challenge you to use keyboard shortcuts.

It might feel clumsy at first, but persevere and it'll pay off in spades as you become more efficient in your work.

To see all of the available shortcuts, go to **Help > Keyboard shortcuts**

#2. Notifications in Google Sheets

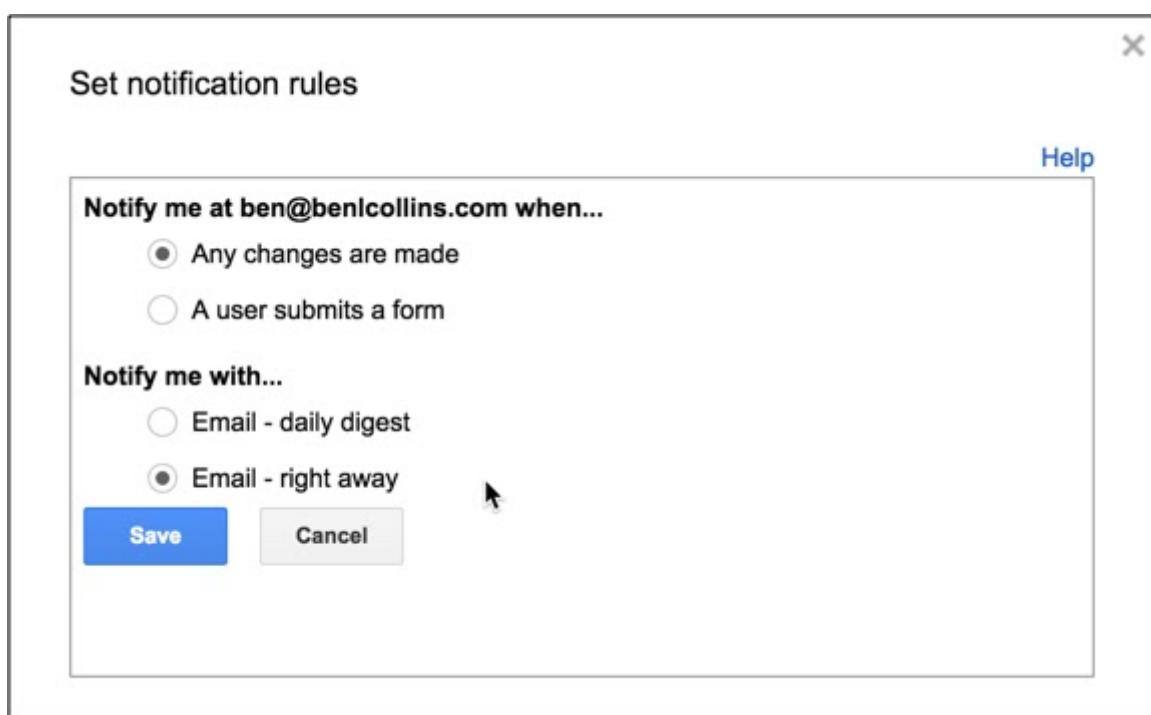
Do you collaborate with colleagues in Google Sheets?

Here's how you can stay informed when any changes are made to your Sheets, for example when new data is added.

Stay informed of changes to a Google Sheet by setting Notification Rules to send you an email.

You'll find it under this menu: **Tools > Notification rules...**

Here you can decide what should trigger a notification and how they should be delivered (immediately or a daily summary):



Never miss an update to your Google Sheets again!

#3. Datetime Tip in Google Sheets

Did you know that dates and times in your spreadsheet are stored as numbers, like 43,241.5866? Plain old numbers!

(That's 5/21/2018 14:04:43 by the way. Want to see for yourself? Try entering a date and formatting it as a number.)

This means we can use the same formulas on dates that we use for Numbers.

Say you have a datetime in cell A1.

If you enter `=INT(A1)` into cell B1, you'll get the date with the time piece removed. That's because the decimal part of the number represents the time, and the function INT returns just the integer part (the whole number part) of a number:

	<i>fx</i>	<code>=int(A1)</code>
	A	B
1	5/21/2018 14:04:43	5/21/2018 0:00:00
2		
3		

If you enter `=MOD(A1, 1)` into cell B1, you'll get the time on its own without the date. The MOD function does Modulo arithmetic and just returns the remainder (portion after the decimal point), which in this case is the time:

	<i>fx</i>	<code>=mod(A1, 1)</code>
	A	B
1	5/10/2018 9:52:40	9:52:40
2		
3		

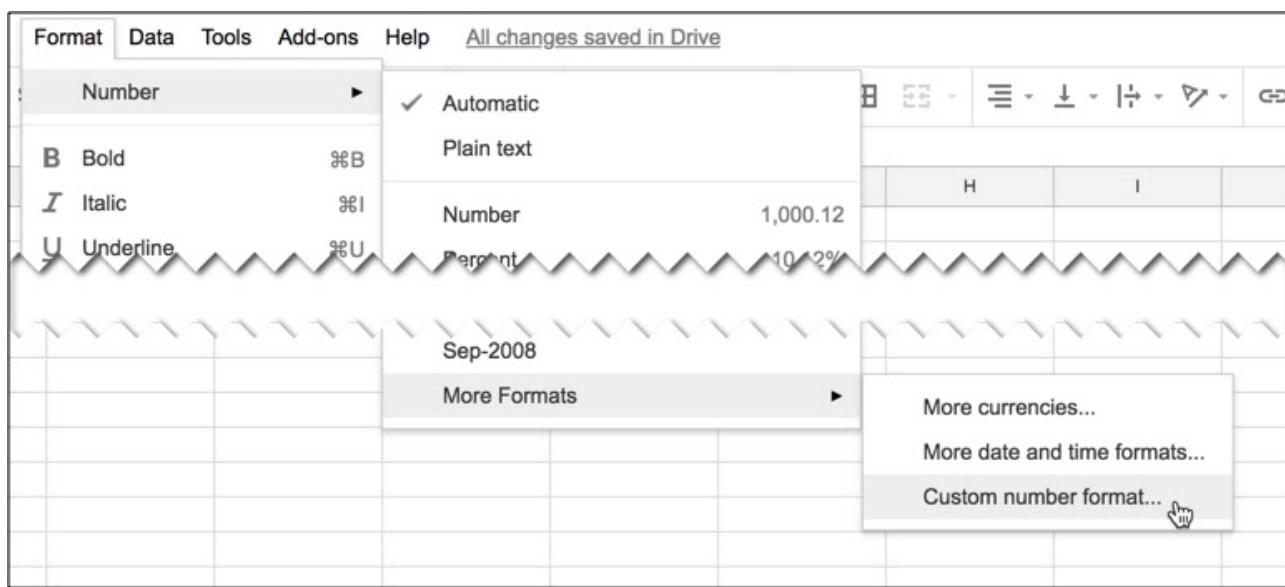
Understanding that dates and times are just numbers will really help you to work with them effectively in your Sheets!

#4. Super Cool (And Clever) Way to Combine Text and Numbers in a Cell

Oftentimes when you reference a number repeatedly you put it in a separate cell, so you can link to it in your formulas. Perhaps it's a tax rate or threshold number for a set of calculations.

Well, here's a cool tip to enhance these reference cells: you can use custom formatting to add a text label to a number in a cell, but still use it as a number.

Highlight your reference cell and navigate to **Custom number format...** in the menu:



In the custom format option box, add your label inside quotations, followed by a 0 or 0.0 or 0.00 to specify the number and decimal places if required. For example:

"Threshold: "0.00

as shown here:

The screenshot shows a Google Sheets interface with a 'Custom number formats' dialog box open. The dialog has a title 'Custom number formats' and a preview area showing 'Threshold: "0.00"' next to 'Sample: Threshold: 1234.56'. There is an 'Apply' button and a 'Help' link. The main sheet shows a table with columns A, B, C, and D, and rows 1 through 10. Cell A2 contains the value '10'.

When you click Apply, the cell will now have your text label showing in front of the number.

You can use that cell in calculations as if it was just written as a plain number, because it's still a number datatype under the hood.

The screenshot shows a Google Sheets interface with a formula bar containing '=A2*20'. Cell A2 contains the text 'Threshold: 10.00', which is highlighted with a dashed orange border. A tooltip above cell A2 shows '200 ×'. Cell C2 contains the formula '=A2*20'.

#5. Replace Numerical IF Functions With MIN or MAX

Don't write unnecessarily verbose IF statements that could easily be replaced with a MIN or MAX function. This tip is all about writing formulas efficiently. Here's how to replace numerical IF formulas with MIN or MAX functions.

Suppose you have a column of values that you want to cap at a certain level, so that everything above a threshold value, e.g. 200, gets set to that value.

Most of us would approach this by writing an IF formula that checks whether the value is above 200 and then set it to 200 if TRUE, or the actual value if FALSE, like so:

	A	B	C
1	Value	Above threshold of 200	
2	22?	=min(A2, 200)	
3	173	173	
4	184	184	
5	e.g. 155	155	
6	226	200	
7	26	26	

However, you can replace the whole IF function with a much more succinct MIN function, which chooses the value 200 if the actual value is larger, since 200 is the minimum:

=MIN(A2, 200)

	A	B	C
1	Value	Above threshold of 200	
2	22	<code>=min(A2, 200)</code>	
3	173	173	
4	184	184	
5	155	155	
6	226	200	
7	26	26	

Similarly, we can use the MAX function to replace IF statements when we're looking at a threshold on the low side.

For example, suppose you're not interested in any small values in your dataset, and you want to write a formula to replace them all, e.g. with 10. This time we can use the MAX function:

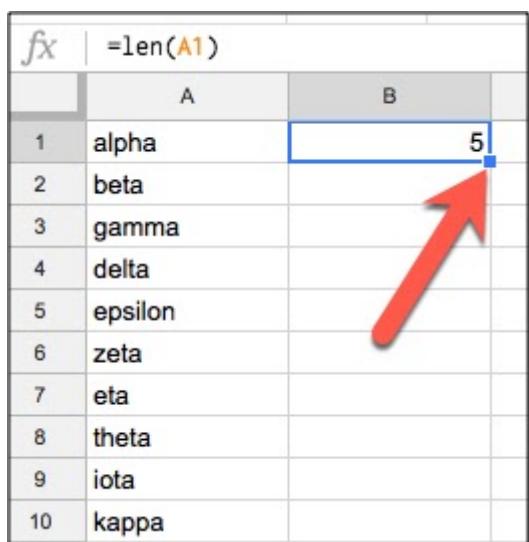
`=MAX(A2, 10)`

It's good practice to write efficient formulas because it's quicker and you're less likely to make mistakes.

#6. Quick Ways to Copy Data Down a Column

Follow this tip to copy data down columns or across ranges more efficiently: you can quickly fill down columns by double-clicking the blue button or using Ctrl + D (Cmd for Mac) or Ctrl + Enter (Cmd for Mac).

Suppose you have a column of values with a formula or value in the top cell of the adjacent column, as shown in the following image:



	A	B
1	alpha	5
2	beta	
3	gamma	
4	delta	
5	epsilon	
6	zeta	
7	eta	
8	theta	
9	iota	
10	kappa	

To copy the formula quickly down the column, **double-click the blue mark** in the corner of the highlighted cell, shown by the red arrow. This will copy the cell contents and format down as far as the contiguous range in preceding column (column A in this case).

An alternative way to quickly fill in a column is to highlight the range you want to fill, including the first value that you want to copy down, as shown in the following image:

	A	B	C
1	alpha	5	
2	beta		
3	gamma		
4	delta		
5	epsilon		
6	zeta		
7	eta		
8	theta		
9	iota		
10	kappa		
11			

Then press **Ctrl + D (PC and Chromebook)** or **Cmd + D (Mac)** to copy the contents and format down the whole range, like so:

	A	B	C
1	alpha	5	
2	beta	4	
3	gamma	5	
4	delta	5	
5	epsilon	7	
6	zeta	4	
7	eta	3	
8	theta	5	
9	iota	4	
10	kappa	5	
11			

You can also do this with **Ctrl + Enter (PC and Chromebook)** or **Cmd + Enter (Mac)**, which will fill down the column. This shortcut also has the nice property that if you highlight a range of cells, like this:

	A	B	C	D
1	0.2631152258			
2				
3				
4				
5				

It will fill out the whole range, down the column and across the rows:

	A	B	C	D
1	0.7457901683	0.561209344	0.8996954393	
2	0.3973999779	0.9368691901	0.02514129582	
3	0.4690810386	0.2985405833	0.5349196826	
4				
5				

Note, that with all these cases, the upper left cell is the one that will be copied down (and across) your range.

#7. Multiply an Entire Column With a Single MMULT Formula

Follow this tip to quickly transform a column of numbers, for example changing the sign, scaling up or scaling down. Here's how to multiply or divide value columns with a single MMULT formula or ArrayFormula.

Suppose you have a column of values in column A, and you want to negate all the numbers. I recently needed to do this when I wanted to display a chart series as negative values.

To quickly transform an entire column of numbers to negatives (or positives if values are already negative), use a single MMULT formula at the top of the column, like so:

<i>fx</i>	<code>=MMULT(A2:A1000 , -1)</code>	
	A	B
1	Values	Negative values
2	61	=MMULT(A2:A1000 , -1)
3	28	-28
4	34	-34
5	32	-32
6	60	-60
7	36	-36
8	53	-53
9	9	-9
10	92	-92

To scale a number up by a constant factor, simply replace the -1 above with the new constant, for example 10 as shown in this image:

fx | =MMULT(A2:A1000 , 10)

	A	B	
1	Values	Multiply by 10	
2	61	=MMULT(A2:A1000 , 10)	
3	28	280	
4	34	340	
5	32	320	
6	60	600	
7	36	360	
8	53	530	
9	9	90	
10	92	920	

And if you want to divide your values, you can use MMULT with a decimal to achieve the division, for example to divide by 10 use 0.1:

fx | =MMULT(A2:A1000 , 0.1)

	A	B	
1	Values	Divide by 10	
2	61	=MMULT(A2:A1000 , 0.1)	
3	28	2.8	
4	34	3.4	
5	32	3.2	
6	60	6	
7	36	3.6	
8	53	5.3	
9	9	0.9	
10	92	9.2	

Pretty cool huh?

MMULT multiplies matrices together and this works because the number of columns of the first matrix, A2:A1000, is 1, which equals the number of rows of the second matrix, a single 1 by 1 matrix (i.e. 1 row).

Another way to achieve this calculation is to use an Array Formula, for example:

```
=ArrayFormula( A2:A1000 * -1 )
```

#8. The OFFSET Function

Here's how to move ranges of data with the OFFSET function.

The OFFSET function returns a reference to a range that is offset from a starting point in a worksheet.

It's not a function you need particularly often, but it's worth knowing about because it allows you to move ranges around very easily.

For example, look at the following table where the columns of data are not lined up in the first row:

	A	B	C	
1	Column 1	Column 2	Column 3	
2		1		
3		2	6	
4		3	7	10
5		4	8	11
6		5	9	12
7				
8				

You want to align these columns at the top of the column, as shown in the following image:

Column 1	Column 2	Column 3
1	6	10
2	7	11
3	8	12
4	9	
5		

The OFFSET function can do this with a single formula for each column, which is quicker and easier than copy-pasting or creating complex nested formulas.

The OFFSET formula in this example is:

```
=offset( B2:B6, 1, 0, count(B2:B6) )
```

The first argument, **B2:B6**, is the original range of data we want to move. This is what the formula starts with:

	A	B	C	
1	Column 1	Column 2	Column 3	
2		1		
3		2	6	
4		3	7	10
5		4	8	11
6		5	9	12
7				

The second argument, **1**, is the number of rows we want to offset. In this case it's 1 because we want to move down one row to grab the first value, like this:

	A	B	C	
1	Column 1	Column 2	Column 3	
2		1		
3		2	6	
4		3	7	10
5		4	8	11
6		5	9	12
7				
8				

The third argument, **0**, is the number of columns we want to offset. In this case we don't have a column offset so we set this to 0.

The final argument is an optional value for the height of the range to return. In this case I've set it to **count(B2:B6)** because I only want to return the four values and don't need to include the 5th empty cell, so the formula is essentially grabbing this range:

	A	B	C	
1	Column 1	Column 2	Column 3	
2		1		
3		2	6	
4		3	7	10
5		4	8	11
6		5	9	12
7				
8				

Finally, this new range is output in cell F2, or wherever I've typed the OFFSET function:

	A	B	C	D	E	F	G	H
1	Column 1	Column 2	Column 3		Column 1	Column 2	Column 3	
2		1				1	=offset(B2:B6, 1, 0, count(B2:B6))	
3		2	6			2	7	11
4		3	7	10		3	8	12
5		4	8	11		4	9	
6		5	9	12		5		
7								
8								
9								
10								

For the data in column C, I need to move that up by two rows, so I need to change my OFFSET formula slightly to this, which I typed in cell G2 in my example:

```
=offset( C2:C6, 2, 0, count(C2:C6) )
```

#9. The ISEMAIL Function for Better Data Entry

Here's a simple trick to force users to enter valid email addresses in cells: use the ISEMAIL function with data validation to ensure only valid email addresses are accepted.

The ISEMAIL function in Google Sheets gives you the ability to check whether an email is valid or not, which is particularly useful if you work with contact or customer data in Sheets.

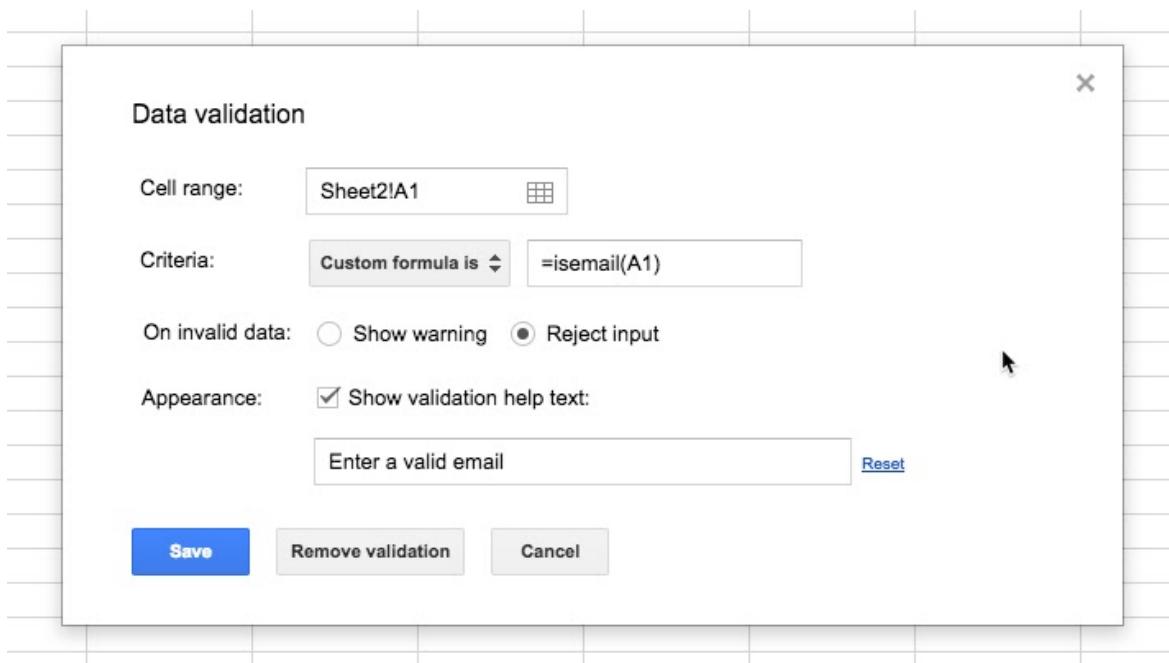
It's a super easy formula and works like this. Here's an example of a valid email:

	A	B	C
1	ben@benlcollins.com	TRUE	
2			

Combine it with data validation and you can ensure only valid email addresses are entered into your Sheets, ensuring better data accuracy going forward.

In the cell where an email address will be entered, for example A1 in the image above, right click and choose **Data Validation...** or go to **Data > Data Validation...** menu option, which opens this popup.

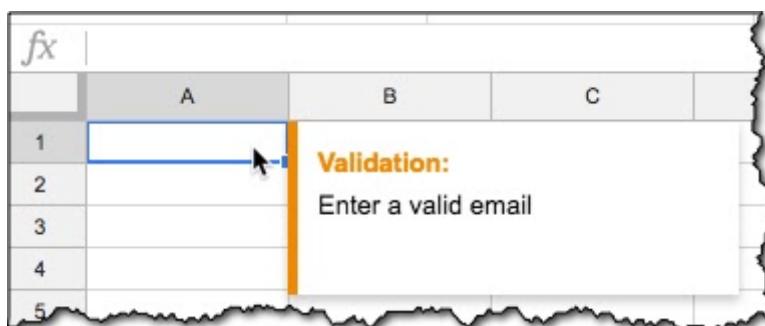
Choose **Custom Formula** in the second option and enter `=isemail(A1)`, as shown in this image:



I've chosen to reject the input if the formula evaluates to FALSE (because of an invalid email address), which shows this warning and leaves the cell empty:



The data validation in the cell also shows a hint if the user hovers over this cell, in this case prompting the user to enter a valid email address. This is customizable in the data validation popup builder.



Sweet!

Data validation is a useful technique for ensuring that your data entry is clean, correct and secure.

#10. The CHAR Function to Insert Special Characters, Images and Emojis

Have you heard of the CHAR function? It's a little gem! You can use this function to insert special characters, images or even emojis into cells.

The CHAR function in Google Sheets is a nifty little function that converts a number into a character according to the current Unicode table.

It's super easy to use.

Here's an example:

fx	= "Cohort "&char(8594)&char(10)&" Months after signup "&char(8595)	A	B	C	D
89					
90	Cohort → Months after signup ↓	Dec-2016	Jan-2017	Feb-2017	
91	0	5	16	24	
92	1	3	10	19	
93	2	2	7	15	

I use three different CHAR functions to add context and force a line break.

- CHAR(8594) produces the right arrow.
- CHAR(10) produces a carriage return (new line).
- CHAR(8595) produces a down arrow.

It makes the table much easier to interpret and reduces the chance of a user misreading the data.

Here's a few more CHAR function outputs you can try on for size:

	A	B
1	169	©
2	8482	™
3	8592	←
4	8593	↑
5	8594	→
6	8595	↓
7	8596	↔
8	8597	↕
9	9758	👉
10	9760	☠
11	9786	😊
12	9789	🌙
13	9928	🌧
14	128077	👍

There are thousands more CHAR characters to explore, so I encourage you to go and experiment.

I've found this tool, [Graphemica](#), to be the most useful for finding them.

Search for a character, and then use the number from the **HTML Entity (Decimal)** metadata section in your CHAR function.

#11. Group Rows or Columns so You Can Toggle Between Showing or Hiding Them

If you highlight several rows (or columns) by clicking on their numbers (or letters), you can then right click and choose to Group rows (or columns) which adds a button above the rows (or columns), which you can toggle to show/hide these grouped rows (or columns).

Here's an example:

D	E	F	G	H	
	In group	In group	In group	In group	

#12. Use the F4 Key to Quickly Repeat Your Previous Action in a New Cell

For example, say you've just given a cell a red background and you want to apply that again to another cell or range. Simply highlight the new cell or range and hit F4 before doing anything else, to change the background to red. Voila!

(Note, this will also copy-paste static values, but not formulas. If you type some text into a cell and then hit F4 in a different cell before doing anything else, it'll copy that same text to the new cell. This quirk only works cell to cell, not cell to range though.)

#13. Tag People in Comments in Google Sheets

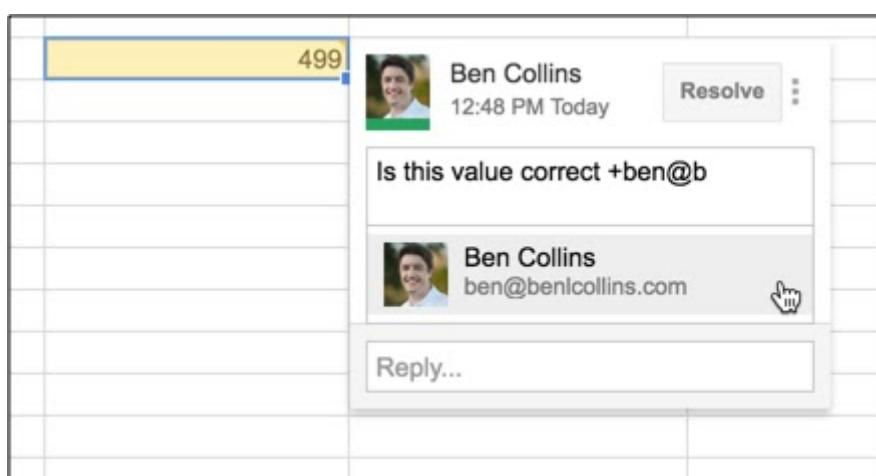
Comments are one of the killer collaboration features embedded in Google Sheets.

You add a comment by right-clicking a cell and choosing **Insert comment**, or through the menu **Insert > Comment**.

You can then tag specific people in each comment if you wish, which means they'll receive an alert that they've been mentioned in a comment.

For example, you might tag Jenny in Finance to ask her to check if your calculation is correct. It's way easier than trying to explain something via email!

You tag someone in a comment very easily by typing a "+" or "@" in the comment box, and then adding the email address of the person you want to notify. As you start typing, Google will auto-suggest email addresses from your contacts list.



#14. Conditional Formatting Across Entire Row

Here's how to apply conditional formatting across an entire row using the custom formula option.

Conditional formatting is a super useful technique for formatting cells in your Google Sheets based on whether they meet certain conditions.

You can apply the formatting across an entire row based on the value in a single cell, by using custom formulas. For example, you could highlight rows in your tables if a value exceeds some threshold. And of course, the power of conditional formatting means this is dynamic, so it will apply the formatting to any rows that change and subsequently meet the condition.

Choose “Custom formula is” at the end of the drop-down list in the conditional formatting editor and enter your formula. Add a \$ (dollar sign) in front of the column reference only, to apply the formatting to the entire row. [Read more details on the blog.](#)

For example, I've highlighted all of the students who scored less than 60 in class, using this formula in the custom formula field:

```
= $C2 < 60
```

File Edit View Insert Format Data Tools Add-ons Help All changes saved in Dr...

fx | Student 1

	A	B	C	D
1	Student Name	Class	Score	
2	Student 1	Math	65	
3	Student 2	Physics	59	
4	Student 3	Comp Sci	64	
5	Student 4	Comp Sci	51	
6	Student 5	Comp Sci	74	
7	Student 6	Math	76	
8	Student 7	Physics	69	
9	Student 8	Math	66	
10	Student 9	Math	57	
11	Student 10	Math	62	
12				
13				
14				

Conditional format rules X

Single color Color scale

Apply to range A2:C11

Format cells if... Custom formula is =\$C2<60

Formatting style Default

B I U S A - Done Cancel

Add another rule >

#15. Trick to Round Numbers to the Nearest Ten, Hundred, etc.

You can use the ROUND function with negative places to round numbers to the nearest ten, hundred, etc.

The ROUND function is a handy function that's used to round decimal places in your numbers when you don't need that level of precision.

Did you know you can specify a negative number of places in the function though?

With a negative number of places specified, it will round your numbers to the nearest ten, hundred, thousand, etc...

For example, this formula would round a number in cell A1 to the nearest ten:

```
= round( A1, -1 )
```

It's obvious with hindsight but something I didn't know about for years! Here's an example:

	A	B	C	D	E	F
1	Number	Round	Round -1	Round -2	Round -3	Round -4
2	22,108.26	22,108	22,110	22,100	22,000	20,000
3						
4						

The formulas for this example are:

	A	B	C	D	E	F
1	Number	Round	Round -1	Round -2	Round -3	Round -4
2	22,108.26	=round(A2)	=round(A2,-1)	=round(A2,-2)	=round(A2,-3)	=round(A2,-4)
3						
4						

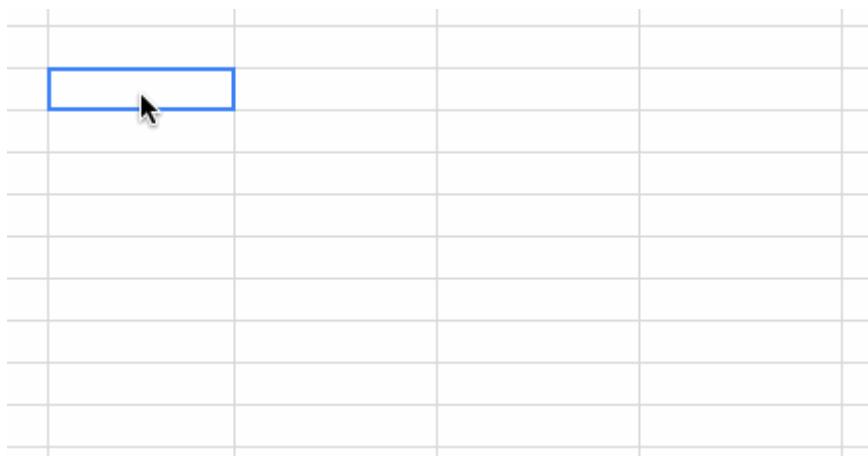
#16. Speed Up Your Data Entry

For this quick and easy tip, you can simply highlight a row range and use your Enter key to move horizontally.

Fill out ranges of data horizontally and use the Enter key to complete the data entry and move sideways to the next cell.

Here's an example of adding column headings efficiently:

1. Start by highlighting the range where the column headings will exist.
2. Type the first column heading in the first cell.
3. Hit Enter and the cursor will jump sideways, not down, to the adjacent cell, where you can enter your next column heading.



You can also achieve this without the highlighting, by using the TAB key, but I find it's an awkward motion that goes against 15 years of muscle memory. 😂

#17. Generate Random Letters With the CODE Function

We use the RAND or RANDBETWEEN functions to generate random numbers, but what about random letters?

There isn't an equivalent random letter generator function, so we have to use a few functions combined to build our own.

I've written before about using the CHAR function to convert numbers into characters according to the current Unicode table. Well, it has a complementary twin, the CODE function, which turns characters back into numbers.

Knowing that, we can construct a random letter generator.

Step 1: convert the letters A and Z into numbers using the CODE function

```
=CODE ("A") => 65
```

```
=CODE ("Z") => 90
```

Step 2: Use RANDBETWEEN to generate a random number between these two numbers

```
=RANDBETWEEN (65, 90) => e.g. 78
```

Step 3: Convert this number back to a letter using the CHAR function

```
=CHAR (78) => N
```

The full formula is:

```
= CHAR( RANDBETWEEN( CODE( "A" ) , CODE( "Z" ) ) )
```

	fx	=char(randbetween(code("A"),code("Z")))		
	A	B	C	
1	N			
2				

Each time you make a change in your Sheet, this formula will generate a new random number.

#18. Create Heatmaps With Your Data

Here's how to create heatmaps from your data using conditional formatting.

Heatmaps are a great way to add context to your data.

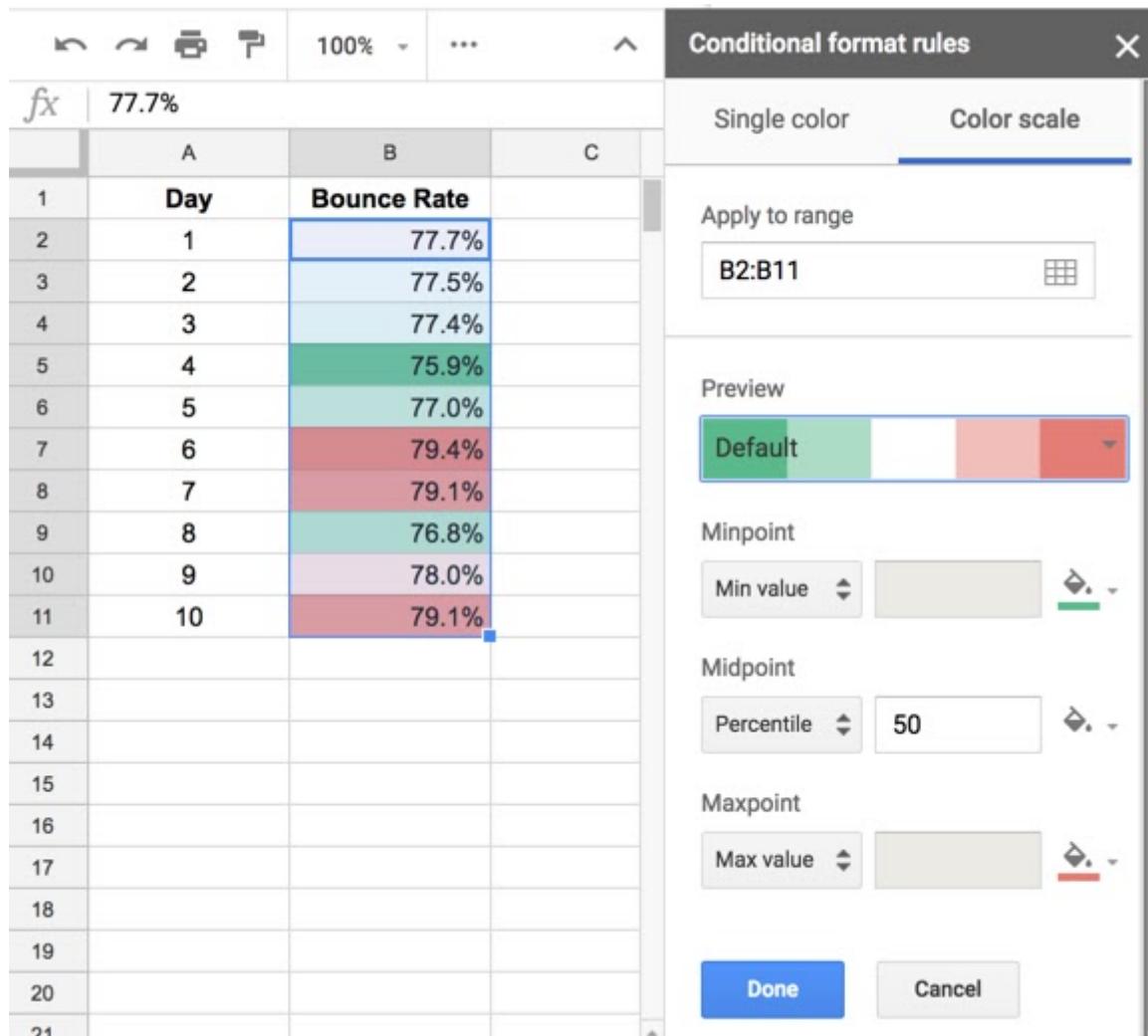
Suppose we have this small dataset of bounce rates:

	A	B	
1	Day	Bounce Rate	
2	1	77.7%	
3	2	77.5%	
4	3	77.4%	
5	4	75.9%	
6	5	77.0%	
7	6	79.4%	
8	7	79.1%	
9	8	76.8%	
10	9	78.0%	
11	10	79.1%	
12			
13			

It's not easy to see, at a glance, which days did well and which did poorly. It takes effort on our part to read and compare each number, and there's a risk that we miss something.

Heatmaps will bring attention to the high and low values in your data, those outliers that demand more attention in your analysis, and they're very easy to create.

Simply highlight the range of data you're using, open Conditional Formatting (under Format menu) and select the Color scale option. Then choose one of the prebuilt color scales or create your own:



You can immediately see the difference.

It's now very easy to identify the days our website underperformed (in red, high bounce rate) and which days it did well (in green, lower bounce rate).

Heatmaps in the wild

Here's another example of a slightly more complex heatmap example, taken from my [Data Analysis in Google Sheets course](#), depicting the customer retention rate for a fictional SaaS company:

	Cohort 12-2016	Cohort 01-2017	Cohort 02-2017	Cohort 03-2017	Cohort 04-2017	Cohort 05-2017	Cohort 06-2017	Cohort 07-2017	Cohort 08-2017	Cohort 09-2017	Cohort 10-2017	Cohort 11-2017	Cohort 12-2017	Cohort 01-2018	Cohort 02-2018	Cohort 03-2018	Cohort 04-2018	Cohort 05-2018
0	100%	69%	60%	59%	54%	50%	49%	49%	47%	94%	94%	86%	87%	92%	93%	95%	95%	96%
1	69%	58%	50%	49%	45%	40%	39%	37%	34%	86%	85%	77%	70%	75%	78%	80%	84%	85%
2	58%	49%	39%	33%	31%	29%	27%	24%	21%	74%	73%	61%	58%	63%	65%	70%	76%	
3	49%	33%	23%	21%	19%	17%	16%	14%	12%	61%	60%	51%	49%	55%	54%	64%		
4	33%	21%	17%	16%	14%	12%	11%	10%	9%	48%	45%	36%	33%	42%	49%	45%		
5	21%	17%	16%	15%	14%	13%	12%	11%	10%	39%	36%	31%	27%	35%	36%			
6	17%	16%	15%	14%	13%	12%	11%	10%	9%	29%	28%	24%	22%	32%	31%			
7	16%	15%	14%	13%	12%	11%	10%	9%	8%	29%	28%	24%	22%	31%				
8	15%	14%	13%	12%	11%	10%	9%	8%	7%	36%	35%	30%	27%	38%				
9	14%	13%	12%	11%	10%	9%	8%	7%	6%	36%	35%	30%	27%	38%				
10	13%	12%	11%	10%	9%	8%	7%	6%	5%	36%	35%	30%	27%	38%				
11	12%	11%	10%	9%	8%	7%	6%	5%	4%	36%	35%	30%	27%	38%				
12	11%	10%	9%	8%	7%	6%	5%	4%	3%	36%	35%	30%	27%	38%				
13	10%	9%	8%	7%	6%	5%	4%	3%	2%	36%	35%	30%	27%	38%				
14	9%	8%	7%	6%	5%	4%	3%	2%	1%	36%	35%	30%	27%	38%				
15	8%	7%	6%	5%	4%	3%	2%	1%		36%	35%	30%	27%	38%				
16	7%	6%	5%	4%	3%	2%	1%			36%	35%	30%	27%	38%				
17	6%	5%	4%	3%	2%	1%				36%	35%	30%	27%	38%				

Again, see how the heatmap adds context and gives you a clear sense of what's happening.

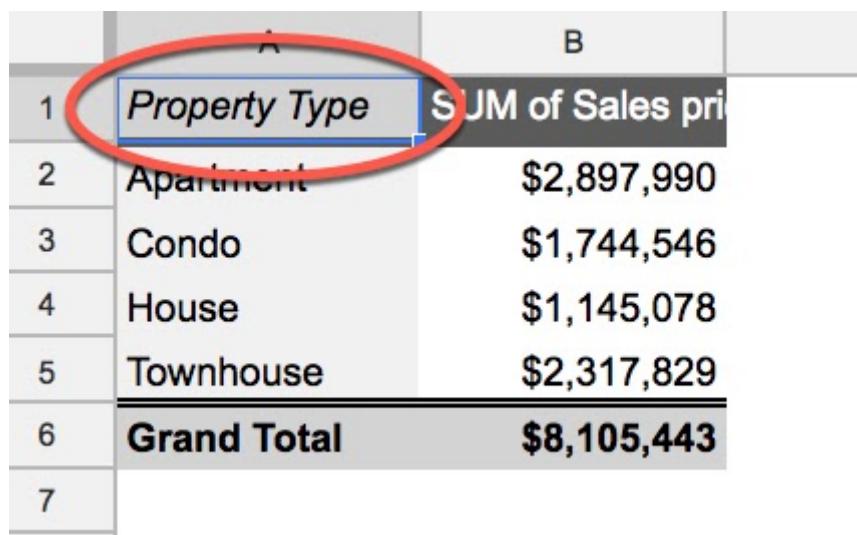
#19. Quickly Copy Pivot Tables

This tip shows you how to quickly copy Pivot Tables using the top left corner cell.

Oftentimes you'll find yourself wanting to replicate a Pivot Table, perhaps as a starting point for further data exploration.

There's a quick trick for copying an existing Pivot Table, rather than starting over. It also gives you the option of moving your Pivot Table to a different tab.

Click into the **top left corner cell** of your Pivot Table and click copy (Cmd + C on a Mac, or Ctrl + C on a PC/Chromebook). This adds the Pivot Table to your clipboard and you can paste it wherever you want in your Sheet (Cmd + V on a Mac, or Ctrl + V on a PC/Chromebook).



	A	B
1	Property Type	SUM of Sales pri
2	Apartment	\$2,897,990
3	Condo	\$1,744,546
4	House	\$1,145,078
5	Townhouse	\$2,317,829
6	Grand Total	\$8,105,443
7		

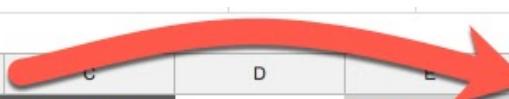
Note: You need to ensure there is enough space available wherever you wish to paste a copy of your Pivot Table (i.e. enough empty cells) or you'll see the #REF! error.

#20. Rename Pivot Table Column Headings

With this simple tip, you can rename your Pivot Table headings to make your data much more presentable and understandable.

Did you know that **you can type over those ugly default column headings** in your Pivot Tables?

For example, instead of your column heading saying "SUM of Sales price", you could change it to "Revenue, \$", which is more presentable and easier to understand.



	A	B	C	D	E	F	G
1	Property Type	COUNTA of Prop	SUM of Sales pri		Property Type	Count	Revenue, \$
2	Apartment		8	\$3,315,497	Apartment	8	\$3,315,497
3	Condo		5	\$1,992,772	Condo	5	\$1,992,772
4	House		2	\$2,333,643	House	2	\$2,333,643
5	Studio		2	\$448,052	Studio	2	\$448,052
6	Townhouse		3	\$2,317,829	Townhouse	3	\$2,317,829
7	Grand Total		20	\$10,407,793	Grand Total	20	\$10,407,793
8							

Go on, give it a try!

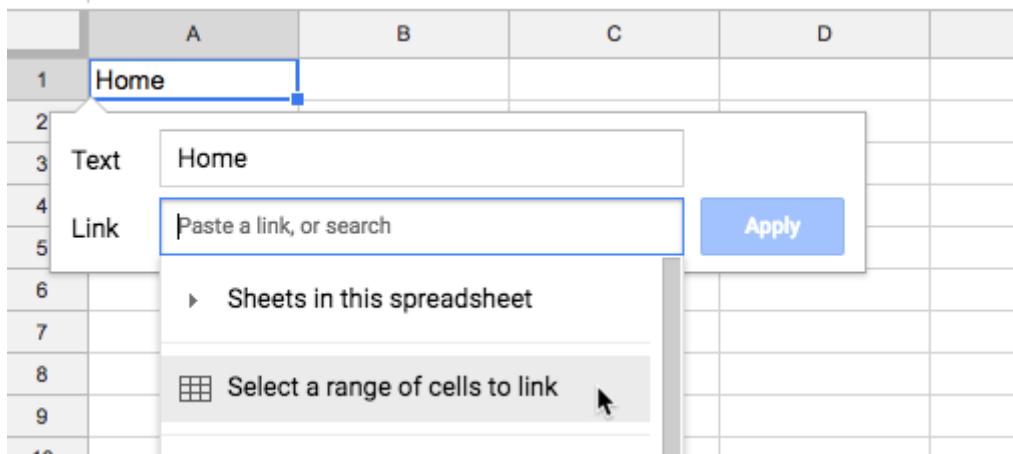
#21. Hyperlink to Different Parts of Your Sheet

You can add hyperlinks to different tabs or even directly to individual cells within your Google Sheets.

They're super easy to create. You don't even have to write any formulas yourself. Simply:

1. Right click on the cell that you want to turn into a clickable hyperlink
2. Click "Insert link"
3. Choose either "Sheets in this spreadsheet" or "Select a range of cells to link"

That's it!



Here's a few examples of how you could use this:

- Add a Home button to every tab in your Sheet so you can quickly get back to the first tab
- Create a "table of contents" for your Sheet
- Link to important calculation cells so they can be easily accessed

#22. Repeat Number Groups With the ROW & MOD Functions

Suppose you're organizing a seating plan or a conference or some other event that requires putting people into teams ("All the one's on me! All the two's with John! All the three's with Sue!"), like this:

	A	B
1	Group	Name
2	1	Person 1
3	2	Person 2
4	3	Person 3
5	1	Person 4
6	2	Person 5
7	3	Person 6
8	1	Person 7
9	2	Person 8
10	3	Person 9
11		
12		

Rather than create the grouping manually, which involves a lot of tedious copying and pasting, you can use the ROW and MOD functions to automate the process.

Step 1: Insert the row formula into the cell where your first group starts

```
=ROW()
```

Step 2: Subtract the current row number (e.g. if you're on row 1 subtract 1, if you're on row 17 subtract 17), so your formula looks like this:

```
=ROW() - 1
```

Step 3: Wrap the formula in step 2 with the MOD function, which gives the repeating number pattern, and set the divisor to equal the number of groups you want (e.g. 3):

```
=MOD(ROW() - 1, 3)
```

Step 4: Add 1 to the formula to start your group index from 1 instead of 0

```
=MOD(ROW() - 1, 3) + 1
```

Step 5: Drag the formula down as far as you need and it'll create repeating numbered groups for you!

	A	B
1	1	
2	2	
3	3	
4	1	
5	2	
6	3	
7	1	
8	2	
9	3	
10	1	
11		
12		
13		
14		



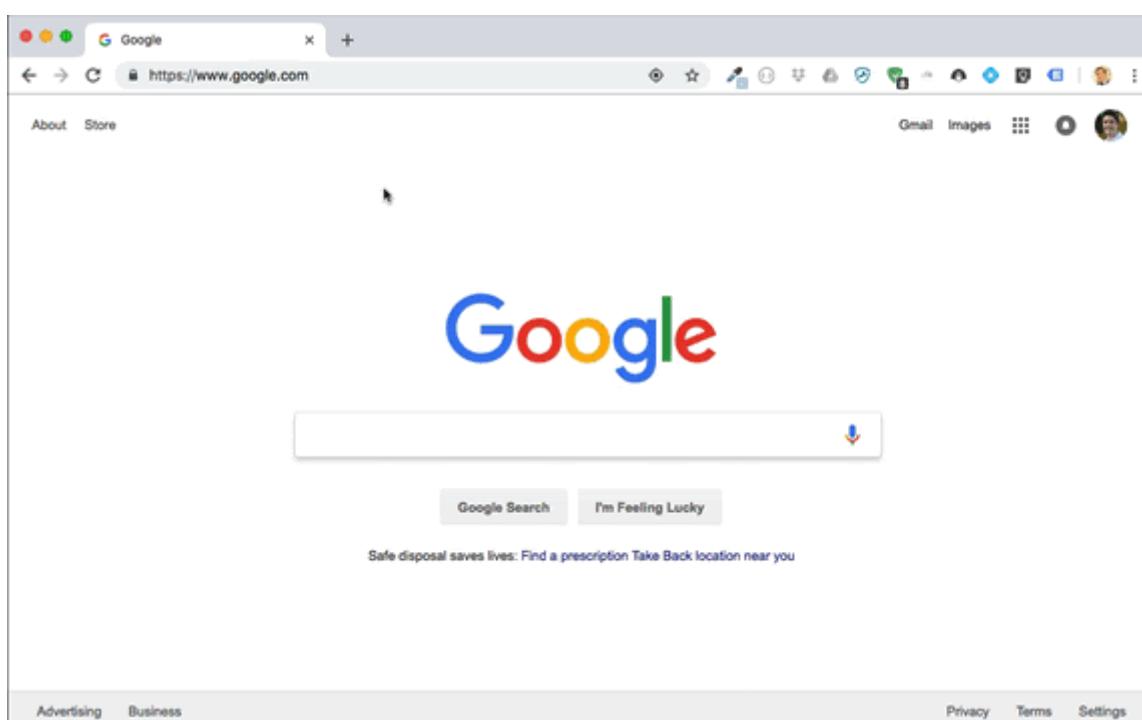
Another example: If you wanted to start your grouping on row 17 and repeat in groups of 7, your formula would look like this:

```
=MOD(ROW() - 17, 7) + 1
```

#23. Create Google Sheets Instantly

Create a new Google Sheet instantly with just your web browser.

Type **sheets.new** into your web browser to instantly create a new Google Sheet (if you're not logged into your Google account you'll be prompted to do that first).



It also works with Docs, Forms, Sites and Slides as well, and it worked when I tested it in a Firefox browser too.

Wow!

#24. Advanced Formula Workout Challenge

(Editor's note: this example worked at the time of writing (November 2018) but if Wikipedia makes changes to the webpage in this example, the formula may no longer work.)

This tip will show you how to solve challenging formulas by taking baby steps.

Let's suppose we want to scrape the population data from this [table in Wikipedia](#):

Cities proper by population <small>[edit]</small>								
Rank	City	Image	Population	Definition	Total area (km²)	Population density (/km²)	Country	
1	Chongqing		30,165,500 ^[8]	Municipality	82,403 ^[7]	366	 China	
2	Shanghai		24,183,300 ^[8]	Municipality	6,340.5 ^[9]	3,814	 China	
3	Beijing		21,707,000 ^[10]	Municipality	16,411 ^[7]	1,267	 China	

And put it into our Google Sheet like this, so we can use it for analysis:

	A	B	C	
1	City	Country	Population	
2	Chongqing	China	30,165,500	
3	Shanghai	China	24,183,300	
4	Beijing	China	21,707,000	
5	Istanbul	Turkey	15,029,231	
6	Karachi	Pakistan	14,910,352	
7	Dhaka	Bangladesh	14,399,000	

The challenge is to do this with a single formula...!

Ready?

Step 1: Open a new Google Sheet (bonus points for using the last tip and just typing `sheets.new` into your browser window)

Step 2: In cell A1, insert a basic IMPORTHTML formula to scrape the raw table of population data from Wikipedia:

```
=IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population","table",2)
```

	A	B	C	D	E	F	G	H
1	Rank	City	Image	Population	Definition	Total area (km²)	Population density (/km²)	Country
2	1	Chongqing		7007301655000	Municipality	7004824030000	366	China
3	2	Shanghai		7007241833000	Municipality	7003634050000	3,814	China
4	3	Beijing		7007217070000	Municipality	7004164110000	1,267	China

The data has some issues, but it's a start.

Step 3: Pick just the columns we want, by wrapping the Import formula with a Query formula. Note that we have to use the Col1 notation rather than the column letter in our Select statement.

```
=QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by population","table",2),"select Col2, Col18, Col4",1)
```

	A	B	C	D	E	
1	City	Country	Population			
2	Chongqing	China	70073016550000000000±30,165,500[6]			
3	Shanghai	China	70072418330000000000±24,183,300[8]			
4	Beijing	China	70072170700000000000±21,707,000[10]			
5	Istanbul	Turkey	7007150292310000000±15,029,231[11]			

Step 4: Hmm, that population column is messed up! Regex to the rescue!

At this point, we'll deal with the population column on its own and come back to our main formula later.

So make a copy of this Import formula in cell F1 and change it to:

```
=QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by population", "table", 2), "select Col4", 1)
```

Now we should have a copy of just the population column in column F, like so:

	A	B	C	D	E	F	G	H
1	City	Country	Population			Population		
2	Chongqing	China	7007301655000000000•30,165,500[6]			7007301655000000000•30,165,500[6]		
3	Shanghai	China	7007241833000000000•24,183,300[8]			7007241833000000000•24,183,300[8]		
4	Beijing	China	7007217070000000000•21,707,000[10]			7007217070000000000•21,707,000[10]		
5	Istanbul	Turkey	7007150292310000000•15,029,231[11]			7007150292310000000•15,029,231[11]		
6	Los Angeles	United States	7007131000000000000•12,100,000[12]			7007131000000000000•12,100,000[12]		

Step 5: Wrap this population only formula in cell F1 with a Regex formula:

```
=REGEXEXTRACT(QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population","table",2),"select Col4",1),"♠([0-9,]*")")
```

Hmm, that gives us a #N/A error... 🤔

Step 6: Turn this into an Array Formula and get the column of population numbers!

```
=ARRAYFORMULA (REGEXEXTRACT (QUERY (IMPORTHTML ("https://en.wikipedia.org/wik  
i/List_of_cities_proper_by_population", "table", 2), "select  
Col4", 1), "♠([0-9,]*)"))
```

F
#N/A
30,165,500
24,183,300
21,707,000
15,029,231
14,910,352
14,000,000

The Regex formula uses this expression "`\p{([0-9,]*)}`" to extract the numbers after the funny **♠** symbol.

We still have two problems to solve though: i) we need to convert the strings into actual numbers and ii) fix the #N/A column heading...

Step 7: Use the SUBSTITUTE function to remove the commas and convert the strings into numbers:

```
=ARRAYFORMULA(SUBSTITUTE(REGEXEXTRACT(QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population","table",2),"select Col4",1),"♦([0-9,]*")",",",")*1))
```

The multiplication by 1 at the very end coerces the strings into numbers after the commas have been removed.

Step 8: Use the IFERROR function to fix that pesky #N/A error at the top of our column heading, and replace the #N/A with the word "Population":

```
=ARRAYFORMULA(IFERROR(SUBSTITUTE(REGEXEXTRACT(QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population","table",2),"select Col4",1),"♦([0-9,]*")",",",")*1,"Population")))
```

Nice, now we have our population column as numbers:

F
Population
30165500
24183300
21707000
15029231
14910352

Step 9: Go back to our main formula in cell A1 and remove the old population column (the one with all the funny numbers). So our formula in A1 should now be:

```
=QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population","table",2),"select Col2, Col18",1)
```

	A	B	C	D	E	F	
1	City	Country				Population	
2	Chongqing	China				30165500	
3	Shanghai	China				24183300	
4	Beijing	China				21707000	
5	Istanbul	Turkey				15029231	
6	Karachi	Pakistan				14910352	
7	Dhaka	Bangladesh				14399000	

Step 10: All that's left is to join these two ranges, in columns A, B and F, using the curly bracket notation, like so (shown with line breaks to illustrate the two ranges):

```
={
  QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population", "table", 2), "select Col2, Col8", 1)
,
  ARRAYFORMULA(IFERROR(SUBSTITUTE(REGEXEXTRACT(QUERY(IMPORTHTML("https://en.wikipedia.org/wiki/List_of_cities_proper_by_population", "table", 2), "select Col4", 1), "♠([0-9,]*), , , "")*1, "Population")))
}
```

Ok, we're done 😊

The output is:

	A	B	C	
1	City	Country	Population	
2	Chongqing	China	30,165,500	
3	Shanghai	China	24,183,300	
4	Beijing	China	21,707,000	
5	Istanbul	Turkey	15,029,231	
6	Karachi	Pakistan	14,910,352	
7	Dhaka	Bangladesh	14,399,000	

(Feel free to delete the workings in column F.)

Debrief:

Yes, it would have probably been much quicker to cut and paste the table from Wikipedia and fix the funny formats manually.

But where's the fun in that? 😅

In all seriousness though, think of this as an exercise in combining some of the most useful single functions in Google Sheets to create really powerful formulas.

Lastly, there are probably other ways to solve this! That's the beauty of spreadsheets.

Resources:

The Query function is covered in Days 14 and 15 of my free [30 Day Advanced Formulas course](#) and in this [blog post](#).

Array Formulas are covered in Day 17 of my free 30 Day Advanced Formulas course and in this [blog post](#).

The Import function is covered in Day 19 of my free 30 Day Advanced Formulas course and in this [blog post](#).

The Regex function is covered in Day 20 of my free 30 Day Advanced Formulas course.

#25. Neat Trick With Freeze Panes

Freeze panes is a useful and simple technique to lock the top row (or rows) of your spreadsheet, so that they remain in view even as you scroll down your datasets. You're effectively anchoring them in place.

It's really useful because you can keep your column headings in view as you move about your data.

Traditionally, you freeze panes through the **View > Freeze** menu, but there is another quicker way.

Hover over the bottom of the column heading letters until you see the hand icon, like this:

	A	B	C
1	Listing	Property Type	Lead Sou
2	1	Apartment	Referral
3	2	House	Previous cli

Click and hold down your mouse button, and then drag the thick line down:

	A	B	C	
1	Listing	Property Type	Lead Source	R
2	1	Apartment	Referral	
3	2	House	Previous client	

Let go when you've reached the row level you want to freeze (in this case, just the top row):

	A	B	C	
1	Listing	Property Type	Lead Source	R
2	1	Apartment	Referral	
3	2	House	Previous client	

This technique also works for freezing columns!

#26. Protect Individual Sheets

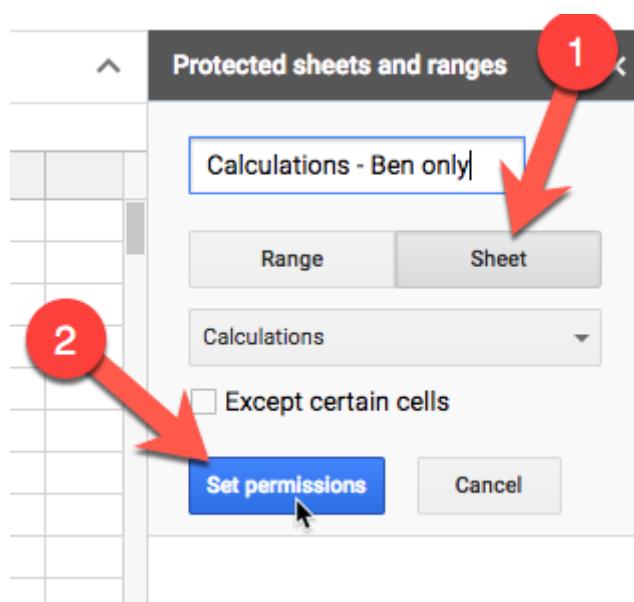
Control who can edit individual tabs (sheets) within a Google Sheet with the Protect Sheet option.

You have a lot of control when it comes to sharing Google Sheets and deciding who can edit which parts of your Google Sheet.

Suppose you're working on a complex Google Sheet with multiple team members (or maybe sharing a Google Sheet with lots of students) and you're worried about somebody messing up the calculations tab, or the charts tab, etc.. What you really want is for your team members to only work on their tab.

Under **Tools > Protect sheet...** you can set tab level permissions for users of a Google Sheet.

You can ensure people just edit their sections, and you retain overall control of the Sheet.



And then under **Set permissions**, you can check or uncheck who has access to the specific tab:

Range editing permissions

- Show a warning when editing this range
- Restrict who can edit this range

Custom...

Choose who can edit



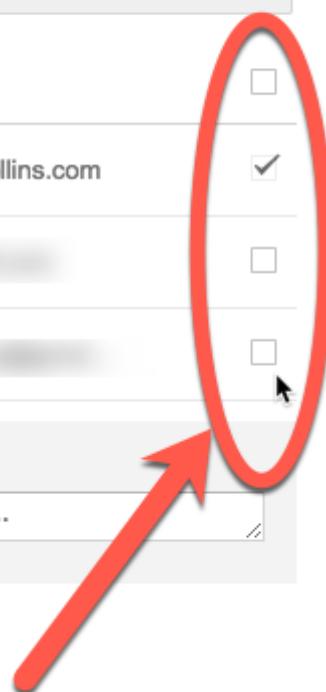
Ben Collins (you) ben@benlcollins.com



Add editors:

Enter names or email addresses...

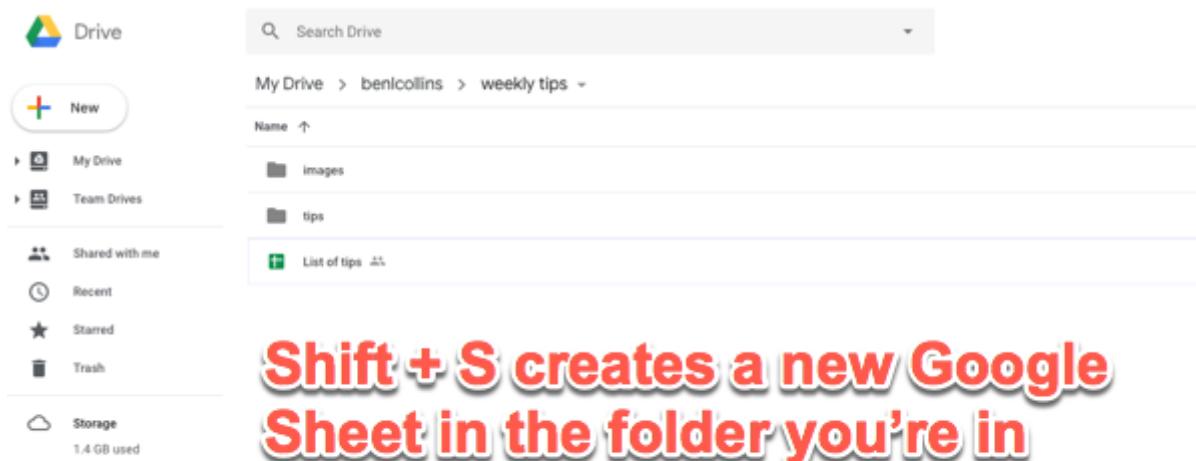
Done



#27. Creating a New Sheet in the Folder You're In

Previously, I shared how you can type **Sheets.new** into your browser to create a new Google Sheet. It has the one drawback of creating the Sheet in your root Drive folder, so you have to manually move it to the folder you want.

Instead, if you have a specific folder open in Drive and you want to create a Sheet in that folder, just hit the shortcut **Shift + S** when you're inside that folder to create a new Google Sheet there.



#28. Automate your Sheets with Macros

Macros are small programs you create inside of Google Sheets without needing to write any code. They allow you to automate tasks.

They work by recording your actions as you do something and saving these actions as a “recipe” that you can reuse again with a single click.

You can also write Apps Script functions directly and import them into your Macros menus, which gives you a lot of flexibility.

Here's my favorite macro, written in script and imported to the macro menu:

Resetting Filters

I use filters on my data tables all the time, and it's annoying that there's no way to clear all your filters in one go. You have to manually reset each filter in turn or completely remove the filter and re-add from the menu, both of which are tedious.

So let's create a macro to remove and re-add the filter!

Then we can be super efficient by running it with a single menu click or, even better, with a shortcut key.

- From your Google Sheet, open the Script Editor window: **Tools > Script editor**

- Copy in the following code:

```
function resetFilter() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var range = sheet.getDataRange();
  range.getFilter().remove();
```

```
range.createFilter();  
}
```

3. Back in your Google Sheet, use the Macro Import option to import this function as a macro (**Tools > Macros > Import**)

4. When you run it, it will remove and then re-add filters to your data range in one go.

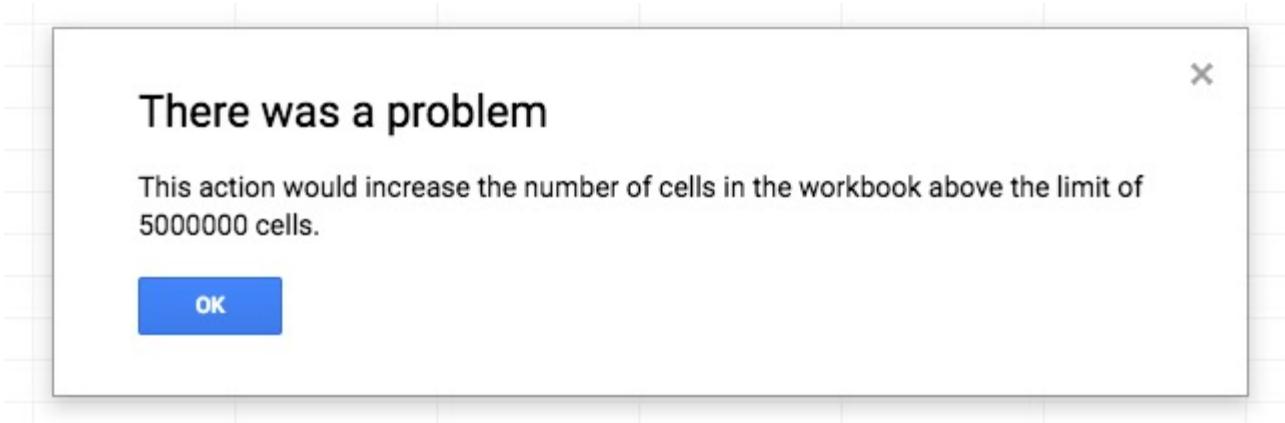
For example, in the following GIF, I have a data table with three filters applied. Rather than remove them one by one, I simply run the macro which automatically removes and re-adds filters to my table to get me back to my unfiltered data.

	A	B	C	D	E	F	G	H	I	J	K
1	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %	Gross Commission Income				
2	1	Apartment	Referral	Buyer	\$328,484	3.00%	\$9,855				
12	11	Apartment	Referral	Buyer	\$376,895	3.00%	\$11,307				
16	15	Apartment	Referral	Buyer	\$417,507	3.00%	\$12,525				
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											
38											
39											

#29. 5 Million Cells

You now have 5 million cells to play with in your Google Sheets!

When I heard this news, the first thing I did was open up a Google Sheet and keep adding rows until I blasted past 2 million and received the new error message at 5 million (because I'm a nerd like that):



I haven't had a chance to test the performance of much larger Sheets yet, but I will share results when I do.

When the Sheets product team [first announced the 5 million cell limit at Google Next 18 conference](#), they insisted the performance side of the equation would be capable enough.

So the question is, what will you do with all that extra real-estate?

For reference, [here are all the file size limits for Google Drive](#) (incl. the 5 million for Sheets).

#30. Collaboration & Filters

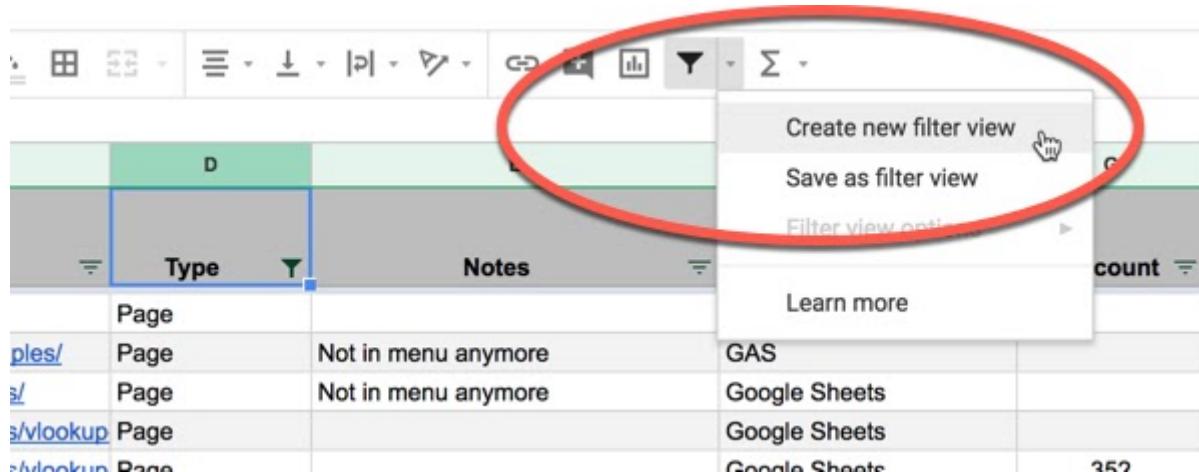
You can use Filter Views to allow different users of a Google Sheet to use their own filters on a dataset.

Earlier this week, my wife and I were writing our Christmas cards for this holiday season. Of course, we use a Google Sheet to keep track of all the people we want to send cards too.

The problem was I wanted to filter the dataset to show just the names I was writing to, whereas my wife wanted to be able to see the names she was working on. Whenever I applied a filter to the dataset, it would change the view in my wife's Google Sheet too. Argh! What to do?

The answer is to save your filter set as a Filter View, give it a name and then you can use it without disturbing others viewing the same dataset.

If you already have a filter setup, then you can just save that as a Filter View and give it a memorable name. Alternatively, you can create a new filter under this menu too:



Your Filter View will then show up, with a dark grey box around it.

Here you can rename it (1), adjust the range (2) and update, delete or duplicate it (3). The X on the right side allows you to also close the Filter View:

ID	Name	Hyperlink	Type	Notes	Topic	Wors
19	About Page	https://www.benlcollins.com/about/	Page		About page	
49	Apps Script Examples	https://www.benlcollins.com/apps-script-examples/	Page	Not in menu anymore	GAS	
50	Formula Examples	https://www.benlcollins.com/formula-examples/	Page	Not in menu anymore	Google Sheets	
51	Google Sheets VLOOKUP left	https://www.benlcollins.com/formula-examples/vlookup-left	Page		Google Sheets	
52	Google Sheets VLOOKUP multiple values	https://www.benlcollins.com/formula-examples/vlookup-multiple-values	Page		Google Sheets	352
54	Google Sheets Zoom	https://www.benlcollins.com/formula-examples/zoom/	Page		Google Sheets	
55	Google Sheets Formula First and Last current month	https://www.benlcollins.com/formula-examples/find-and	Page		Google Sheets	
56	Google Sheets Formula First and Last prior month	https://www.benlcollins.com/formula-examples/find-and-prior-month	Page		Google Sheets	
58	Google Sheets charts data labels	https://www.benlcollins.com/formula-examples/data-labels	Page		Google Sheets	378
59	Google Sheets combine text and numbers	https://www.benlcollins.com/formula-examples/text-number	Page		Google Sheets	
61	Google Sheets Array Formula Intro	https://www.benlcollins.com/formula-examples/array-formula-intro	Page		Google Sheets	499
62	Google Sheets Dynamic named ranges	https://www.benlcollins.com/formula-examples/dynamic-named-ranges	Page		Google Sheets	296

Once you have a Filter View saved, it's available to use again under the filter button menu, along with the *Filter view options* menu:

Create new filter view

Filter view options ▶

- None
- Filter 1
- Filter 2
- ✓ Page Filter
- Learn more

Rename

Update Range

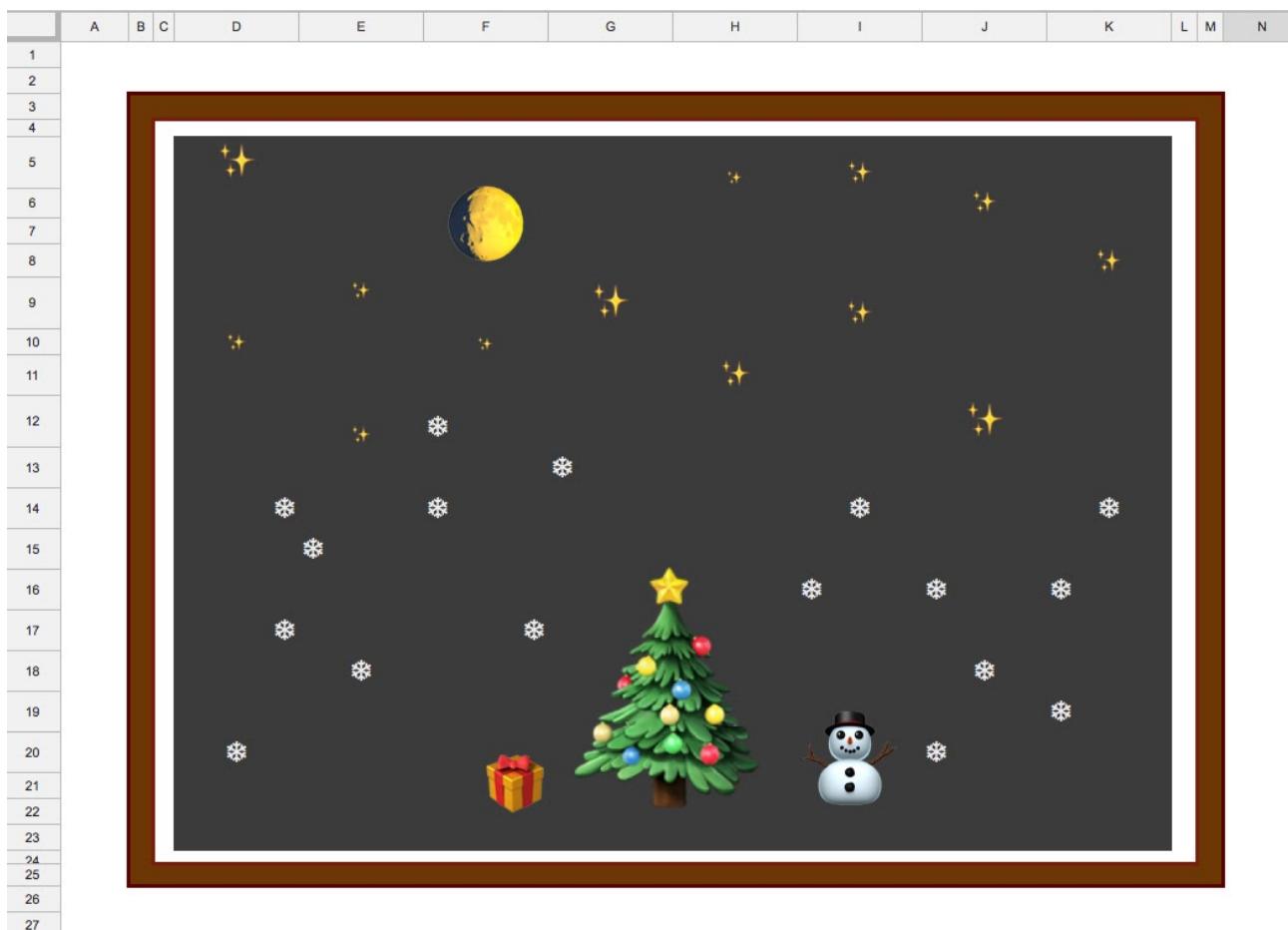
Duplicate

Delete

The major benefit of these Filter Views is that other people working with this dataset in the same Sheet are unaffected. They can continue to see the whole dataset or create their own Filter Views, independent of yours.

Another benefit of course, is that you can save more complex filters using multiple columns to easily return to them.

#31. Create Pictures With the CHAR Function



The CHAR function converts a number into a character according to the current Unicode table, for example CHAR(127764) creates the waxing gibbous moon symbol.

These are the unicode characters used in the picture above:

	fx	=char(10052)
1	10052	❄️
2	127764	🌙
3	10024	✨
4	127876	🎄
5	9924	⛄
6	127873	🎁
7		

(Unfortunately, the Christmas tree may not show up for you, so feel free to replace it with something else! Just Google "unicode character for XYZ" and then grab the number part of the HTML Entity decimal.)

#32. Use the YEARFRAC Function to Calculate the % of the Year Left

This tip was inspired by a [Twitter account](#) that simply shares the % of the year completed:



Let's replicate it in Google Sheets using the obscure YEARFRAC function.

We'll build it up in steps, starting with today's date in cell A1:

```
=TODAY()
```

Extract the year from this date by changing the formula to:

```
=YEAR(TODAY())
```

Then calculate the date at the start of the year:

```
=DATE(YEAR(TODAY()), 1, 1)
```

Next, use the YEARFRAC function with the start of the year date you just calculated and today's date as the start/end dates, like so (spaces added for comprehension):

```
=YEARFRAC(DATE(YEAR(TODAY()), 1, 1), TODAY())
```

Finally, format the output as a percentage.

As you can see, when I took this screenshot, we were over 99% the way through the year:

The screenshot shows a Google Sheets interface. In the top-left corner, there's a formula bar with the text '=YEARFRAC(DATE(YEAR(TODAY()),1,1),TODAY())'. Below the formula bar is a table with three columns labeled A, B, and C. Row 1 contains the formula in cell A1. Row 2 contains the result '99.17%' in cell A2. Row 3 is empty. Row 4 is also empty. The background of the table cells is light gray, while the header row has a darker gray background.

	A	B	C
1	99.17%		
2			
3			

An extra challenge for you: use the SPARKLINE formula to create a mini in-cell bar chart showing the year's progress.

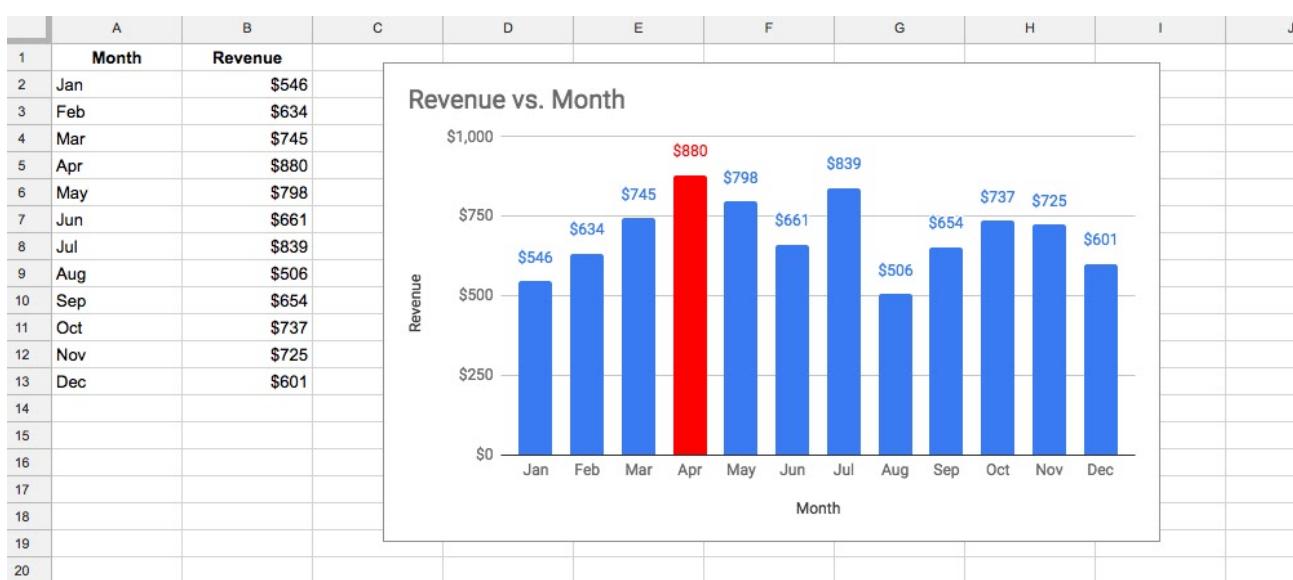
#33. Format Individual Data Points In Charts

Here's how to format individual data points in your Google Sheets charts to make them stand out.

By formatting specific parts of your chart, you can draw attention to them. This makes it quicker and easier for your audience to read your chart. And that's a good thing.

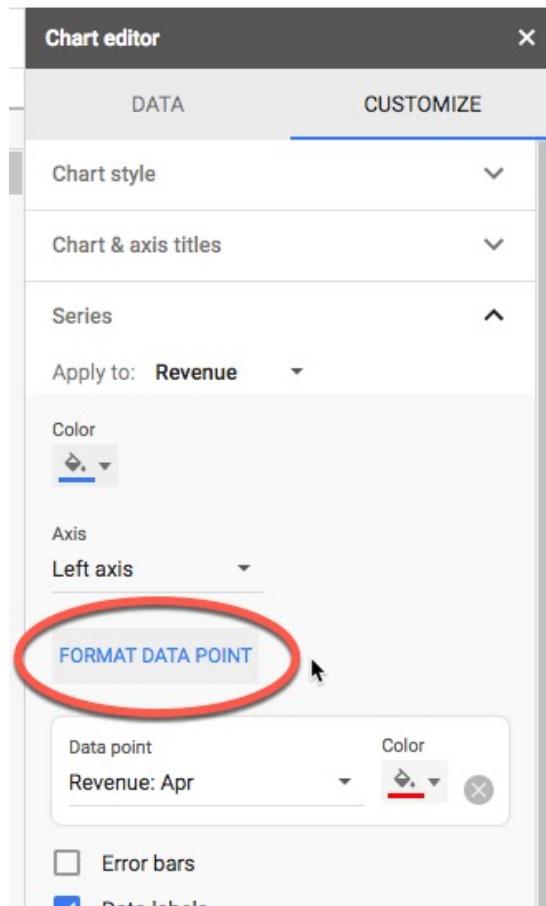
For example, you could highlight the largest values, or values over a certain threshold.

Google Sheets has the ability to format data points individually in charts, like this:



It's a really useful addition to the chart tool and it's very easy to use too.

There are two methods: either right click a specific data point on your chart and select "Format data point", or click the the **Chart Editor > CUSTOMIZE > Series > FORMAT DATA POINT** button in the chart editor sidebar:



Can you format multiple data points in a chart?

Yes, you can!

Does it work for other types of charts?

Yes, it works with bar charts, column charts, line charts and pie charts. It does not work with the area charts.

Can you set it to format points automatically, for example when they go above a threshold value?

Unfortunately not. However, I'd envisage this being possible in the future. I hope they add it to Apps Script too, so it's possible programmatically.

[Read more about formatting data points in Google Sheets charts, including how to label data points individually.](#)

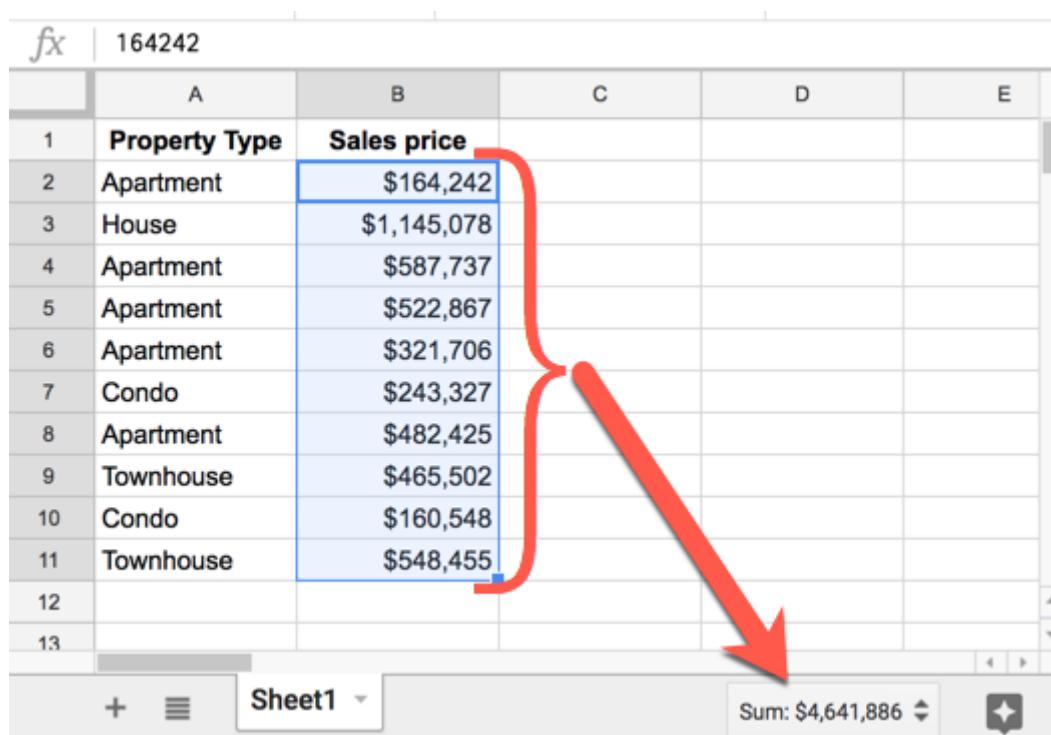
#34. Quickly Summarize Your Data

Summarize your data at a glance by highlighting it and looking at the status bar tool.

This is a super easy tip.

You don't need to write any formulas or click any menus to get summary statistics about your data.

Simply highlight the data and look down in the bottom right corner of the status bar, where you'll see metrics about your data:



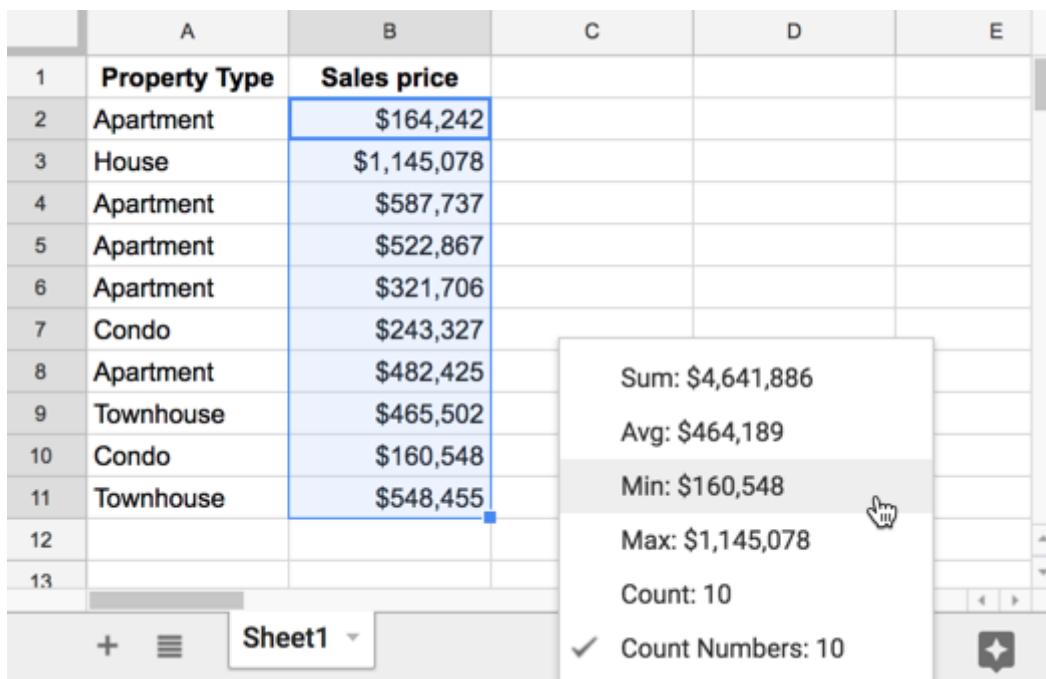
	A	B	C	D	E
1	Property Type	Sales price			
2	Apartment	\$164,242			
3	House	\$1,145,078			
4	Apartment	\$587,737			
5	Apartment	\$522,867			
6	Apartment	\$321,706			
7	Condo	\$243,327			
8	Apartment	\$482,425			
9	Townhouse	\$465,502			
10	Condo	\$160,548			
11	Townhouse	\$548,455			
12					
13					

Easy, huh?!

FAQ's

Can you display different metrics?

Yes! Click on the metric to open the menu where you can choose Count, Count of Numbers, Average, Min and the Max metrics.



A screenshot of a Google Sheets spreadsheet titled "Sheet1". The data is as follows:

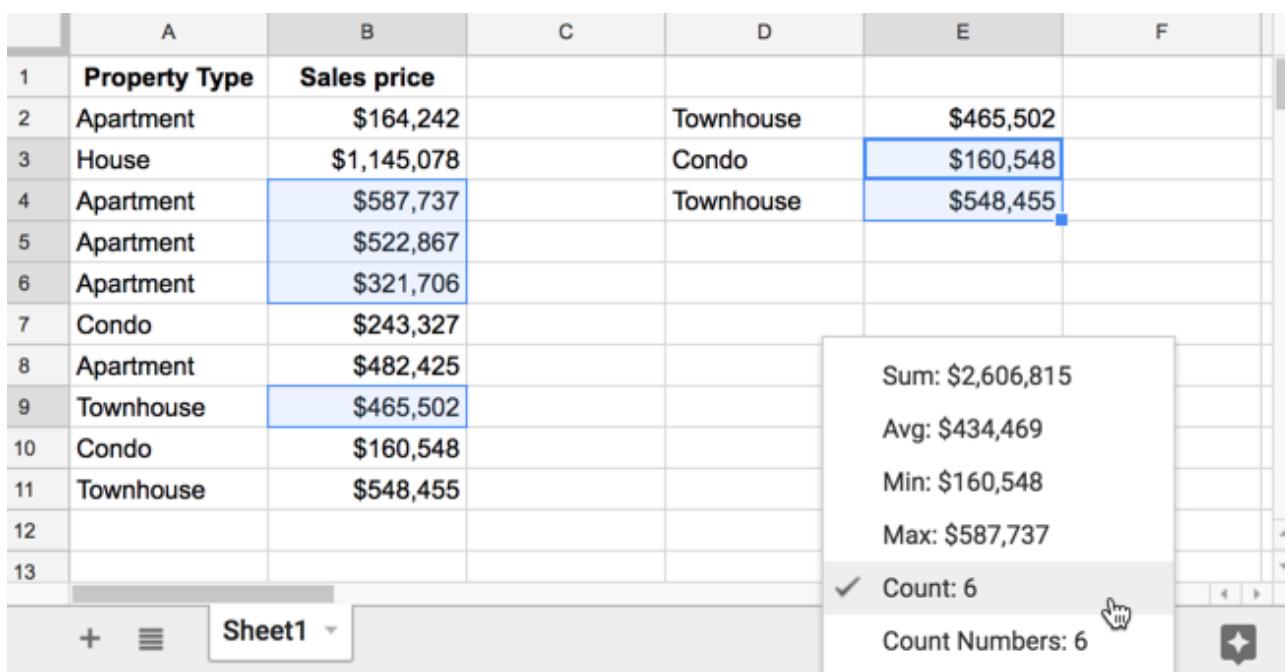
	A	B	C	D	E
1	Property Type	Sales price			
2	Apartment	\$164,242			
3	House	\$1,145,078			
4	Apartment	\$587,737			
5	Apartment	\$522,867			
6	Apartment	\$321,706			
7	Condo	\$243,327			
8	Apartment	\$482,425			
9	Townhouse	\$465,502			
10	Condo	\$160,548			
11	Townhouse	\$548,455			
12					
13					

The cell B2 is selected. A context menu is open, listing the following metrics:

- Sum: \$4,641,886
- Avg: \$464,189
- Min: \$160,548
- Max: \$1,145,078
- Count: 10
- Count Numbers: 10

Does the data have to be continuous?

No! Whatever data you highlight will be included in the summary metrics, e.g.:



A screenshot of a Google Sheets spreadsheet titled "Sheet1". The data is as follows:

	A	B	C	D	E	F
1	Property Type	Sales price				
2	Apartment	\$164,242		Townhouse	\$465,502	
3	House	\$1,145,078		Condo	\$160,548	
4	Apartment	\$587,737		Townhouse	\$548,455	
5	Apartment	\$522,867				
6	Apartment	\$321,706				
7	Condo	\$243,327				
8	Apartment	\$482,425				
9	Townhouse	\$465,502				
10	Condo	\$160,548				
11	Townhouse	\$548,455				
12						
13						

The cells B2, B3, B4, B5, B6, B7, B8, B9, B10, and B11 are selected. A context menu is open, listing the following metrics:

- Sum: \$2,606,815
- Avg: \$434,469
- Min: \$160,548
- Max: \$587,737
- Count: 6
- Count Numbers: 6

#35. How To Remove Duplicates

This tip is a single formula you can use to quickly remove duplicates in Google Sheets.

Check out the accompanying [article covering five different methods to remove duplicates in Google Sheets](#) using Add-Ons, Formulas, Pivot Tables, Conditional Formatting or Apps Script.

Here's how to use the UNIQUE function to remove duplicates in Google Sheets.

This single formula removes duplicates from your data ranges. It's handy both for de-duplicating datasets and inside nested formulas to remove duplicate entries.

It's super easy to and only requires one argument (the range of data), e.g.:

```
=UNIQUE(A1:D11)
```

The screenshot shows a Google Sheets interface with a table of data in columns A through D. Column A is labeled "Invoice Number", column B is "Product ID", column C is "Date", and column D is "Revenue". Row 1 contains the headers. Row 2 has data: 174-A, ABC123, 2/16/2017, \$150. Rows 3 and 4 show 196-X, XYZ100, 4/3/2017, \$50. Rows 5 and 6 show 327-A, ABC123, 4/3/2017, \$150. Rows 7 and 8 show 485-ABC, ABC123, 3/7/2017, \$150. Rows 9 and 10 show 511-XYZ, XYZ100, 1/28/2017, \$50. Rows 11 and 12 show 585-ABC, XYZ100, 3/7/2017, \$50. Row 13 is blank. Cell F1 contains the formula =UNIQUE(A1:D11). A tooltip for the UNIQUE function is displayed, stating: "Returns unique rows in the provided source range, discarding duplicates. Rows are returned in the order in which they first appear in the source range." It also includes a "range" section describing the parameter and a link to learn more about the function.

A	B	C	D	E	F	G	H	I
1	Invoice Number	Product ID	Date	Revenue	=UNIQUE(A1:D11)			
2	174-A	ABC123	2/16/2017	\$150	UNIQUE(range)			
3	196-X	XYZ100	4/3/2017	\$50	Example			
4	196-X	XYZ100	4/3/2017	\$50	UNIQUE(A2:B20)			
5	327-A	ABC123	4/3/2017	\$150	Summary			
6	327-A	ABC123	4/3/2017	\$150	Returns unique rows in the provided source range, discarding duplicates. Rows are returned in the order in which they first appear in the source range.			
7	485-ABC	ABC123	3/7/2017	\$150	range			
8	485-ABC	ABC123	3/7/2017	\$150	The data to filter by unique entries.			
9	511-XYZ	XYZ100	1/28/2017	\$50	Learn more about UNIQUE			
10	585-ABC	XYZ100	3/7/2017	\$50				
11	585-ABC	XYZ100	3/7/2017	\$50				
12								
13								

The UNIQUE formula removes the duplicate rows in this example. In other words, it compares all of the columns to find duplicates.

The output looks like this, with the original table on the left and the de-duplicated table on the right:

fx =UNIQUE(A1:D11)

	A	B	C	D	E	F	G	H	I	
1	Invoice Number	Product ID	Date	Revenue		Invoice Number	Product ID	Date	Revenue	
2	174-A	ABC123	2/16/2017	\$150		174-A	ABC123	2/16/2017	\$150	
3	196-X	XYZ100	4/3/2017	\$50		196-X	XYZ100	4/3/2017	\$50	
4	196-X	XYZ100	4/3/2017	\$50		327-A	ABC123	4/3/2017	\$150	
5	327-A	ABC123	4/3/2017	\$150		485-ABC	ABC123	3/7/2017	\$150	
6	327-A	ABC123	4/3/2017	\$150		511-XYZ	XYZ100	1/28/2017	\$50	
7	485-ABC	ABC123	3/7/2017	\$150		585-ABC	XYZ100	3/7/2017	\$50	
8	485-ABC	ABC123	3/7/2017	\$150						
9	511-XYZ	XYZ100	1/28/2017	\$50						
10	585-ABC	XYZ100	3/7/2017	\$50						
11	585-ABC	XYZ100	3/7/2017	\$50						
12										
13										

And it's as easy as that!

#36. Numbers Starting With Zero

This tip explains how to deal with leading zeros in your data. You can use an apostrophe (') in front of numbers starting with zero to convert them to text and keep any leading zeros.

Have you ever found yourself entering numbers into Google Sheets that begin with 0? Maybe zip codes or invoice numbers or product part numbers.

You've probably been frustrated when leading zeros disappear.

Thankfully there's an easy trick to keep that leading zero. Simply precede the number with an apostrophe (') to keep the number exactly as it is.

Note: It changes the number into a text value (so you can't do math with it), but since we're talking about zip numbers or invoice numbers etc. this is acceptable.

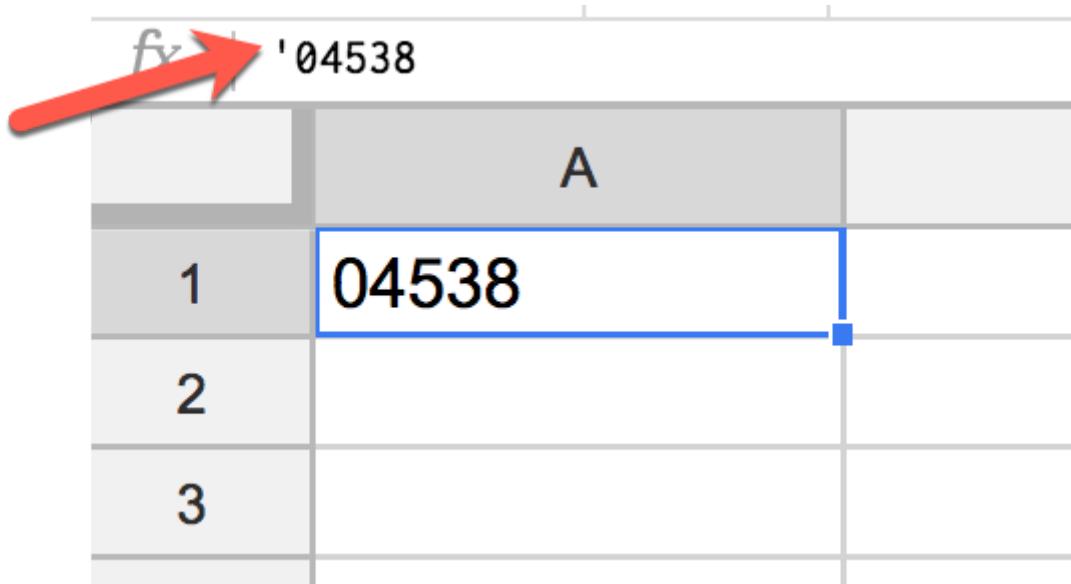
For example, consider the zip code for Boothbay Harbor in Maine, which has a zip code of 04538 (incidentally, this was the most northerly point of a [bike tour I did with my brother in 2014](#)).

Type that zip code into Google Sheets and this is what happens:

	fx A	B
1	4538	
2		
3		
4		

Hmm, that's no good.

Now type it with the apostrophe in front of the number and you'll see the difference (the apostrophe does not show up in the cell display):



A screenshot of a Google Sheets spreadsheet. The formula bar at the top shows the formula '=04538'. A large red arrow points from the left towards the formula bar. Below the formula bar is a table with three rows and three columns. The first column contains the numbers 1, 2, and 3. The second column contains the text '04538' in bold black font, which is highlighted with a blue selection border. The third column contains the letter 'A'. The cell containing '04538' is aligned to the left, while the other cells are aligned to the center.

1	04538	A
2		
3		

You'll notice that the zip code is now left-aligned because it's now stored as text rather than a number.

(Note: for true numeric values with leading zeros that you want to keep, you'll want to use Custom Number Formatting. See Tip #59 for more details about Custom Number Formatting.)

#37. Vlookup Formula Challenge

This tip is a formula challenge walkthrough. We'll look at how you can use array formulas to build "virtual" tables nested inside other formulas, like a Vlookup. You'll learn how to create new data tables dynamically to nest as ranges inside other functions.

Problem we want to solve:

	A	B	C	D		G	H	I
1	First Name	Last Name	Payment Date	Amount, \$		Name	Lookup Amount	
2	Jess	Smith	1/3/2014	\$892				
3	Hugo	Barnard	1/4/2014	\$854				
4	Victor	Jones	1/4/2014	\$567				
5	Bob	Davis	1/5/2014	\$383				
6	Jeff	Miller	1/7/2014	\$923		Bob Davis	?	
7	Sarah	Wilson	1/7/2014	\$580				
8	Steve	Johnson	1/8/2014	\$563				
9	Sue	Miller	1/9/2014	\$135				
10								
11								
12								
13								

We have a data table and we want to search for "Bob Davis" using Vlookup and return the amount associated with him.

The problem is that in the data table, names are split across two columns, First Name and Second Name. So a standard Vlookup isn't possible at the moment.

Solution:

There's an easy solution: create a quick helper column to combine the first and last names into a full name and use this as the search column.

But what about doing it without creating a new column in the data table?

The trick is to create a new table dynamically, which has the helper column. Let's see it in three steps:

Step 1. Create the full name column

First we need to generate the array of full names using this formula:

```
=ArrayFormula(A2:A9&" "&B2:B9)
```

The " " adds a space between the first and last names. The output of this single formula is an array of full names:

	G	H	I	J
Name	Lookup Amount			
Bob Davis	<pre>=ArrayFormula(A2:A9&" "&B2:B9)</pre>			
	Hugo Barnard			
	Victor Jones			
	Bob Davis			
	Jeff Miller			
	Sarah Wilson			
	Steve Johnson			
	Sue Miller			

(This is an Array formula. You enter the formula and then hit **Ctrl + Shift + Enter**, or **Cmd + Shift + Enter** (Mac) to add the **ArrayFormula** designation.)

Step 2. Add the other columns from the original table

In this step we build the new search table by adding the other columns from the original table that we want to search, using this formula:

```
=ArrayFormula( { A2:A9&" "&B2:B9 , C2:D9 } )
```

The curly braces { ... } combine arrays. Using a comma (,) between curly brace arrays treats them as columns next to each other, which is what we want, as you can see in the image:

	G	H	I	J	
Name	Lookup Amount				
Bob Davis	<code>=ArrayFormula({ A2:A9&" "&B2:B9 , C2:D9 })</code>				
	Hugo Barnard	1/4/2014		\$854	
	Victor Jones	1/4/2014		\$567	
	Bob Davis	1/5/2014		\$383	
	Jeff Miller	1/7/2014		\$923	
	Sarah Wilson	1/7/2014		\$580	
	Steve Johnson	1/8/2014		\$563	
	Sue Miller	1/9/2014		\$135	

This array formula combines columns A and B into a full name column and then adds back columns C and D to create a new table in H2:J9.

Step 3. Perform the Vlookup

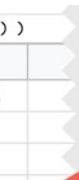
Now we have created the new table, we simply nest it as the range input in a standard Vlookup, with this formula:

```
=ArrayFormula( VLOOKUP( G2 , { A2:A9&" "&B2:B9 , C2:D9 } , 3 , false ) )
```

which gives the desired output of \$383:

fx	<code>=ArrayFormula(VLOOKUP(G2 , { A2:A9&" "&B2:B9 , C2:D9 } , 3 , false))</code>				
	A	B	C	D	
1	First Name	Last Name	Payment Date	Amount, \$	
2	Jess	Smith	1/3/2014	\$892	
3	Hugo	Barnard	1/4/2014	\$854	
4	Victor	Jones	1/4/2014	\$567	
5	Bob	Davis	1/5/2014	\$383	
6	Jeff	Miller	1/7/2014	\$923	
7	Sarah	Wilson	1/7/2014	\$580	
8	Steve	Johnson	1/8/2014	\$563	
9	Sue	Miller	1/9/2014	\$135	
10					
11					
12					

	G	H	I
Name	Lookup Amount		
Bob Davis	\$383		



Nice!

Note: if you want to use this Vlookup on another row to look up another full name, you'll want to lock those range references to avoid getting errors.

This concept of creating "virtual" tables that you nest inside of other formulas is super useful. You can also use the Filter function, the Query function etc. to create nested tables.

Further Reading

For more information on this Vlookup, see this post: [Vlookup with Multiple Criteria in Google Sheets](#).

For more information on the Array Formula, see this post: [How do array formulas work in Google Sheets?](#)

#38. Text Trickery Part I

This tip was prompted by a recent question from a reader:

How do I make a diagonal line to split a cell, so that I can enter text into two triangular subdivisions?

You can't split a cell explicitly, but you can use some text trickery to achieve this effect. 🐍

Here's how to use Text Rotation to customize the look of your tables in Google Sheets.

Here's the example of a cell with a diagonal line to clearly show your row and column heading labels in a single cell:

	A	B	C	D	E
1	Regions Sales	Q1	Q2	Q3	Q4
2	North	\$127	\$142	\$182	\$133
3	South	\$130	\$112	\$196	\$117
4	East	\$117	\$103	\$165	\$194
5	West	\$164	\$136	\$175	\$171
6					
7					
8					

To achieve this effect above, use the CHAR function and rotate the text.

CHAR(10) adds a line break.

CHAR(8213) adds a horizontal bar.

Combine these into this formula:

```
= "Sales" & char(10) & char(8213) & char(8213) & char(8213) & char(8213) &
char(8213) & char(8213) & char(10) & "Regions"
```

where you can add as many CHAR(8213) as you require to create the horizontal line.

Center align the cell and then, under the Format menu, rotate the text downwards:

The screenshot shows a Google Sheets interface with a formula bar containing the formula `= "Sales" & char(10) & char(8213) & char(8213) & char(8213) & char(8213) & char(8213) & char(8213) & char(10) & "Regions"`. Below the formula bar, a table is displayed with columns A and B. Row 1 contains the formula in cell A1 and the word "Regions" in cell B1. Row 2 contains "North" in cell A2 and "Q1" in cell B2. Rows 3 through 5 contain "South", "East", and "West" respectively. The "Format" menu is open, and the "Text rotation" submenu is selected, showing various options for text orientation.

You can choose a custom degree of rotation to get the best effect.

Finally, the formula will look something like this:

fx

	A	B	C	D
1				E
2				
3				
4				
5				

#39. Text Trickery Part II

When I shared the first part of this tip (see #38) in an email, I had tons of responses and readers suggested some great, often simpler, ways to do it.

Here's the example from the last tip, with the diagonal line to clearly show your row and column heading labels in a single cell:

	A	B	C	D	E	
1	Regions Sales	Q1	Q2	Q3	Q4	
2	North	\$127	\$142	\$182	\$133	
3	South	\$130	\$112	\$196	\$117	
4	East	\$117	\$103	\$165	\$194	
5	West	\$164	\$136	\$175	\$171	
6						
7						
8						

Here are some of the great suggestions that I received.

From Melissa

I wouldn't recommend doing it in the sample case you gave, because you've just made the column headings a mess for pretty much anything else, like pivot tables or charts. The labels in column A really need their own proper column heading.

This is a great point, Melissa! I should have included this caveat in last week's email. It's really only suitable for a presentation table where you don't need to do anything further with the data. 

From Doug

I played around with it and was able to get a solid line by using an n-dash, which I typed with option-dash on my Mac. That's CHAR(8211) for the way you did it.

A screenshot of a Google Sheets cell. The cell contains the letter 'A'. Two text labels, "Regions" and "Sales", are rotated diagonally and positioned near each other. A solid black line connects the top of the "Regions" label to the top of the "Sales" label. The cell has a grid background with rows labeled 1 and 2, and columns labeled A.

Nice! I agree, the solid line looks better

From Brian

Tighten that up using formulas from a previous lesson!

```
= "Sales" & CHAR(10) & REPT(CHAR(8213), 7) & CHAR(10) & "Regions"
```

Brian has used the REPT function to repeat the CHAR function 7 times, instead of typing it in multiple times. Neat! Thank you

From JC

There is a much simpler way for this one.

- First, just type your text in the cell, using Ctrl-Enter at the end of each title to put the next title on another line in the same cell. For the middle line, see my next step.
- Second, for the dividing line, use a series of Em-dashes (–) or En-dashes (‐), which are shorter. You can enter these directly from the keyboard by holding down the Alt key while you type from the 10-key pad the numbers 0151 for Em-dash, or 0150 for En-dash.
- After this, use rotation as in your example. Choosing “-45° Angle” probably works best.

Yes! Simpler is better 

From Willem

The way I do this using CONTROL + ENTER for the line break and a few of this character: – to form a line and then another CONTROL + ENTER

After that of course rotate.

I found the – character on <https://www.copypastecharacter.com/>

All characters you find there you can paste wherever in your docs and sheets!

Yes! Simpler is better 

From Martin

I think this is much simpler.

```
= "Row"
" & repet ("-", 20) &
    Col"
```

Better still would be a formula to replace the "20" above with something like $\text{sqrt}(\text{sq}(\text{rowheight}())+\text{sq}(\text{colwidth}()))$ but I can't find anything that will return either column width or row height.

Oooh! Interesting idea. 

Create a function to determine the required number of dashes based on the width and height of the cell. Not possible with standard formulas but it sounds like a great contender for a custom formula using the [getColumnWidth](#) and [getRowHeight](#) methods in Apps Script...

Thank you to everyone who responded! I enjoyed reading all these suggestions and love that people have shared better, simpler ways to do things.

#40. Text Trickery Part I

Ciao, come stai?

If you deal with multiple languages in your Google Sheets, then this tip is for you. Here's how to combine the GOOGLETRANSLATE function and the DETECTLANGUAGE function to build a translation engine in your Google Sheet.

Google Sheets have an amazingly diverse and powerful set of functions. There are functions for everything from engineering to finance, from statistics to web scraping, from images to hyperlinks, and there's even a function to translate foreign languages!

GOOGLETRANSLATE translates text from one language into another. It takes three arguments: 1) the text to translate, 2) the language of the original text (as a 2 letter code), and 3) the language you want to translate into (as a 2 letter code).

You might have a function like this:

```
=GOOGLETRANSLATE("Ciao, come stai?", "it", "en")
```

which translates from Italian to English and gives the answer "Hello how are you?". You can also reference text in other cells, like this:

```
=GOOGLETRANSLATE(A1, "it", "en")
```

The DETECTLANGUAGE function takes text (or a cell with text in) and determines the language. It returns the two letter code for the language detected. So:

```
=DETECTLANGUAGE("Ciao, come stai?")
```

returns the answer "**it**", since the function has detected Italian as the language.

So, combine these two functions to create a powerful translation engine in your Google Sheets!

```
=GOOGLETRANSLATE(A1, DETECTLANGUAGE(A1), "en")
```

This will translate whatever text is in cell A1, whether it's Italian, Spanish, Russian, Arabic, Chinese, or hundreds of other languages.

	A	B	C
1	Ciao, come stai?	Hello how are you?	
2	你好，你好吗？	Hello how are you?	
3	مرحبا كيف حالك؟	Hello how are you?	
4	Hallo hoe gaat het?	Hello how are you?	
5	γεια πώς είσαι?	Hi how are you?	
6	halló, hvernig ertu?	hello, how are you?	
7	Привет, как ты?	Hello how are you?	
8	¿Hola como estas?	Hi how are you?	
9			

Amazing, huh? To have that much power at your fingertips!

Formula Challenge #1

Here's a formula challenge for you to try. I'll share a solution in the next tip.

Start with a straightforward IMAGE function in cell A1, like this:

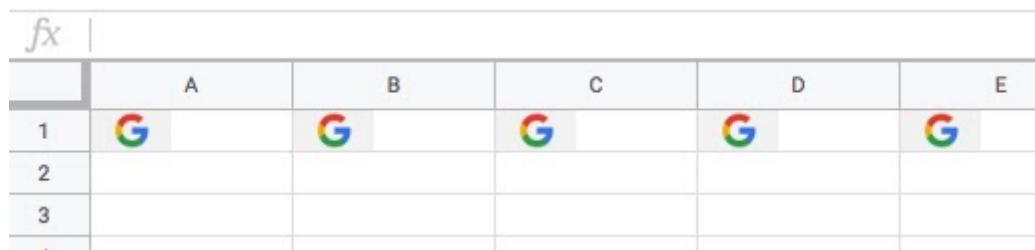
```
=IMAGE("https://blog.hubspot.com/hubfs/image8-2.jpg")
```

	A	B	C	D
1				
2				
3				

(You can use whatever image you like.)

Your Challenge:

Modify the formula in cell A1 only to repeat the image across multiple columns (say 5 as in this example), so it looks like this:



	A	B	C	D	E
1	Google logo				
2					
3					
.					

Rules: You're only allowed to use a single formula in cell A1. 😊

#41. ROW/COLUMN Functions

(Editor's note: since this tip was written (March 2019), Google released the SEQUENCE function which makes vector creation even easier. See tip #72)

Here's how you can use the ROW and COLUMN functions to create lists of numbers to use in Array Formulas.

ROW returns the row number of a specific cell. COLUMN returns the column number of a specific cell.

To create a row or column of numbers, use the ROW or COLUMN function inside an Array Formula:

```
=ArrayFormula(ROW(1:5))
```

which gives a column output like this:

	A	B	C
1	1		
2	2		
3	3		
4	4		
5	5		
6			
7			

The column function version is

```
=ArrayFormula(COLUMN(A:E))
```

which gives a row output like this:

	<i>fx</i>	=ArrayFormula(COLUMN(A:E))				
	A	B	C	D	E	F
1	1	2	3	4	5	
2						
3						
4						
5						

If you work with Array Formulas, then you'll find these sorts of helper functions super useful.

For example, you can use these number vectors to help with sorting data.

Ok, now we know how to create these vectors, let's see how we can use them to solve the formula challenge from the previous tip.

Formula Challenge #1 Solution

The Challenge:

Starting with a straightforward IMAGE function in cell A1, like this:

```
=IMAGE ("https://blog.hubspot.com/hubfs/image8-2.jpg")
```

The challenge was to modify the formula in cell A1 to repeat the image across multiple columns (say 5 as in this example), so it looked like this:

	<i>fx</i>					
	A	B	C	D	E	
1						
2						
3						
-						

The problem is that the IMAGE function can't be nested inside a REPT function, so you have to get a bit more creative.

Solution 1: using ROW or COLUMN counts:

```
=ArrayFormula(IF(COLUMN(A:E), IMAGE("https://blog.hubspot.com/hubfs/image8-2.jpg")))
```

Using this week's tip, the array formula COLUMN(A:E) will output an array 1 to 5. The IF statement treats the numbers as TRUE values, so prints out the image 5 times. For brevity, we can omit the FALSE value, since we don't call it.

Solution 2: using REPT inside the IMAGE formula!

Thanks to Ryan K., Federico J. and Goran K. for this ingenious solution!

As I mentioned, the REPT function doesn't work with the IMAGE function, but flip it round, with the REPT inside the IMAGE function, and it does work!

Start with this formula in cell A1, which creates a single string of joined URLs, with a pipe (|) delimiter between them:

```
=ArrayFormula(REPT("https://blog.hubspot.com/hubfs/image8-2.jpg" & " | ", 5))
```

Now, split these into an array of 5 separate URLs:

```
=ArrayFormula(SPLIT(REPT("https://blog.hubspot.com/hubfs/image8-2.jpg" & " | ", 5), " | "))
```

Finally, wrap this with the IMAGE function to get the five images in a row:

```
=ArrayFormula(IMAGE(SPLIT(REPT("https://blog.hubspot.com/hubfs/image8-2.jpg" & " | ", 5), " | ")))
```

What I like about this solution is that you could put the number 5 into a different cell and reference it, so that you can easily change how many times the image is repeated. You could even embed another formula to calculate how many times to repeat the image ;)

#42. The Endless Possibilities of Custom Functions

This tip will demonstrate how to easily insert company logos into your Google Sheets using a custom function combined with the IMAGE function. You'll learn how to create Custom Functions using Apps Script to solve unique problems and save time.

The company Clearbit provides a [free logo service API](#), which will show a company's logo based on only their website domain.

To use it, you append a website domain to their logo url:

```
https://logo.clearbit.com/{domain}
```

and it will return an image of that company's logo.

By passing that url into the IMAGE function you can display the company logo in your Google Sheet. For example this will display the Google logo:

```
=IMAGE("https://logo.clearbit.com/google.com")
```

or the Nike logo:

```
=IMAGE("https://logo.clearbit.com/nike.com")
```

That's pretty cool!

But we can make it more user friendly by creating a custom function -- let's call it LOGO -- to create that Clearbit url for us. Custom functions are little [Apps Script](#) programs that perform custom actions or calculations. With some exceptions, you can literally create your own Google Sheet functions to do whatever calculations you want!

In your Google Sheet, under **Tools > Script Editor** clear out the existing code and paste in the following code:

```
function LOGO(input) {
  return 'https://logo.clearbit.com/' + input;
}
```

All this does is combine the Clearbit URL and the domain name, to save you having to type it out every time.

Save the project.

Back in your Google Sheet, add a domain in cell A1 and then call the custom function:

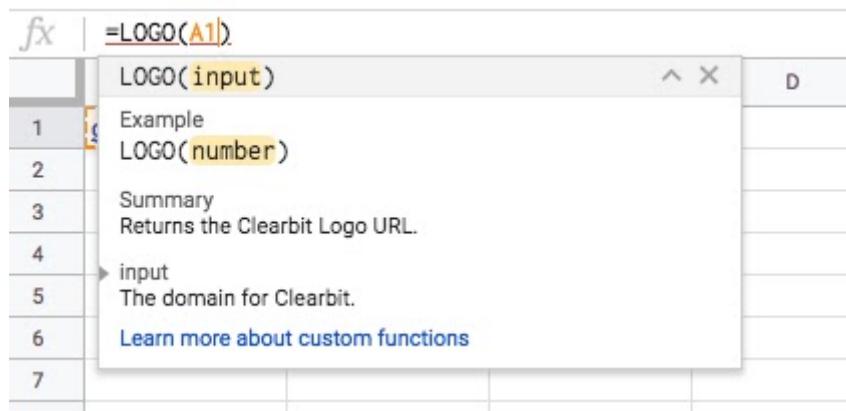
```
=LOGO(A1)
```

Wrap it with the IMAGE function and ta-da! You'll have your company logos.

```
=IMAGE(LOGO(A1))
```

	A	B	C
1	google.com		
2	pwc.com		
3	bmw.com		
4	nike.com		
5			
6			

If you want to explore further, you can add a `@customfunction` comment between the stars `/**...*/` to get the auto-complete box:



The code to add this auto-complete box is:

```
/**  
 * Returns the Clearbit Logo URL.  
 *  
 * @param {number} input The domain for Clearbit.  
 * @return The Clearbit API url.  
 * @customfunction  
 */  
function LOGO(input) {  
  return 'https://logo.clearbit.com/' + input;  
}
```

#43. The SUBTOTAL Function

Here's how to use the SUBTOTAL function with filtered datasets to display a total for the shown values only.

The SUBTOTAL function is a powerful aggregation function. It returns a subtotal for a range of data, based on an aggregation function. In its most common form, which I'll show below, it's used to show a sum of values.

Suppose you have a dataset of values with filters added, like so:

	A	B	C	D	E	F	G	H
1	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %	Gross Commission Income	
2	1	Apartment	Referral	Buyer	\$328,484	3.00%	\$9,855	
3	2	House	Previous client	Buyer	\$1,145,078	3.00%	\$34,352	
4	3	Apartment	Previous client	Buyer	\$587,737	3.00%	\$17,632	
5	4	Apartment	PPC	Seller	\$522,867	3.00%	\$15,686	
6	5	Apartment	Website	Seller	\$321,706	3.00%	\$9,651	
7	6	Condo	Website	Seller	\$486,653	3.00%	\$14,600	
8	7	Apartment	Referral	Seller	\$482,425	3.00%	\$14,473	
9	8	Townhouse	Referral	Buyer	\$931,004	3.00%	\$27,930	
10	9	Condo	Referral	Seller	\$321,095	3.00%	\$9,633	
11	10	Townhouse	Previous client	Buyer	\$548,455	3.00%	\$16,454	
12	11	Apartment	Referral	Buyer	\$376,895	3.00%	\$11,307	
13	12	Condo	Previous client	Buyer	\$709,642	3.00%	\$21,289	
14	13	Apartment	PPC	Buyer	\$277,876	3.00%	\$8,336	
15	14	Townhouse	Referral	Seller	\$838,370	3.00%	\$25,151	

When you work with this dataset, maybe you filter on different categories during your research, e.g. all the Referral clients or all the Buyer deals.

Wouldn't it be nice to add a total value that changes to match whatever filter you had applied?

You can do this by adding a SUBTOTAL function into row 1 above the values columns:

```
=SUBTOTAL(9, E3:E)
```

The first argument of the SUBTOTAL function (9) specifies the type of aggregation you want to apply. In this case, the 9 tells the function to SUM the values.

Now, when you filter on Referral for example, the SUBTOTAL function will update to show just the total for the Referrals:

	A	B	C	D	E	F
1					\$4,121,411	
2	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %
3	1	Apartment	Referral	Buyer	\$328,484	3.00%
9	7	Apartment	Referral	Seller	\$482,425	3.00%
10	8	Townhouse	Referral	Buyer	\$931,004	3.00%
11	9	Condo	Referral	Seller	\$321,095	3.00%
13	11	Apartment	Referral	Buyer	\$376,895	3.00%
16	14	Townhouse	Referral	Seller	\$838,370	3.00%
17	15	Apartment	Referral	Buyer	\$417,507	3.00%
18	16	Studio	Referral	Buyer	\$198,475	3.00%
19	17	Condo	Referral	Buyer	\$227,156	3.00%
23						
24						
25						

The SUBTOTAL function in cell E1 shows a total of \$4.1m for Referrals only, out of the full \$10m or so for the entire dataset.

Note: You can change the aggregation function by changing the first number in the SUBTOTAL function. For example, instead of 9 (SUM), the number 1 calculates AVERAGE or number 2 calculates COUNT. For a full list, have a look at the [documentation](#).

#44. Year In Progress Sparkline

Let's take a look at Sparklines, which are miniature charts that exist inside a single cell. I'll show you how to use the SPARKLINE function to add miniature charts to your cells in Google Sheets.

In tip #32, I shared a formula that calculates how far through the year we are:

```
=YEARFRAC( DATE(YEAR(TODAY()),1,1) , TODAY() )
```

Taking inspiration from this Twitter account, which tweets out yearly progress each day, let's re-create a similar chart in our Google Sheet:



Start with the Year Fraction formula above in cell A1 and wrap it with curly brackets {} to create an array and add a second value of 1:

```
={ YEARFRAC(DATE(YEAR(TODAY()),1,1),TODAY()) , 1 }
```

	<i>fx</i>	= {YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY()),1}	
	A	B	C
1	0.2444444444	1	
2			
3			

This is done to provide a second value to the Sparkline formula, so that it can show the percentage of the year remaining as well.

Wrap this entire formula with the Sparkline function and, in the second argument, specify the type as "bar chart", inside the curly brackets:

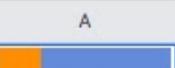
```
=SPARKLINE( { YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY() ) , 1 } , {  
"charttype","bar" } )
```

Next, add another option to the Sparkline to set the max value to 1, which ensures that the sparkline shows the 24% correctly:

```
=SPARKLINE( { YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY() ) , 1 } , {  
"charttype","bar" ; "max", 1 } )
```

Use the semicolon to separate options in your Sparkline function.

Your output now will look like this:

<i>fx</i>	=SPARKLINE({YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY()),1}, {"charttype","bar";"max",1})
	A B C D E F G
1	 A blue bar representing 24% of the year, starting from black at the left and transitioning to blue. The bar is approximately 24 units long out of a maximum of 100 units.
2	
3	
4	

You can replace the default colors by specifying options for "color1" and "color2", like this:

```
=SPARKLINE( { YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY() ) , 1 } , {  
"charttype","bar" ; "max", 1 ; "color1" , "black" ; "color2" , "#a9a9a9"  
} )
```

Finally, you can add a title and the % formula to show the complete picture:

<i>fx</i>	=SPARKLINE({YEARFRAC(DATE(YEAR(TODAY()),1,1) , TODAY()),1}, {"charttype","bar";"max",1;"color1","black";"color2","#a9a9a9"})
	A B C D
1	Year Progress  A bar divided into two segments: black on the left and grey on the right. The total length of the bar is 24 units. To its right, the text "24%" is displayed.
2	
3	
4	

This will automatically update throughout the year, to periodically remind you how quickly time is passing 😊

For more information about Sparklines and other examples, including mini-line and -column charts, check out this article: [Everything you ever wanted to know about Sparklines in Google Sheets.](#)

For European Google Sheets users, your syntax is a little different so check out [this article](#) which highlights the differences.

#45. Find the Most Frequent Word

You can use the MODE function to find the most commonly occurring value in a dataset.

The MODE function takes a range of numbers for an input and finds the most commonly occurring value.

However, what happens if you have a range of text values and what to find the most frequent?

For example, what's the most frequent U.S. State in a set of survey responses. Well you can still use the MODE function but you need to add some other functions into the mix to make it work.

Imagine you have this dataset and you want to know the most frequent State:

	A	B
1	New Mexico	
2	Texas	
3	Florida	
4	Texas	
5	Oklahoma	
6	Florida	
7	Arizona	
8	Texas	
9	California	
10	California	
11	Arizona	
12	California	

As per usual, let's build the formula in steps.

In cell B1, add this formula:

```
=ArrayFormula(MATCH(A1:A20,A1:A20,0))
```

which outputs an array of the position of the first occurrence of the words in column A. In the image above, you'll have a 2 next to every occurrence of Texas for example, because the first time it occurred was in position 2.

Now, wrap it with the MODE function to find the most frequently occurring position:

```
=ArrayFormula(MODE(MATCH(A1:A20,A1:A20,0)))
```

MODE returns the most frequent value, which in this case is the first position of the most frequently occurring text value.

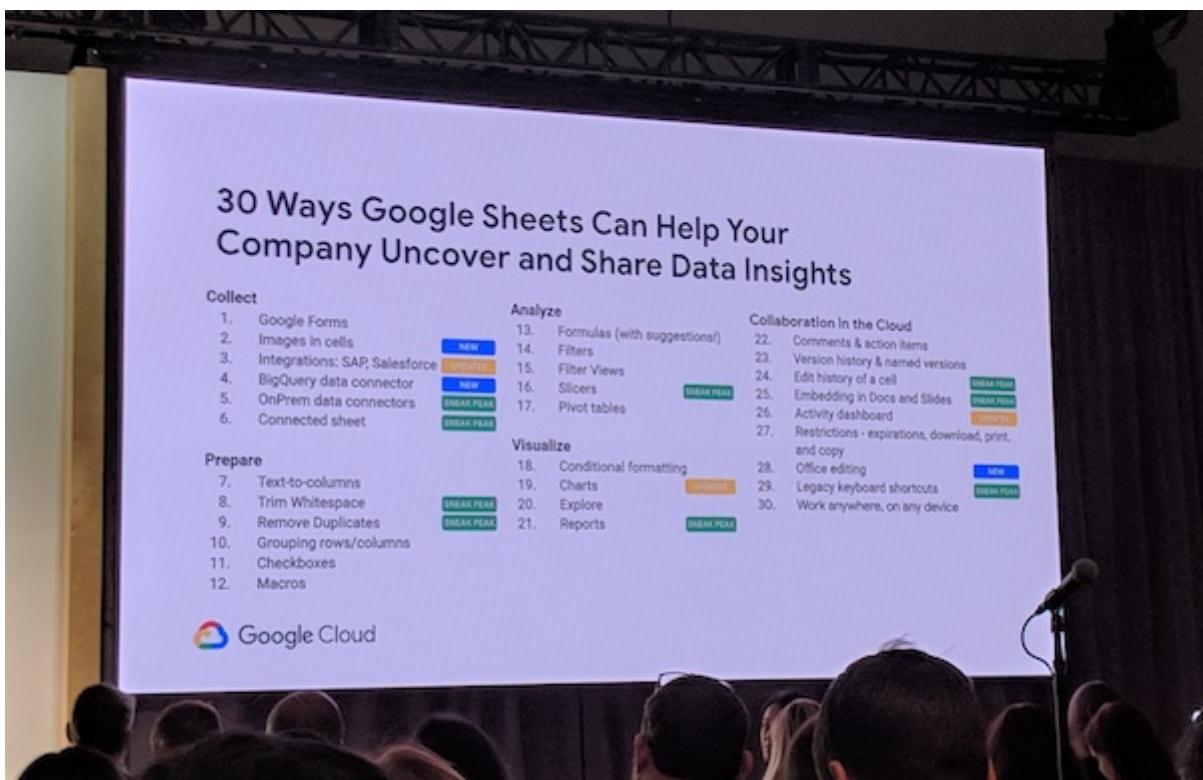
You can then retrieve that by adding the INDEX function like this:

```
=ArrayFormula(INDEX(A1:A20,MODE(MATCH(A1:A20,A1:A20,0))))
```

This will give the output Texas in this specific example. Nice!

#46. Google Sheets Features Roadmap

The [original tip #46](#) was a list of features coming soon to Google Sheets announced at Google Next 19 conference.



Many of those features have now been released (although we're still waiting for [Connected Sheets general release](#), which is going to be HUGE. Literally.)

Instead, I thought I'd share a couple of useful places to keep track of the Google Sheets roadmap:

[What's New in G Suite](#)

[List of Recent Releases](#)

[G Suite Updates Blog](#)

#47. Macro to Show/Hide Tabs

You can create a simple tool to focus on your active tab, with just a few lines of Apps Script.

Imagine this: Jess works for a large real-estate brokerage that runs on G Suite. She's a sales team lead and manages a large Google Sheet.

It's become a sprawling monster, and she finds it hard to focus on the task at hand because the Sheet is so complex.

She likes to hide all the tabs in her Sheet, except the one she's working in. It helps keep her focussed.

Doing this manually is tedious, so she wrote a few lines of Apps Script to automate the process. Check it out:

	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %	Gross Commission Income
4	1	Apartment	Referral	Buyer	\$328,484	3.00%	\$9,855
5	2	House	Previous client	Buyer	\$1,145,078	3.00%	\$34,352
6	3	Apartment	Previous client	Buyer	\$587,737	3.00%	\$17,632
7	4	Apartment	PPC	Seller	\$522,867	3.00%	\$15,686
8	5	Apartment	Website	Seller	\$321,706	3.00%	\$9,651
9	6	Condo	Website	Seller	\$486,653	3.00%	\$14,600
10	7	Apartment	Referral	Seller	\$482,425	3.00%	\$14,473
11	8	Townhouse	Referral	Buyer	\$931,004	3.00%	\$27,930
12	9	Condo	Referral	Seller	\$321,095	3.00%	\$9,633
13	10	Townhouse	Previous client	Buyer	\$548,455	3.00%	\$16,454
14	11	Apartment	Referral	Buyer	\$376,895	3.00%	\$11,307
15	12	Condo	Previous client	Buyer	\$709,642	3.00%	\$21,289
16	13	Apartment	PPC	Buyer	\$277,876	3.00%	\$8,336
17	14	Townhouse	Referral	Seller	\$838,370	3.00%	\$25,151
18	15	Apartment	Referral	Buyer	\$417,507	3.00%	\$12,525
19	16	Studio	Referral	Buyer	\$198,475	3.00%	\$5,954
20	17	Condo	Referral	Buyer	\$227,156	3.00%	\$6,815
21	18	Condo	Website	Buyer	\$248,226	3.00%	\$7,447
22	19	Studio	PPC	Buyer	\$249,577	3.00%	\$7,487

The tabs at the bottom of the sheet are: Budget Summary, Budget Detail, Projections, Sales (highlighted in green), Mortgage Payments, Q1 goals, and Q2.

Much better. Now she feels like she's working with just one tab, so she can focus all her energy on it.

But what about when she wants to start using the other tabs again?

No problem, she adds another few lines of Apps Script to automatically unhide the tabs again.

Jess then adds a custom menu (only 8 more lines of code!) so that she can run the script from her Google Sheet directly (as shown in the GIF images above).

In total, it's about 25 lines of code and takes a couple of minutes to put together.

Want to set this up yourself?

From your Google Sheet, open the script editor: **Tools > Script editor**

Paste in the following code:

```
// menu in Google Sheets
function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('Focus')
    .addItem('Focus', 'focus')
    .addItem('Unfocus', 'unfocus')
    .addToUi();
}

// function hides all Sheets except the active one
function focus() {
  var sheets = SpreadsheetApp.getActiveSpreadsheet().getSheets();
  sheets.forEach(function(sheet) {
    if (sheet.getName() !== SpreadsheetApp.getActiveSheet().getName())
      sheet.hideSheet();
  });
}

// function unhides all the Sheets except the active one
```

```
function unfocus() {  
  var sheets = SpreadsheetApp.getActiveSpreadsheet().getSheets();  
  sheets.forEach(function(sheet) {  
    sheet.showSheet();  
  });  
}
```

Hit save and give your project a name.

Still in the Script Editor, run the onOpen function: **Run > Run function > onOpen**

Click through the permissions popup to allow your Script to run.

(You may see "This app isn't verified", in which case click on "Advanced" and then "Go to [name] project - unsafe" and then click "Allow". These are extra security steps introduced by Google for non-verified scripts. Since we're the author of this script though, it's safe to run. For more info on [permissions click here](#)).

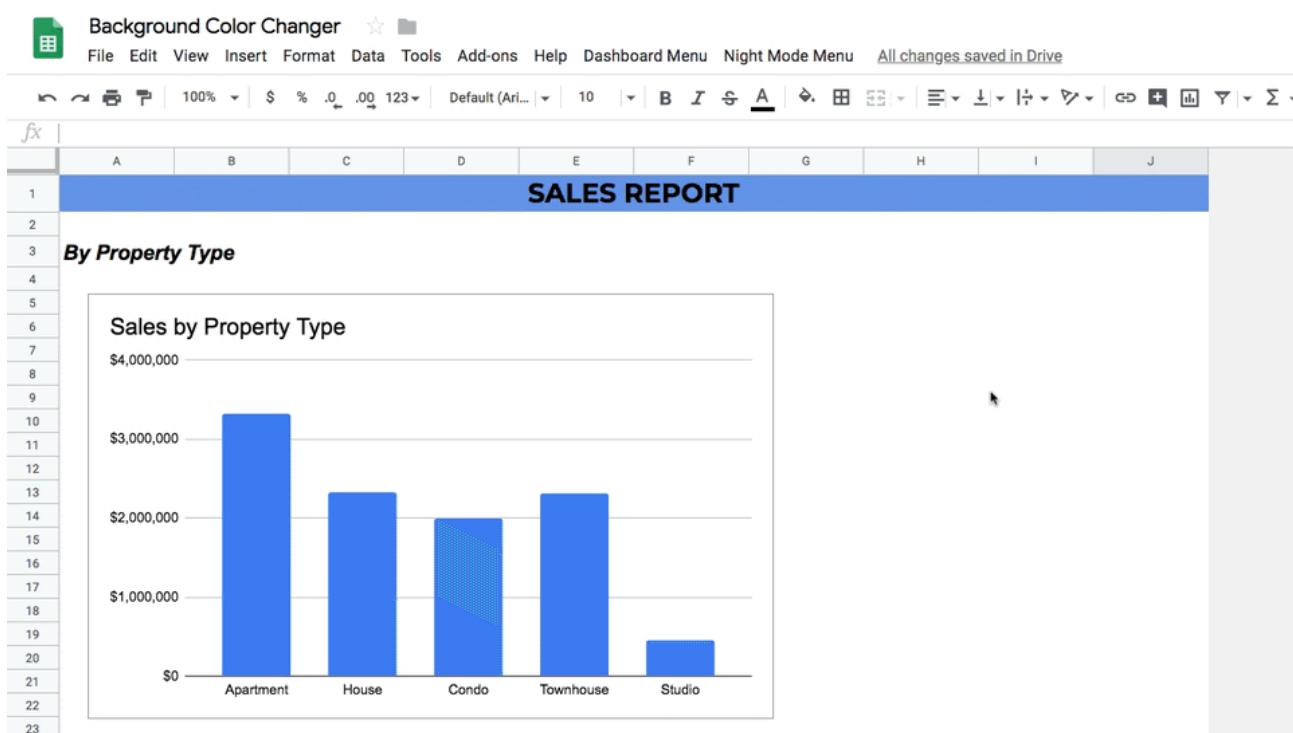
Voila! You'll have a new menu in your Google Sheet where you can run the focus/unfocus tool. (You'll need multiple tabs to see it in action!)

It's pretty cool what you can do with a little Apps Script!

#48. Night Mode, Anyone?

Check out this fun Apps Script program to switch your Sheets from "day" to "night" mode. Here's how to change the background color of your Sheets with just a few lines of Apps Script.

Granted this may not have real-world, practical implications, but it's a fun little program that demonstrates how to interact with Sheets from Apps Script.



You can add a custom menu to run the script from your Google Sheet directly (as shown in the GIF images above).

In total, it's about 45 lines of code and takes a couple of minutes to put together.

The loop code may look complex if you're unfamiliar with Apps Script, but it's not hard to learn.

The script grabs all the rows of data and loops over each row in turn. With each row, it loops over all the cells and changes any white backgrounds to black, and any black backgrounds to white. Any text in the Sheet also gets flipped from black to white, or vice-versa.

Want to set this up yourself?

From your Google Sheet, open the script editor: **Tools > Script editor**

Paste in the following code:

```
// menu in Google Sheets

function onOpen() {
  var ui = SpreadsheetApp.getUi();
  ui.createMenu('Night Mode Menu')
    .addSubMenu(ui.createMenu('Choose theme...'))
    .addItem('Day theme', 'dayTheme')
    .addItem('Night theme', 'nightTheme'))
  .addToUi();
};

// set night theme

function nightTheme() {
  var color1 = '#000000';
  var color2 = '#ffffff';
  changeTheme(color1, color2);
}

// set day theme

function dayTheme() {
  var color1 = '#ffffff';
  var color2 = '#000000';
  changeTheme(color1, color2);
}

// change colors

function changeTheme(color1, color2) {
  var sheet = SpreadsheetAppgetActiveSheet();
  var cells =
sheet.getRange(1,1,sheet.getMaxRows(),sheet.getMaxColumns());
```

```

var backgrounds = cells.getBackgrounds();

backgrounds.forEach(function(row) {
    row.forEach(function(cell, i, array) {
        if (cell === "#ffffff") {
            array[i] = color1;
        }
        if (cell === "#000000") {
            array[i] = color1;
        }
    });
});

cells.setBackgrounds(backgrounds);
cells.setFontColor(color2);
}

```

Hit save and give your project a name.

Still in the Script Editor, run the onOpen function: **Run > Run function > onOpen**

Click through the permissions popup to allow your Script to run.

(You may see "This app isn't verified", in which case click on "Advanced" and then "Go to [name] project - unsafe" and then click "Allow". These are extra security steps introduced by Google for non-verified scripts. Since we're the author of this script though, it's safe to run.)

Voila! You'll have a new menu in your Google Sheet where you can change your background color.

Granted this only works in limited cases but it's a cool demonstration of what you can do with a little Apps Script!

#49. Create New Tabs Instantly

This tip is nice and short, but super handy! You can create new tabs in your Google Sheets instantly with **Shift + F11**.

Using shortcuts in spreadsheets dramatically speeds up your workflows and makes you a more efficient worker.

They're super easy to use, but it takes a little while to train your muscle memory!

#50. Images in Cells

Did you know you can add images directly to cells from the Insert menu?

Before this feature was released, the only way to add an image to a cell was to use the IMAGE() function. This is rather cumbersome and requires a URL for the image, which means the image has to be publicly visible online somewhere.

Alternatively, you can add floating images above the cells. This has the disadvantage that the images are not "locked" in place, so they don't feel part of the Sheet. Images sit on top of the Sheets row/column grid and do not move with the data.

Now though, you can insert images directly into cells. There's no requirement to use a URL, which means you can add images from your local drive.

Images inside cells move with the data. If you add rows, filter or sort the data, then the images will move too.

This is a great feature for adding company logos to your Google Sheets, or icons to your dashboards for example.

You find this image feature under the menu: **Insert > Images > Image in cell**

Copy of New Sheet star file

File Edit View Insert Format Data Tools Add-ons Help Last edit was on /

100% Row above Row below Column left Column right Cells and shift down Cells and shift right

Chart Image Drawing... Form...

Image in cell New
Image over cells

A screenshot of a Google Sheets spreadsheet titled "Copy of New Sheet". The spreadsheet has a single row with columns A through F. Cell A1 contains the letter "A". The "Insert" menu is open, showing options like "Row above", "Row below", "Column left", etc., followed by a separator line and "Image". A dropdown menu for "Image" is displayed, containing "Image in cell" (which is highlighted with a green border and labeled "New") and "Image over cells". The background shows the rest of the spreadsheet grid.

#51. Remove Duplicates

You can use the new Remove Duplicates feature under the Data Menu, instead of formulas or add-ons, to remove duplicate entries.

We all need to remove duplicate entries from our data from time-to-time. So it seems crazy to me that we had to wait this long to get a native remove duplicates feature.

It means you'll no longer have to wrestle with the UNIQUE function (unless you have reason to) or use a (paid!) add-on to do this simple task.

The new feature is super easy to use. You find this feature under the menu: **Data > Remove Duplicates**

The screenshot shows a Google Sheets interface with a data range selected containing two columns: 'First Name' and 'Last Name'. The 'Data' menu is open, and the 'Remove duplicates' option is highlighted with a green box and labeled 'New'.

	First Name	Last Name
1	Jess	Smith
2	Jeff	Miller
3	Sarah	Wilson
4	Steve	Johnson
5	Sue	Miller
6	Sarah	Wilson
7	Jeff	Miller
8		
9		
10		
11		
12		
13		
14		

The Data menu options shown are:

- Sort sheet by column A, A → Z
- Sort sheet by column A, Z → A
- Sort range by column A, A → Z
- Sort range by column A, Z → A
- Sort range...
- Create a filter
- Filter views...
- Data validation...
- Pivot table...
- Randomize range
- Named ranges...
- Protected sheets and ranges...
- Split text to columns...
- Remove duplicates** (highlighted)

When you click Remove Duplicates, you'll be prompted to choose which columns you want to check for duplicates.

You may want to remove duplicates where the rows entirely match, or you may wish to choose a specific column, such as an invoice number, regardless of what data is in the other columns.

B	C	D	E	F
Last Name	Payment Date	Amount, \$		
th	5/17/2019	\$	Remove duplicates X	
er	5/14/2019	\$		
son	5/13/2019	\$		
nson	5/9/2019	\$		
er	5/7/2019	\$		
son	5/13/2019	\$		
er	5/14/2019	\$		

Remove duplicates

8 rows and 4 columns selected

Data has header row

Columns to analyze

Select all

Column A - First Name

Column B - Last Name

Column C - Payment Date

Column D - Amount, \$

[Cancel](#)

[Remove duplicates](#) 

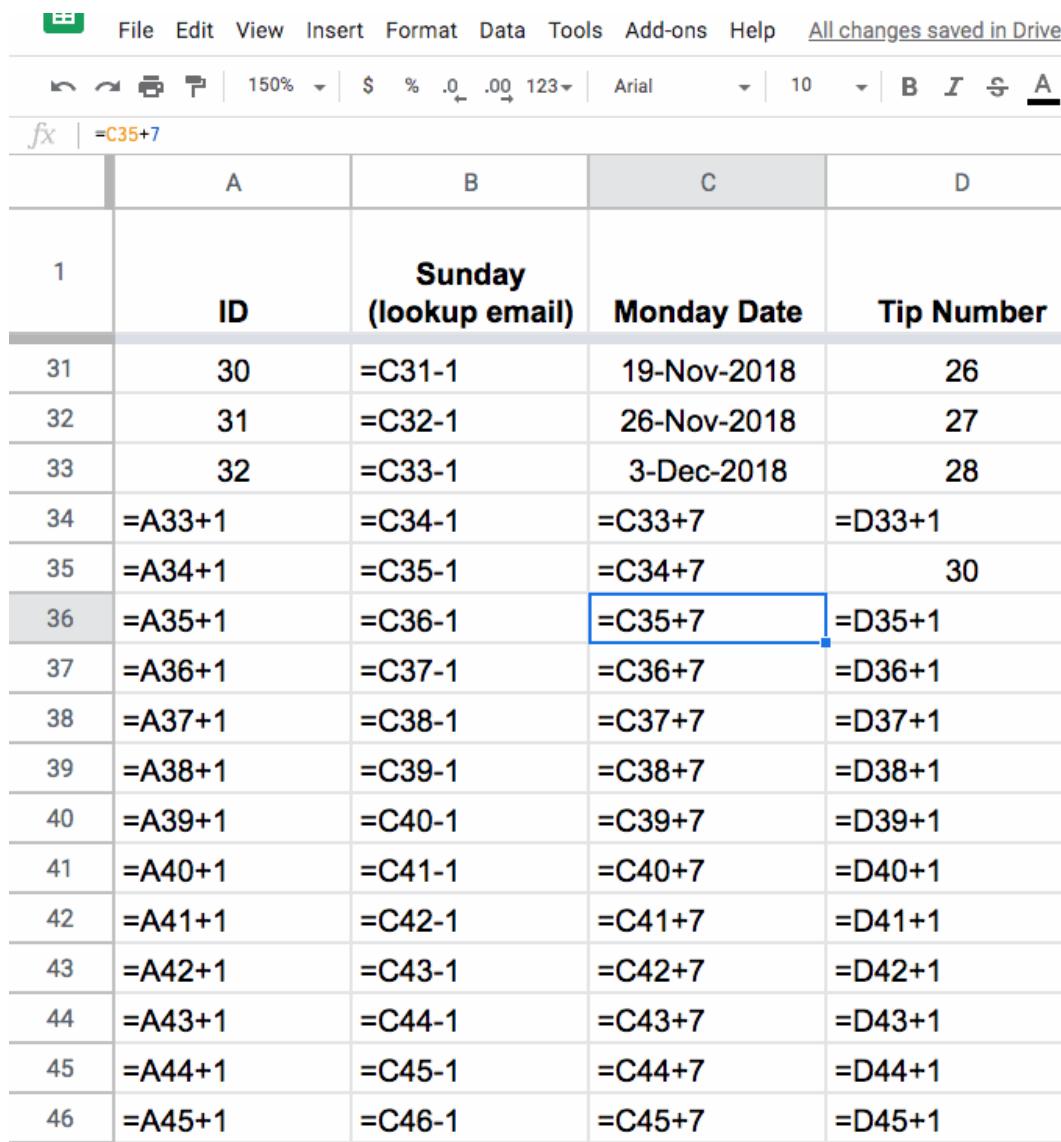
#52. Show Formulas

You can show all the formulas in your Sheet with the shortcut **Ctrl + `**

This is a really handy shortcut when you're trying to debug (fix) your Google Sheet or if you've been handed someone else's Google Sheet and you need to understand it.

Press Ctrl and the back-tick mark (underneath the ESC button) to toggle between showing formula results and formula statements.

Look at how it helped uncover the formula mess in this Sheet:



	A	B	C	D
1	ID	Sunday (lookup email)	Monday Date	Tip Number
31	30	=C31-1	19-Nov-2018	26
32	31	=C32-1	26-Nov-2018	27
33	32	=C33-1	3-Dec-2018	28
34	=A33+1	=C34-1	=C33+7	=D33+1
35	=A34+1	=C35-1	=C34+7	30
36	=A35+1	=C36-1	=C35+7	=D35+1
37	=A36+1	=C37-1	=C36+7	=D36+1
38	=A37+1	=C38-1	=C37+7	=D37+1
39	=A38+1	=C39-1	=C38+7	=D38+1
40	=A39+1	=C40-1	=C39+7	=D39+1
41	=A40+1	=C41-1	=C40+7	=D40+1
42	=A41+1	=C42-1	=C41+7	=D41+1
43	=A42+1	=C43-1	=C42+7	=D42+1
44	=A43+1	=C44-1	=C43+7	=D43+1
45	=A44+1	=C45-1	=C44+7	=D44+1
46	=A45+1	=C46-1	=C45+7	=D45+1

Formula Challenge #2

Start with this small data table in your Google Sheet:

	A	B	C
1	Result	Terms	
2	One	Apple, Banana	
3	Two	Peach, Apple	
4	Three	Strawberry	
5	Four	Lemon, Lime	
6	Five	Apple, Lime	
7	Six	Banana, Orange	
8	Seven	Orange, Pear	
9	Eight	Blueberries	
10	Nine	Raspberries, Orange, Apple	
11	Ten	Peach, Strawberries	
12			

Your challenge is to create a single-cell formula that takes a string of search Terms and returns all the Results that have at least one matching term in the Terms column.

For example, this search (in cell E2 say)

Raspberries, Orange, Apple

would return the results (in cell F2 say):

One

Two

Five

Six

Seven

Nine

like this (where the yellow is your formula):

E	F
Search	Results
Raspberries, Orange, Apple	One
	Two
	Five
	Six
	Seven
	Nine

If these instructions are not crystal clear, see if this ready-made [Formula Challenge template helps](#).

I'll share some solutions in the next tip!

#53. Use the SPLIT Function to Separate Text Into an Array Output of Constituent Parts

In tip #52, I challenged you to start with this small data table, in range A2:A11, in your Google Sheet:

	A	B	C
1	Result	Terms	
2	One	Apple, Banana	
3	Two	Peach, Apple	
4	Three	Strawberry	
5	Four	Lemon, Lime	
6	Five	Apple, Lime	
7	Six	Banana, Orange	
8	Seven	Orange, Pear	
9	Eight	Blueberries	
10	Nine	Raspberries, Orange, Apple	
11	Ten	Peach, Strawberries	
12			

And create a single-cell formula in cell F2 that takes a string of search Terms as an input, from cell E2, and returns all the Results that have at least one matching term in the Terms column, like this (where the yellow is your formula):

E	F
Search	Results
Raspberries, Orange, Apple	One
	Two
	Five
	Six
	Seven
	Nine

Formula Challenge #2 Solution

Solution 1:

```
=FILTER(A2:A11, REGEXMATCH(B2:B11, JOIN(" | ", SPLIT(E2, ", "))))
```

Congratulations to Maryam B. for this extremely elegant solution, which was also the shortest solution submitted.

There were a lot of similar entries that had an **ArrayFormula** function inside the Filter, but this is not required since the Filter function will output an array automatically.

How does this formula work?

Let's begin in the middle and rebuild the formula in steps:

```
=SPLIT(E2, ", ")
```

splits out the three fruits in cell E2 into separate cells:

Raspberries Orange Apple

Next, join them back together with the pipe "|" delimiter with

```
=JOIN(" | ", SPLIT(E2, ", "))
```

so the output is now:

Raspberries | Orange | Apple

Then bring the power of regular expression to the table, to match the data in column B. The pipe character means "OR" in regular expressions, so this formula will match Raspberries OR Orange OR Apple in column B:

```
=REGEXMATCH(B2:B11, JOIN(" | ", SPLIT(E2, ", ", )) )
```

On its own, this formula will return a #VALUE! error message. (Wrap this with the ArrayFormula function if you want to see what the array of TRUE and FALSE values looks like.)

However, when we put this inside of a FILTER function, the correct array value is passed in:

```
=FILTER(A2:A11, REGEXMATCH(B2:B11, JOIN(" | ", SPLIT(E2, ", ", )) ) )
```

and returns the desired output. Kaboom! 💥

Solution 2:

```
=QUERY(A1:B11, "select A where B contains '"&JOIN("' or B contains  
'", SPLIT(E2, ", ", )) &"'" )
```

Thanks to Eric S. for this excellent solution. Again, there is no requirement to use an ArrayFormula anywhere. Impressive!

This formula takes a different approach to solution 1 and uses the QUERY function to filter the rows of data.

The heart of the formula is similar though, splitting out the input terms into an array, then recombining them to use as filter conditions.

```
=join("' or B contains '", split(E2, ", ", 0))
```

which outputs a clause ready to insert into your query function, viz:

'Raspberries' or B contains 'Orange' or B contains 'Apple'

The QUERY function uses a pseudo-SQL language to parse your data. It returns rows from column A, whenever column B contains Raspberries OR Orange OR Apple.

Wonderful!

[Click here to open a read-only version of the template with solutions added.](#) (**File > Copy** to make your own editable copy.)

I hope you enjoyed this challenge and learnt something from it. I really enjoyed reading all the submissions and definitely learnt some new tricks myself.

#54. Colorful Sheets!

Today's tip is an easy but useful technique to improve the clarity of your Google Sheets: use colors to add context to data in your Google Sheets.

Here are four ideas for how you might add colors to your Google Sheets to make them more readable or understandable:

1. Alternating Colors

Make your tables easier to read by adding alternating colors. You can optionally define Header and Footer styles too. Add via the **Format > Alternating colors...** menu

	A	B	C	D	E	F	G	H
1	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %	Gross Commission Income	
2	1	Apartment	Referral	Buyer	\$328,484	3.00%	\$9,855	
3	2	House	Previous client	Buyer	\$1,145,078	3.00%	\$34,352	
4	3	Apartment	Previous client	Buyer	\$587,737	3.00%	\$17,632	
5	4	Apartment	PPC	Seller	\$522,867	3.00%	\$15,686	
6	5	Apartment	Website	Seller	\$321,706	3.00%	\$9,651	
7	6	Condo	Website	Seller	\$486,653	3.00%	\$14,600	
8	7	Apartment	Referral	Seller	\$482,425	3.00%	\$14,473	
9	8	Townhouse	Referral	Buyer	\$931,004	3.00%	\$27,930	
10	9	Condo	Referral	Seller	\$321,095	3.00%	\$9,633	
11	10	Townhouse	Previous client	Buyer	\$548,455	3.00%	\$16,454	
12	11	Apartment	Referral	Buyer	\$376,895	3.00%	\$11,307	
13	12	Condo	Previous client	Buyer	\$709,642	3.00%	\$21,289	

2. Heatmaps

Make your data pop and bring attention to the highest or lowest values. Add via the **Format > Conditional formatting... > Color scale** menu

Cohort 12-2016	Cohort 01-2017	Cohort 02-2017	Cohort 03-2017	Cohort 04-2017	Cohort 05-2017	Cohort 06-2017	Cohort 07-2017	Cohort 08-2017	Cohort 09-2017
Dec-2016	Jan-2017	Feb-2017	Mar-2017	Apr-2017	May-2017	Jun-2017	Jul-2017	Aug-2017	Sep-2017
100%	89%	100%	84%	100%	93%	90%	99%	97%	95%
60%	56%	79%	65%	83%	82%	77%	84%	86%	88%
40%	39%	63%	51%	71%	73%	65%	74%	73%	72%
0%	33%	46%	35%	56%	61%	56%	62%	63%	64%
0%	28%	42%	30%	46%	54%	44%	55%	50%	52%
0%	17%	33%	24%	39%	48%	35%	43%	41%	40%
0%	17%	33%	24%	29%	45%	29%	36%	34%	33%
0%	17%	29%	22%	29%	39%	27%	36%	31%	30%
0%	17%	25%	22%	27%	38%	27%	36%	30%	29%
0%	17%	25%	22%	27%	38%	27%	32%	28%	27%
0%	17%	25%	19%	27%	36%	27%	32%	32%	31%
0%	17%	25%	19%	27%	36%	27%	32%	32%	31%
0%	17%	25%	19%	27%	36%	27%	32%	32%	31%

3. Distinguish new columns in your datasets

I find it helpful, from an audit perspective, to know which columns are original columns from the data source and which are calculation columns I've added. It becomes more important when you're cleaning big datasets. Simply highlight the entire new column and give it a background color.

1	Listing	Property Type	Lead Source	Representation	Sales price	Gross Commission Income %	Gross Commission Income	Gross Commission Income @ 5%	H
2	1	Apartment	Referral	Buyer	\$320,404	3.00%	\$9,612	\$16,424	
3	2						\$34,352	\$57,254	
4	3						\$17,632	\$29,387	
5	4						\$15,686	\$26,143	
6	5	Apartment	Website	Seller	\$321,706	3.00%	\$9,651	\$16,085	
7	6	Condo	Website	Seller	\$486,653	3.00%	\$14,600	\$24,333	
8	7	Apartment	Referral	Seller	\$482,425	3.00%	\$14,472	\$24,121	
9	8	Townhouse						\$46,550	
10	9	Condo					\$9,633	\$16,055	
11	10	Townhouse					\$16,454	\$27,423	
12	11	Apartment	Referral	Buyer	\$376,895	3.00%	\$11,307	\$18,845	
13	12	Condo	Previous client	Buyer	\$709,642	3.00%	\$21,289	\$35,482	
14	13	Apartment	PPC	Buyer	\$277,876	3.00%	\$8,336	\$13,894	
15	14	Townhouse	Referral	Seller	\$838,370	3.00%	\$25,151	\$41,919	
16	15	Apartment	Referral	Buyer	\$417,507	3.00%	\$12,525	\$20,875	

4. Highlight formula cells

To make it easier to find my formulas, I often highlight the formula cells yellow :)

fx =FILTER(A2:A11,REGEXMATCH(B2:B11,JOIN(" | ",SPLIT(E2,", "))))

	A	B	C	D	E	F
1	Result	Terms			Search	Results
2	One	Apple, Banana			Raspberries, Orange, Apple	One
3	Two	Peach, Apple				Two
4	Three	Strawberry				Five
5	Four	Lemon, Lime				Six
6	Five	Apple, Lime				Seven
7	Six	Banana, Orange				Nine
8	Seven	Orange, Pear				
9	Eight	Blueberries				
10	Nine	Raspberries, Orange, Apple				

Formula cell highlighted



#55. Use the MID Function to Return a Segment of a String

Google has hidden a Pride-color theme in Sheets (these hidden features are called Easter Eggs, because, well, you have to hunt for them...).

Let's find it with the MID function.

Start with this formula in cell A1 of a blank Sheet:

```
=COLUMNS ($A$1:A1)
```

Notice how the first A1 is an absolute reference surrounded by \$ signs (so that it's locked in place) but the second A1 is a relative reference without any \$ signs (so it will move).

Drag this across row 1 and you'll get a set of ascending numbers: 1, 2, 3 etc.

Let's bring the MID function into play and split the word "HELLO" into its separate letters. Try this in cell A1:

```
=MID ("HELLO", COLUMNS ($A$1:A1), 1)
```

Drag across 5 columns (staying on row 1) and you'll get an "H" in cell A1, an "E" in cell B1, an "L" in C1 etc. to split out HELLO.

Cool, huh!

Let's up the ante a bit and create a single array formula to achieve this.

In a new tab, type this formula into cell A1:

```
=LEN("HELLO")
```

Gives the answer 5, not rocket science. Next, add this:

```
="1:"&LEN("HELLO")
```

which outputs 1:5 in cell A1.

Wrap this with the INDIRECT function to convert this string into a valid range reference. Then wrap it with the ROW function and hit Ctrl+Shift+Enter to convert to an Array Formula, like this (still in A1):

```
=ArrayFormula(ROW(INDIRECT("1:"&LEN("HELLO"))))
```

This gives a column of numbers, 1 to 5, in column A.

Great! Now we're ready to use these numbers with the MID function again to split our word again. Add the MID function to our formula in cell A1:

```
=ArrayFormula(MID("HELLO",ROW(INDIRECT("1:"&LEN("HELLO"))),1))
```

This single formula splits HELLO into separate letters in column A. Finally, let's transpose that from a column to a row:

```
=ArrayFormula(TRANSPOSE(MID("HELLO",ROW(INDIRECT("1:"&LEN("HELLO"))),1)))
```

Looking good. That formula outputs HELLO as separate letters across row 1.

So what the heck has any of this got to do with colors or Pride I hear you ask?

Swap the word "HELLO" in your formula in cell A1 to "PRIDE" and watch your Google Sheet light up!

```
=ArrayFormula(TRANSPOSE(MID("PRIDE",ROW(INDIRECT("1:"&LEN("PRIDE"))),1)))
```

And here's the output!

fx | =ArrayFormula(TRANSPOSE(MID("PRIDE",ROW(INDIRECT("1:"&LEN("PRIDE"))),1)))

	A	B	C	D	E	F
1	P	R	I	D	E	
2						
3						
4						
5						
6						
7						

Note: you can also just type the letters “P”, “R”, “I”, “D” and “E” into the 5 adjacent columns to see this effect.

#56. Quick Entry & Exit With Formulas

Use F2 or Shift + Return to enter into your formulas. Use Escape to exit your formulas.

Have you ever found yourself needing to copy part of a formula to use elsewhere?

Or maybe you've found yourself trying to click out of your formula, but Sheets thinks you want to highlight a new cell and it messes up your formula...

Here are the shortcut keys you need to quickly enter and exit your formulas!

Start by selecting a cell containing a formula.

Press the **F2** key, or press **Shift + Return**, to enter into the formula.

Press the **Escape** key to exit your formula and return to the result view.

Any changes are discarded if you hit the **Escape** key (to save changes you just hit the usual **Return** key).

	H	I	J	K	L	
1	Lookup Amount					
2	\$383					
3						
4						

Here's another quick trick that's helpful for longer formulas:

When you're inside the formula view, press the **Up arrow** to go to the front of your formula (in front of the equals sign). Similarly, pressing the **Down arrow** takes you to the last character in your formula.

	H	I	J	K	L	
1	Lookup Amount					
2		=ArrayFormula(VLOOKUP(G2 , { A2:A9&" "&B2:B9 , C2:D9 } , 3 , FALSE))				
3						
4						

#57. Discover New Formulas Today

At the time of writing this tip (July 2019) there are 466 (!) documented functions available in Google Sheets. How many have you used?

I encourage you to find and use a new function today!

Here are a few ways you can explore the functions available in Google Sheets:

- 1) Import all 490+ Google Sheets functions into your own Google Sheet with this function:

```
=IMPORTHTML("https://support.google.com/docs/table/25273" , "table" , 1)
```

	A	B	C	D
1	Type	Name	Syntax	Description
2	Date	DATE	DATE(year, month, day)	Converts a provided year, month, and day into a date. Learn more
3	Date	DATEDIF	DATEDIF(start_date, end_date, unit)	Calculates the number of days, months, or years between two dates. Learn more
4	Date	DATEVALUE	DATEVALUE(date_string)	Converts a provided date string in a known format to a date value. Learn more
5	Date	DAY	DAY(date)	Returns the day of the month that a specific date falls on, in numeric format. Learn more

How meta!

- 2) With the function above in cell A1, you have a list of functions in columns A to D. Select a random one from this list with this formula:

```
=INDEX(A:D , RANDBETWEEN(2 , 467))
```

<i>fx</i>	=INDEX(A:D,RANDBETWEEN(2,467))	F	G	H	I	J	
1	Type	Name	Syntax	Description			
2	Statistical	DEVSQ	DEVSQ(value1, [value2, ...])	Calculates the sum of squares of deviations from the mean.			
3							

* the 467 refers to the total number of functions in the list, so you should update it to the newest number (492 at time of writing)

3) Another way: Type a single letter after an equals sign and browse the function list in the auto-complete menu:

<i>fx</i>	=G	A	B	C	D	
1	=G					
2	GT					
3	GCD					
4	GTE					
5	GAMMA					
6	GAUSS					
7	GESTEP					
8	Returns 0 or 1 depending on input.					
9	GROWTH					
10	GAMMALN					
11	GEOMEAN					
12	GAMMAINV					
13						
14						

This is how I discovered the GESTEP function!

#58. Make Conditional Formatting Permanent

This tip was inspired by a question from a reader who wanted to remove the conditional formatting rules but keep the formatting applied.

How can we do that?

You can use **Paste Special > Paste Format Only** to make Conditional Formatting permanent.

You've probably used conditional formatting to color rows when certain conditions are met, or perhaps to highlight data on specific dates.

Sometimes you're finished with the "conditional" part and your data is set. You want to delete the conditional formatting rule to make your sheet simpler (so it's faster and less complex for others to interact with). But how?

If you simply delete the conditional formatting rule then ZAP! away goes all the formatting...

This is what you need to do instead:

1. Highlight the whole range to which you've applied the conditional formatting
2. Copy this range
3. Delete the Conditional Formatting rule
4. Right click or go to the Edit menu and use **Paste Special > Paste Format Only**

Bingo! All that formatting comes back and this time it's permanent.

#59. Use Custom Number Formatting to Pluralize Data

Have you ever used Custom Number Formatting? It's hidden deep under the format menu so it's not well known. However, it's an incredibly powerful tool to have in your toolbox.

Let's imagine a scenario where we have a column for counting the number of days:

	A	
1	Day Count	
2		1
3		2
4		3
5		

We can apply custom number formatting to add "day" or "days" to the numbers, as follows:

<i>fx</i>	1
	A
1	Day Count
2	1 day
3	2 days
4	3 days
5	

Take a look at the formula bar and notice that the underlying data is still just a number.

In other words we haven't changed the datatype from number to text. We can still add numbers to this column for example.

To apply Custom Number Formatting, go to the Format menu:

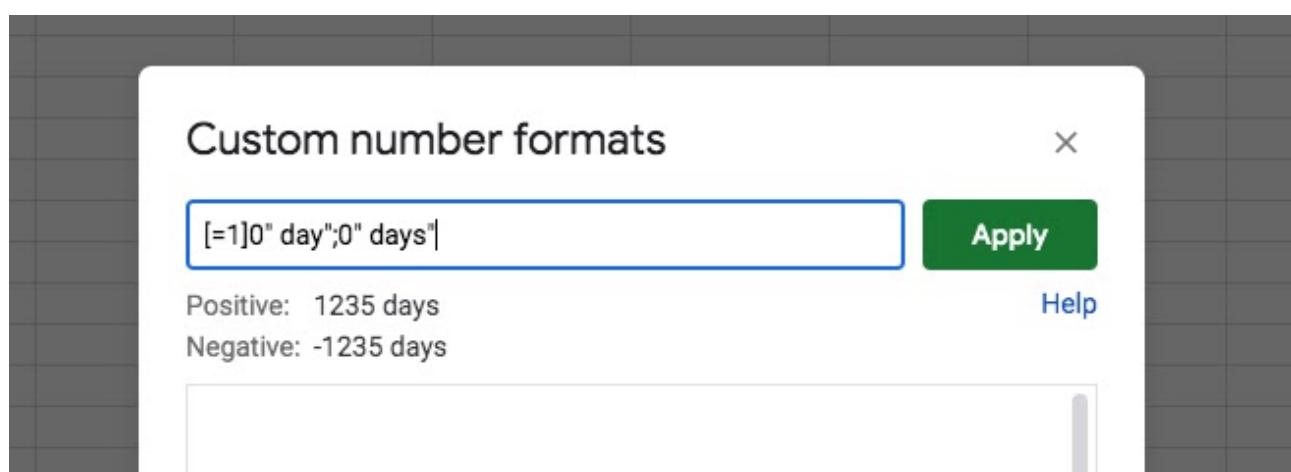
Format > Number > More Formats > Custom number formats...

And set the rule to

[=1]0" day";0" days"

This neat little formula tests for whether the number in the cell is equal to 1 and, if it is, adds "day" to the number to give "1 day".

Everything else in the cell will have " days" added to give e.g. "3 days"



This is just scratching the surface with what's possible with Custom Number Formatting.

You can add colors. Format numbers to "k", "m" for thousands and millions etc. Set negative numbers with brackets. Etc. There are hundreds, maybe even thousands, of different scenarios you could apply it to.

I encourage you to experiment with Custom Number Formatting!

#60. Be Creative With Your Google Sheets!

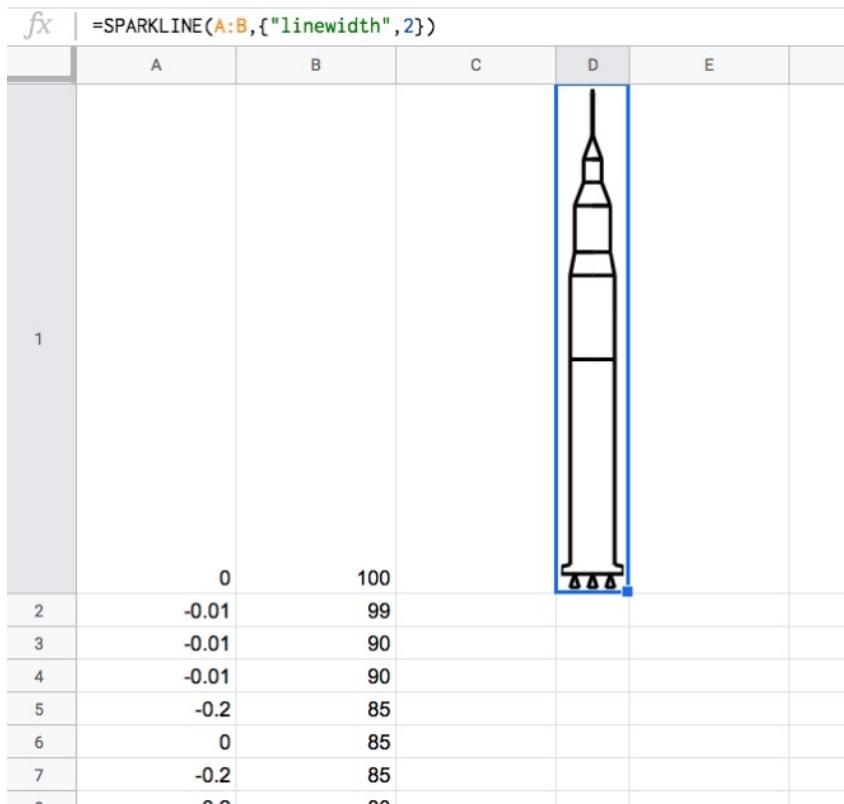
Over 50 years ago, on July 20th, 1969, the lunar module from the Apollo 11 mission touched down on the surface of the moon. Astronauts Neil Armstrong and Buzz Aldrin became the first human beings to step foot on another world.

Last year, I thought it was fitting to celebrate the 50th Anniversary by showcasing a few space themed formulas in Google Sheets!

First off, we begin our journey on planet Earth, which we create with the CHAR function:

	fx	=char(127758)	
	A	B	
1			
2			

To get to the moon, we need a gigantic Saturn V rocket, which we can draw by supplying a series of coordinates to the SPARKLINE function:



If you give a sparkline a range of x- and y-coordinates it will create 2-d shapes.

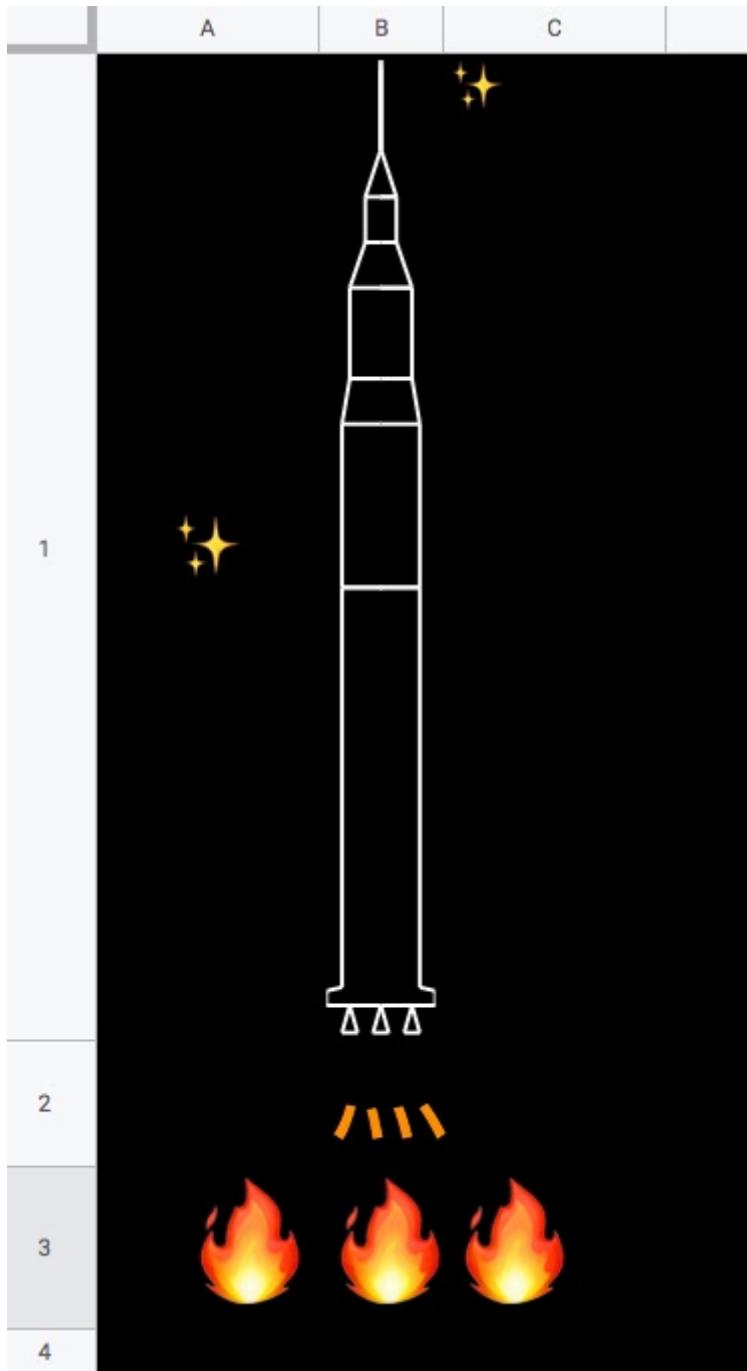
You'll notice columns A and B, the data range for the sparkline function, are filled with numbers, which are coordinates for the sparkline formula to trace our rocket outline.

This was created by an iterative process, starting with a square with a triangle on top, and working from there.

[Feel free to make your own copy of the Saturn V template here \(File > Make a copy...\)](#)

Saturn V stood 363 feet tall, about the same height as a 36-story tall building, and weighed 6.5 million pounds (2,950,000 kg) at liftoff. Wow!

Apollo 17 was the only night launch:



Here the following CHAR formulas are used: CHAR(10024) creates the stars, CHAR(11939) creates the exhaust outflow and CHAR(128293) creates the fire!

This engineering leviathan carried three astronauts to the moon. We can draw the moon in our Google Sheet with the CHAR function:

	$f\!x$	=char(127767)	
	A	B	
1			
2			

What creative things have you done with Google Sheets?

#61. Date Validation

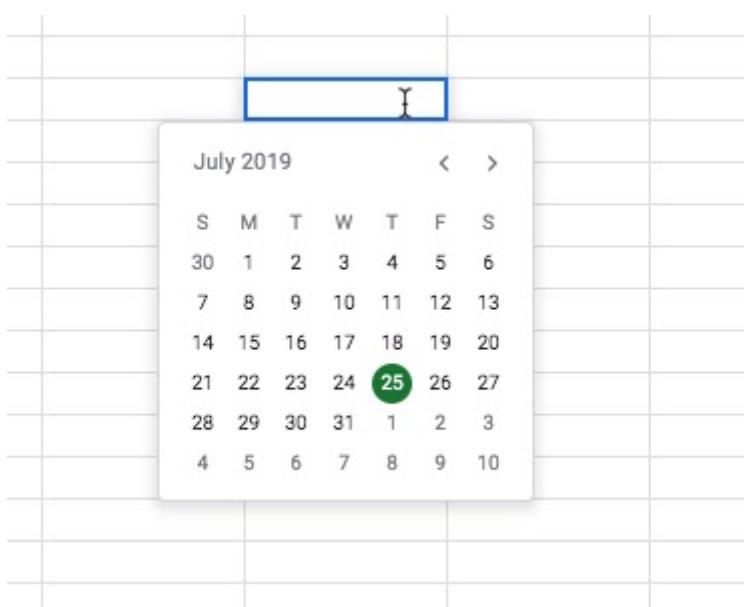
If you enter dates into a cell in your Sheets then you'll love this one.

Here's how to use Data Validation to ensure valid dates are added in a cell:

- 1) From the menu: **Data > Data validation...** or right click in a cell and choose **Data validation...**
- 2) Select **Date** in the Criteria option of the data validation window

This will ensure that only dates can be added in this cell.

Additionally, you can double click on the cell to bring up the date picker:



It's that easy!

#62. Add Star Rating System

Here's how to add a star rating system to your Sheets with the REPT and CHAR functions.

It's quick and easy to add a star rating system to your Google Sheets.

Here's one that shows some of the top movies in 2019, with some fictional ratings (I haven't seen any of these movies so can't comment on the veracity of these ratings!):

	A	B	C
1	Title	Score	Rating
2	Avengers: Endgame	3	★★★
3	Captain Marvel	5	★★★★★
4	Spider-Man: Far From Home	1	★
5	Aladdin	5	★★★★★
6	The Lion King	4	★★★★
7	Toy Story 4	4	★★★★
8	The Wandering Earth	5	★★★★★
9	How to Train Your Dragon: The Hidden World	4	★★★★
10	Pokémon Detective Pikachu	5	★★★★★
11	Alita: Battle Angel	2	★★
12			

In column B is a numerical value representing the rating.

In column C, that value is transformed into a more visually appealing star rating system.

The star is created with our friend the CHAR function. Then we use the REPT function to show the correct number of stars.

The formula for row 2 is:

```
=REPT( CHAR( 9733 ) , B2 )
```

You can drag this down the column to fill in all the other star ratings.

It's that easy!

P.S. If you want to practice your Array Formulas, see if you can convert the formula above into an array version to fill out the whole column with a single formula.

#63. Create QR Codes With Formulas

Every time I discover new functionality in Google Sheets I'm blown away by how powerful and versatile it is. This tip is no exception.

Here's how you can create QR codes in your Google Sheets with a formula.

We call an API with the IMAGE function and it returns a QR code based on our data. It's pretty neat!

	A	B	C	D	E	
1	https://www.benlcollins.com/					
2	https://twitter.com/home					

We can then scan the QR code with your phone camera or a QR reader to quickly access that data, e.g. here's a QR code I created to access benlcollins.com website (phone screenshot):



The formula to generate the QR code is:

```
=IMAGE("https://api.qrserver.com/v1/create-qr-code/?data="&A1,4,70,70)
```

where your data (e.g. a web url or name or address etc.) is in cell A1.

It uses this [QR code generator API](#).

What would you use this function for?

#64. Apps Script Button

A friend recently asked for pointers on creating a script to clear out values in an invoice template with a single button click. This tip will show you how to do it.

In this example I create a basic invoice template with placeholders to hold information:

	A	B	C	D	E
1	Invoice				
2	8/16/2019				
3					
4					
5	Number		To		
6			From		
7					
8	Amount				
9					
10					
11					

The user can enter information into cells B5, B8, E5 and E6 (shown in yellow).

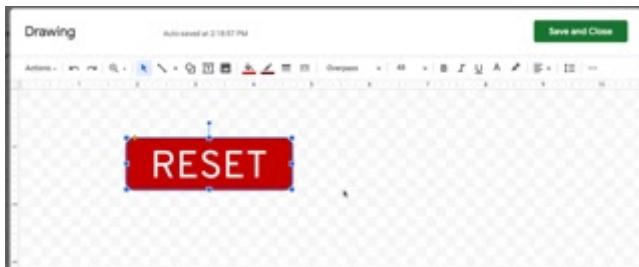
In the script editor, accessed through **Tools > Script Editor**, I add a very simple script to clear these specific cells out:

```
function clearInvoice() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var invoiceNumber = sheet.getRange("B5").clearContent();
  var invoiceAmount = sheet.getRange("B8").clearContent();
  var invoiceTo = sheet.getRange("E5").clearContent();
  var invoiceFrom = sheet.getRange("E6").clearContent();
}
```

I can run this function from the script editor and it will clear out the contents of the invoice. But I want to do that from the front-end of my Google Sheet.

To do that, I can add either a menu or a button.

In this example, I add a button via the **Insert > Drawing** menu. This brings up the drawing editor where I can easily add a box and style it to look like a button:



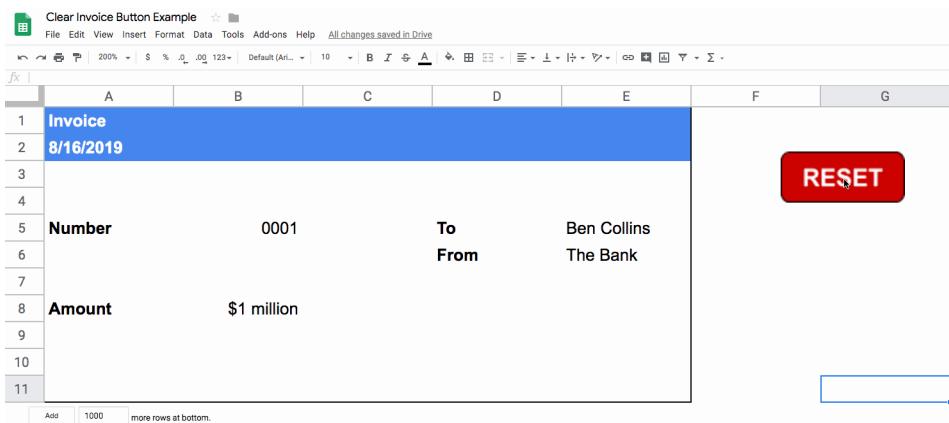
When I click Save and Close, this drawing gets added to my Google Sheet. I can click on it to resize it or drag it around to reposition it.

To assign a script, I click the three little dots in the top right of the drawing and select Assign Script:



Then I type in the name of the function I want to run from my Apps Script code, in this case the **clearInvoice** function.

Now, when I click the button it will clear out the invoice for me!



One last point to note: to edit or move the button after you've assigned it to a script, you now need to right-click on it.

#65. QA Checklists

Here's how you can use Quality Control checklists to reduce errors in your work.

Whenever I work with complex Sheets, especially ones for clients, I spend time checking my work before sharing it.

The last thing I want to do is to share a Sheet full of errors. It's disastrous for both parties.

Complex projects benefit from robust Quality Control checks

But are you consistent in how you do it? Or is it an afterthought?

Next time you check a Sheet, write down all the spot checks you do.

Now, whenever you update this Sheet, you can run through the same list of pre-defined quality control checks.

It'll reduce the likelihood of errors and save you time and energy, because you don't have to figure out what checks to do every time.

It doesn't matter how small or insignificant the check is, write it down on your list. It could be the one that catches an error.

Here's an example QA checklist sheet I use for one client project:

	A	B	C	D	E	F
1	QA checklist					
2						
3	Annual Dashboard Sheet					
4	Month set correctly				<input checked="" type="checkbox"/>	
5	Set to 100% and check total figures match to monthly dashboard				<input checked="" type="checkbox"/>	
6	Spot check totals in forecast table				<input type="checkbox"/>	
7						
8	Monthly Dashboard Sheet					
9	Spot check totals for actual data				<input type="checkbox"/>	
10						
11	Forecast Demand Sheet					
12	Spot check past 12 month figures from Historic Data tab				<input type="checkbox"/>	
13	Spot check past 12 month Patient Actual figures from Patient Historic Data tab				<input checked="" type="checkbox"/>	
14	Spot check past 12 month Staff Actual Hours figures from Staff Historic Data tab				<input checked="" type="checkbox"/>	
15	Spot check future numbers of actual/projected				<input type="checkbox"/>	
16						
17	Historic Period Dashboard					
18	Check all tables and charts reflect the most recent month				<input type="checkbox"/>	

[\(click here to open Sheet\)](#)

In this example, I group the checks according to which tab of my Sheet they relate to.

I add checkboxes and a conditional formatting rule so I can tick things off as I complete them. I can see at a glance what I have or haven't done yet.

Here is the conditional highlighting rule I use in this case (the checkboxes are in column F):

Format rules

Format cells if...

Custom formula is ▾

=F1=TRUE

Formatting style

Custom

B I U S A ▾ | ▾

Cancel

Done

A Quality Control checklist will reduce errors and make you more efficient.

#66. Use the Parser Functions to Format Array Data

I had a question from a reader recently: How can I format an array in Google Sheets so that the output is presented as a percentage, instead of as a number?

Let's explore how you can do this with Parser functions.

The Parser functions are a family of functions that format data in a very specific way:

	A	B
1	Function	Result
2	=TO_PERCENT(0.5)	50%
3	=TO_DATE(43710)	9/2/2019
4	=TO_TEXT(99)	99
5	=TO_DOLLARS(24.99)	\$24.99
6	=TO_PURE_NUMBER(17%)	0.17
7		

To answer the original question, you use the TO_PERCENT function inside an ArrayFormula to present the output as a percentage.

It's a great use case for these functions.

Let's see an example.

Here's a contrived array formula that will output 0.1, 0.11, 0.12, 0.13 etc. up to 0.2:

```
=ArrayFormula(ROW(Sheet1!10:20)/100)
```

To format it as percentages, add the TO_PERCENT function inside the ArrayFormula:

```
=ArrayFormula(TO_PERCENT(ROW(Sheet1!10:20)/100))
```

The output now is 10%, 11%, 12%, 13% etc.

Why not just use the formatting buttons I hear you say?

Well, if your arrays are dynamic, i.e. they change size, then using a Parser function ensures the formatting is applied to any new data too.

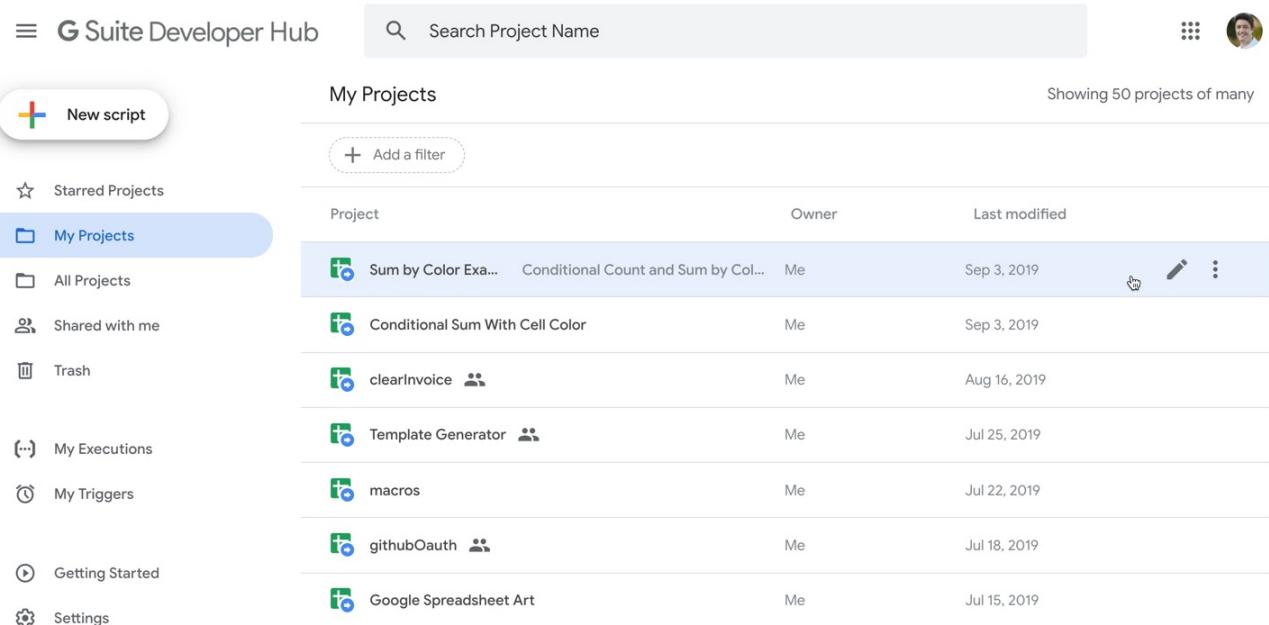
#67. Manage Apps Script Projects From the Dashboard

This tip will show you how to use the Apps Script dashboard to centrally manage your Apps Script projects and triggers.

As you create more and more Apps Script files, keeping track of them (and their triggers!) can become a challenge. Imagine having to open all your Google Sheets to peer into each Editor window to find a specific, [container-bound](#) script.

Of course you don't have to do that.

Instead, use the Apps Script dashboard to see and manage all your script files and triggers in one central place.



The screenshot shows the G Suite Developer Hub interface. On the left, there's a sidebar with navigation links: 'Starred Projects' (disabled), 'My Projects' (selected and highlighted in blue), 'All Projects' (disabled), 'Shared with me' (disabled), 'Trash' (disabled), 'My Executions' (disabled), 'My Triggers' (disabled), 'Getting Started' (disabled), and 'Settings'. The main area is titled 'My Projects' and shows a list of 50 projects. A search bar at the top says 'Search Project Name'. The project list includes columns for 'Project', 'Owner', and 'Last modified'. Each project entry has a small icon, the project name, its description (if any), the owner (Me), the last modification date, and three icons for more actions: a hand cursor, a pencil, and three dots. The projects listed are: 'Sum by Color Ex...', 'Conditional Count and Sum by Col...', 'Me', 'Sep 3, 2019'; 'Conditional Sum With Cell Color', 'Me', 'Sep 3, 2019'; 'clearInvoice', 'Me', 'Aug 16, 2019'; 'Template Generator', 'Me', 'Jul 25, 2019'; 'macros', 'Me', 'Jul 22, 2019'; 'githubOauth', 'Me', 'Jul 18, 2019'; and 'Google Spreadsheet Art', 'Me', 'Jul 15, 2019'.

The dashboard lets you:

- See and search through all your script files at once
- Create new script files
- View details about script files
- Monitor the usage and see any failures of scripts
- Add, remove and edit project triggers
- And more...

Access your Apps Script dashboard at: <https://script.google.com/home>

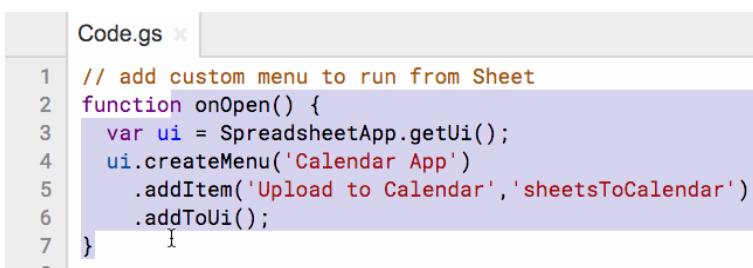
And read more about the dashboard in the [Google documentation here](#).

#68. Brilliant Apps Script Keyboard Shortcuts

Use these keyboard shortcuts to work more efficiently in the Apps Script editor. These simple shortcuts are SO useful when you're working with Apps Script that I implore you to take a few minutes today to try them out!

Auto Comment with Ctrl + /

This works on individual lines or blocks of your Apps Script code.



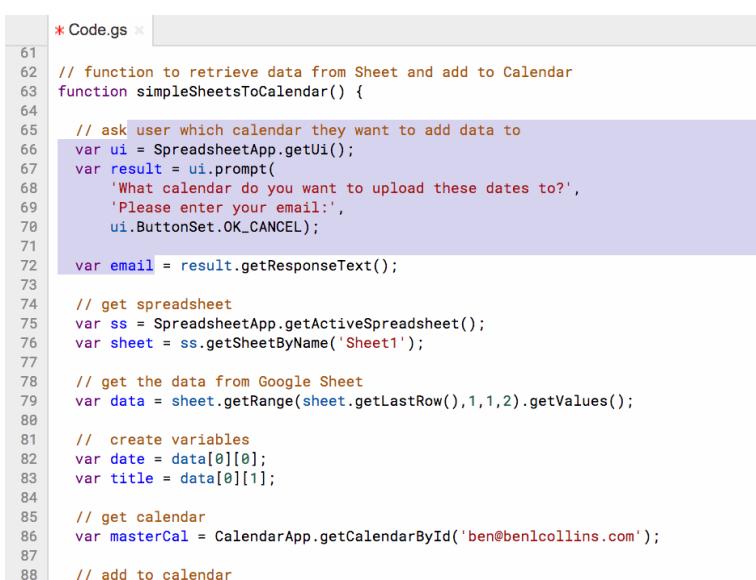
```

Code.gs ✘
1 // add custom menu to run from Sheet
2 function onOpen() {
3   var ui = SpreadsheetApp.getUi();
4   ui.createMenu('Calendar App')
5     .addItem('Upload to Calendar','sheetsToCalendar')
6     .addToUi();
7 }

```

Move code up and down with Alt + Up/Down

If you find yourself wanting to move code around, this is SUPER handy.



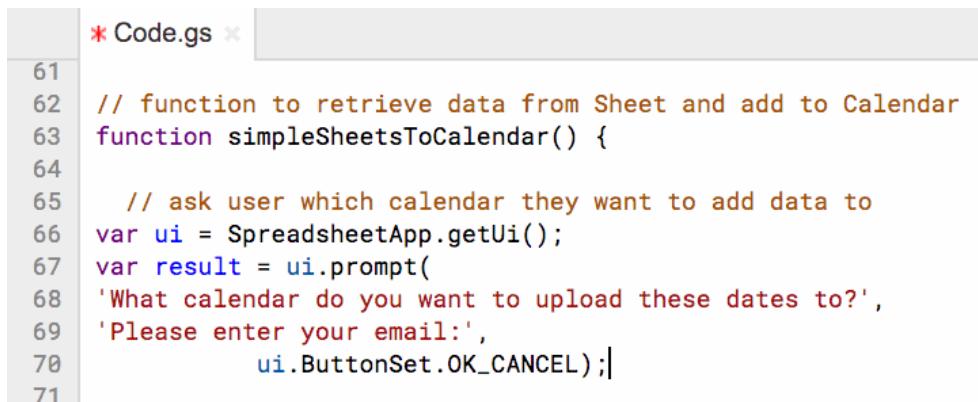
```

* Code.gs ✘
61
62 // function to retrieve data from Sheet and add to Calendar
63 function simpleSheetsToCalendar() {
64
65   // ask user which calendar they want to add data to
66   var ui = SpreadsheetApp.getUi();
67   var result = ui.prompt(
68     'What calendar do you want to upload these dates to?',
69     'Please enter your email:',
70     ui.ButtonSet.OK_CANCEL);
71
72   var email = result.getResponseText();
73
74   // get spreadsheet
75   var ss = SpreadsheetApp.getActiveSpreadsheet();
76   var sheet = ss.getSheetByName('Sheet1');
77
78   // get the data from Google Sheet
79   var data = sheet.getRange(sheet.getLastRow(),1,1,2).getValues();
80
81   // create variables
82   var date = data[0][0];
83   var title = data[0][1];
84
85   // get calendar
86   var masterCal = CalendarApp.getCalendarById('ben@benlcollins.com');
87
88   // add to calendar

```

Tidy up indentation with Tab

Keeping your code properly indented makes it much easier to read and understand. This handy shortcut will help you do that. It's especially useful if you've copied code from somewhere else and the indenting is all higgledy-piggledy.



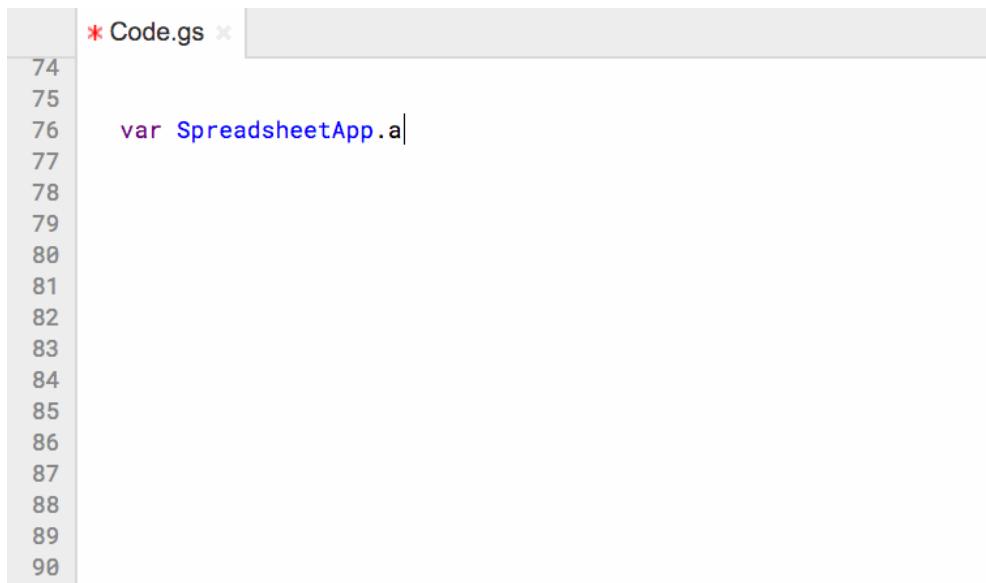
```

61 // function to retrieve data from Sheet and add to Calendar
62 function simpleSheetsToCalendar() {
63
64   // ask user which calendar they want to add data to
65   var ui = SpreadsheetApp.getUi();
66   var result = ui.prompt(
67     'What calendar do you want to upload these dates to?',
68     'Please enter your email:',
69     ui.ButtonSet.OK_CANCEL);
70
71

```

Bring up the Apps Script code auto-complete with Ctrl + Space

How many times have you been typing a class or method, made a spelling mistake only to see the helpful auto-complete list disappear... Bring it back with Ctrl + Space:



```

74
75
76 var SpreadsheetApp.a|
77
78
79
80
81
82
83
84
85
86
87
88
89
90

```

#69. Video Sidebar With Apps Script

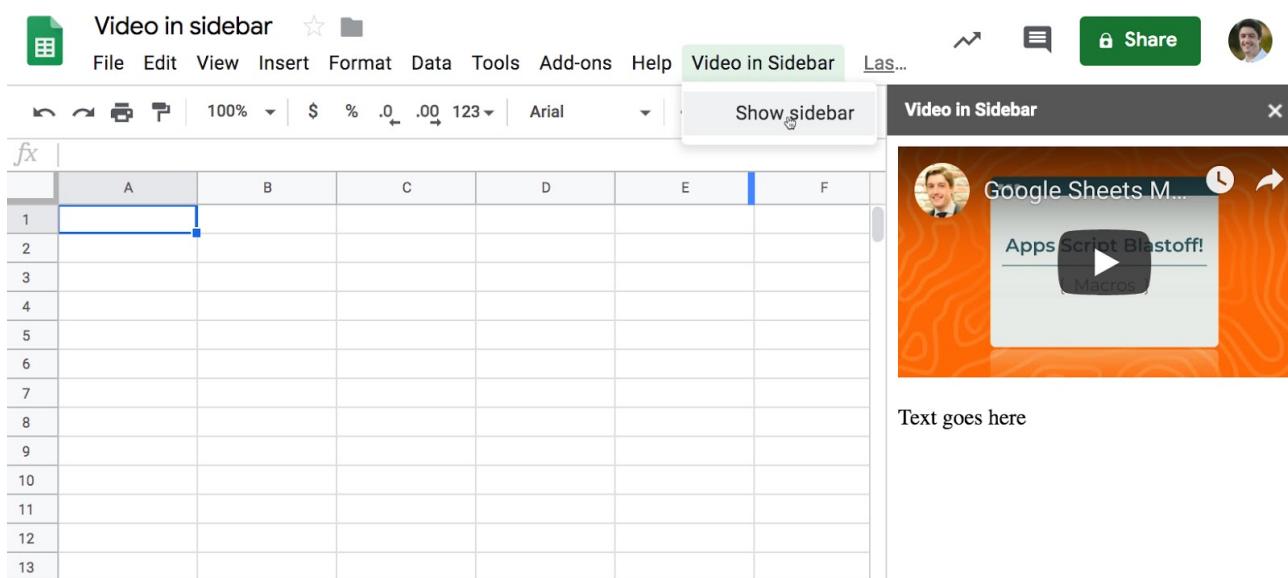
Here's how to embed videos in the sidebar of your Google Sheets using Apps Script.

You can use Apps Script to add sidebars to your Google Sheets. It's a super useful technique for gathering or displaying information for users.

Sidebars can contain HTML and CSS content. You can even embed elements like videos, as in this example.

For example, you could create an internal Sheets add-on for your organization where users can access relevant video tutorials directly inside their Sheets.

Here's an example of a sidebar with an embedded YouTube video:



The sidebar is accessed through a custom menu.

It's a total of 21 lines of code in this example.

In your script editor, add the following code to your **code** file:

```
function onOpen() {
  var ui = SpreadsheetApp.getUi();
  ui.createMenu('Video in Sidebar')
    .addItem('Show sidebar', 'showSidebar')
    .addToUi();
}

function showSidebar() {
  var output =
  HtmlService.createHtmlOutputFromFile('sidebar').setTitle('Video in
Sidebar');
  SpreadsheetApp.getUi().showSidebar(output);
}
```

Create a second **HTML** file, called **sidebar.html**, and insert this code:

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
  </head>
  <body>
    <iframe width="300" height="168.75" frameborder="0"
src="https://www.youtube.com/embed/T2pCuK0oo3I" allowfullscreen></iframe>
    <p>Text can go here</p>
  </body>
</html>
```

#70. Extract Hyperlink URLs With a Formula

Suppose you have a Google Sheet that contains a long list of hyperlinks, which have all been created with formulas like this:

```
=HYPERLINK("https://www.benlcollins.com/", "Ben Collins Website")
```

The second argument is the text to display in the cell as the link.

	A	B	C	D
1	Ben Collins Website			
2				
3				

In this scenario, you want to extract all of the underlying URLs from the hyperlinks.

There are two functions you'll need.

Firstly, turn the hyperlink into text so that the URL is showing, with this formula (assuming the hyperlink formula is in cell A1):

```
=FORMULATEXT(A1)
```

	A	B	C	D	E
1	Ben Collins Website	=HYPERLINK("https://www.benlcollins.com/","Ben Collins Website")			
2					
3					

Next, wrap this with a REGEXEXTRACT function, which extracts just the URL:

```
=REGEXEXTRACT(FORMULATEXT(A1), """(.+?)"""")
```

	A	B	C	
1	Ben Collins Website	https://www.benlcollins.com/		
2				

Now you can drag this formula down the column if you have more hyperlinks to extract.

Quicker than doing it manually!

#71. Annotate Line Charts

We haven't seen many chart tips yet, so let's look at one now. It's a simple technique that helps you add context to your charts. Here's how to use a helper column to add annotations to line charts in Google Sheets.

Let's see it in action with an example.

Suppose you have time series data that you show with a line chart and it would be super helpful to highlight today's value.

You can manually annotate individual points in your chart manually in the chart editor (under the **Customize > Series** menu option) but it's static and won't change over time.

If you want to highlight today's datapoint you'll need a dynamic approach with a helper column. In the helper column, use an IF statement that basically says:

```
IF today THEN value ELSE blank cell
```

Here's an example dataset with a helper column:

<i>fx</i>	<code>=if(A13=today(),B13,"")</code>			
	A	B	C	D
1	Date	Some Value	Today	
9	9/29/2019	88		
10	9/30/2019	51		
11	10/1/2019	80		
12	10/2/2019	78		
13	10/3/2019	72	72	
14	10/4/2019	63		
15	10/5/2019	61		
16	10/6/2019	58		
17	10/7/2019	91		
18	10/8/2019	90		
19	10/9/2019	52		

The IF formula in this example, starting on row 2, is:

```
=IF( A2 = today() , B2 , "" )
```

Create a chart and use columns A, B and C. The date column (A) will be your x-axis, and columns B and C will be your two series. Format series C to show the value and have a large mark size to make it stand out even more (via the menu **Customize > Series** to select the Today series and then **Line Thickness & Data Label**).

Annotated Line Chart showing Today



When you view this chart tomorrow or the next day, the marker point will have automatically shifted along too, so it's always showing today's value.

#72. Build Numbered Lists With the SEQUENCE Function

The SEQUENCE function is a relatively new function in Google Sheets and it's very clever and useful. It's a powerful way to generate numbered lists.

Previously, you had to resort to obscure array formulas like `=ArrayFormula(row(1:5))` to get lists of numbers. Things got ugly fast if you wanted to customize these lists.

Thankfully today, we have the SEQUENCE function.

As arguments, you specify: 1) the number of rows, 2) the number of columns, 3) a start value, and 4) a step size.

Arguments 2, 3 and 4 are optional. However, if you want to set them you need to include the previous ones (e.g. if you want to set a step size in argument 4, then you need to set 1, 2 and 3 as well).

Keep this order in mind as you look through the examples below and you'll soon understand how the function works.

1. Ascending list of numbers

`=SEQUENCE(5)`

	<i>fx</i>	=SEQUENCE(5)
	A	B
1	1	
2		2
3		3
4		4
5		5
6		
7		

2. Horizontal list of numbers

Set the row count to 1 and the column count to however many numbers you want e.g. 5:

=SEQUENCE(1, 5)

	<i>fx</i>	=SEQUENCE(1, 5)				
	A	B	C	D	E	
1	1	2	3	4	5	
2						
3						

3. Two-dimensional array of numbers

Set both row and number values:

=SEQUENCE(5, 5)

	<i>fx</i>	=SEQUENCE(5, 5)				
	A	B	C	D	E	
1	1	2	3	4	5	
2	6	7	8	9	10	
3	11	12	13	14	15	
4	16	17	18	19	20	
5	21	22	23	24	25	
6						
7						

4. Start from a specific value

Set the third argument to the value you want to start from e.g. 100:

=SEQUENCE(5,1,100)

	A	B
1	100	
2	101	
3	102	
4	103	
5	104	
6		

5. Use a custom step

Set the fourth argument to the size of the step you want to use, e.g. 10:

=SEQUENCE(5,1,1,10)

	A	B
1	1	
2	11	
3	21	
4	31	
5	41	
6		

6. Descending numbers

Set the fourth argument to -1 to count down:

=SEQUENCE(5,1,5,-1)

	<i>fx</i>	=SEQUENCE(5,1,5,-1)
	A	B
1	5	
2	4	
3	3	
4	2	
5	1	
6		
-		

7. Negative numbers

Set the start value to a negative number and/or count down with negative step:

=SEQUENCE(5,1,-1,-1)

	<i>fx</i>	=SEQUENCE(5,1,-1,-1)
	A	B
1	-1	
2	-2	
3	-3	
4	-4	
5	-5	
6		

8. Dates

Dates are stored as numbers in spreadsheets, so you can use them inside the SEQUENCE function. You need to format the column as dates:

```
=SEQUENCE(5,1,TODAY(),1)
```

	A	B	
1	10/3/2019		
2	10/4/2019		
3	10/5/2019		
4	10/6/2019		
5	10/7/2019		
6			

(Editor's note: these are US date format MM/DD/YYYY)

9. Decimal numbers

Unfortunately you can't set decimal counts directly inside the SEQUENCE function, so you have to combine with an Array Formula e.g.

```
=ArrayFormula( SEQUENCE(5,1,10,1) / 10 )
```

fx | =ArrayFormula(SEQUENCE(5,1,10,1)/10)

	A	B	C
1	1.00		
2	1.10		
3	1.20		
4	1.30		
5	1.40		
6			

10. Constant numbers

You're free to set the step value to 0 if you want an array of constant numbers:

=SEQUENCE(5,1,1,0)

fx | =SEQUENCE(5,1,1,0)

	A	B	
1	1		
2	1		
3	1		
4	1		
5	1		
6			
7			

Wow! I love the SEQUENCE function 😊

#73. Double Click Dates to Use the Date Picker

Ok, I have a task for you to try right now.

Find a cell in Google Sheets with a date value in (important that it's just a date value and not a date generated by a formula).

Double click it.

Voilà! The date picker comes up for you to select different dates if you wish.

	A	B	C
1	10/3/2019		
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			

It's a handy trick for folks to change dates without having to type them in.

(Editor's note: This feature was also covered in tip #61 about Date Validation.)

#74. Use the RANK Function to Find the Position of a Value in a Dataset

Suppose Eva recently took an exam and scored a highly respectable 83%. She wants to know how she fared in relation to her fellow students who took the same exam.

She can use the RANK function to do this:

```
=RANK( 83 , A2:A101 )
```

It gives a result of 21 in this example.

In other words, Eva's score of 83% placed her 21st out of 100 students who took this exam. Not bad!

	<i>fx</i> =RANK(83,A2:A101)	A	B	C	
1	Exam result				
2	64			21	
3	91				
4	90				
5	89				
6	65				
7	73				
8	57				
9	87				
10	58				
11	71				
12	78				

There is an optional third argument for the RANK function, which determines whether to consider the data as ascending (100 is the best score, 0 the worst score in this example) or descending (0 is the best score, 100 is the worst score in this example).

If this third argument is set to 0, then this is ascending. If it's set to 1, it's descending.

The default option is 0, which is the ascending scenario. Hence we omitted it from our formula above. However, to be really explicit, we could have written the formula like this:

```
=RANK( 83 , A2:A101, 0 )
```

#75. A Fun Use of the FACT Function

We'll start this tip with a mind-blowing fact, and then use the FACT function in Google Sheets to explain it.

Pick up a standard 52 card deck and give it a good shuffle.

The order of cards in a shuffled deck will be unique.

One that has likely never been seen before in the history of the universe and will likely never be seen again.

I'll let that sink in.

Isn't that mind-blowing?

Especially when you picture all the crazy casinos in Las Vegas.

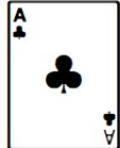
Let's understand why, and in the process learn about the FACT function and basic combinatorics (the study of counting in mathematics).

Here's how to use the FACT function in Google Sheets to show that a shuffled deck of cards is unique, one that's never been seen before.

Four Card Deck

To keep things simple, suppose you only have 4 cards in your deck, the four aces.

You can create this deck in Google Sheets with the CHAR function:

	A	B	C	D	E	
1	Card:	Ace of Clubs	Ace of Spades	Ace of Hearts	Ace of Diamonds	
2	CHAR code:	127185	127137	127153	127169	
3	Ouptut:					
4						
5						

The formulas to create these four cards are:

Ace of Clubs	= char(127185)
Ace of Spades	= char(127137)
Ace of Hearts	= char(127153)
Ace of Diamonds	= char(127169)

Let's see how many different combinations exist with just these four cards.

Pick one of them to start. You have a choice of four cards at this stage.

Once you've chosen the first one, you have three cards left, so there are 3 possible options for the second card choice.

When you've picked that second card, you have two cards left. So you have a choice of two for the third card.

The final card is the last remaining one.

So you have **4 choices * 3 choices * 2 choices * 1 choice = 4 * 3 * 2 * 1 = 24**

There are 24 permutations (variations) with just 4 cards!

Visually, we can show this in our Google Sheet by displaying all the different combinations with the card images from above:

	A	B	C	D	
1					1
2					2
3					
4					
5					
6					

The diagram shows a 6x4 grid of playing cards. Row 1 contains the Ace of Clubs, Ace of Spades, Ace of Hearts, and Ace of Diamonds. Row 2 shows a swap between the Ace of Hearts and the Ace of Diamonds. Row 3 shows another swap. Row 4 shows a third swap. Row 5 shows a fourth swap. Row 6 shows a fifth swap. A large purple double-headed arrow points from the Ace of Hearts in Row 1 to the Ace of Diamonds in Row 2, and from the Ace of Diamonds in Row 1 to the Ace of Hearts in Row 2. Purple circles labeled '1' and '2' are placed next to the arrows.

(I've just shown the first 6 rows for brevity.)

You can see for example, when moving from row 1 to row 2, we swapped the position of the two red suits: the Ace of Hearts and the Ace of Diamonds.

Five Card Deck

This time there are 5 choices for the first card, then 4, then 3, then 2, then 1.

So the number of permutations is $5 * 4 * 3 * 2 * 1 = 120$

Already a lot more! I have not drawn this out in a Google Sheet and leave that as an optional exercise for you if you wish.

The FACT function

The FACT function in Google Sheets is a math function that returns the factorial of a given number. The factorial is the product of that number with all the numbers beneath it.

In other words, exactly what we've done above.

Four:

The 4 card deck formula is `= FACT(4)` which gives an answer of 24 permutations.

Five:

The 5 card deck formula is `= FACT(5)` which gives an answer of 120 permutations.

Six:

A 6 card deck is `= FACT(6)` which gives an answer of 720 permutations.

Twelve:

A 12 card deck, `= FACT(12)`, has 479,001,600 different ways of being shuffled.

(You're more likely to win the Powerball lottery at 1 in 292 million odds, than to get two matching shuffled decks of cards, even with just 12!)

Fifty Two:

Keep going up to a full deck of 52 cards with the formula `= FACT(52)`, and it's a staggeringly large number.

Type it into Google Sheets and you'll see an answer of 8.07E+67, which is 8 followed by 67 zeros!

(This number notation is called scientific notation, where huge numbers are rounded to the first few digits multiplied by a 10 to the power of some number, 67 in this case.)

This answer is more than the number of stars in the universe (about 10 followed by 21 zeros).

Put another way: if all 6 billion humans on earth began shuffling cards at 1 deck per minute every day of the year for millions of years, we still wouldn't even be close to finding all possible combinations.



Here's another fact: combinatorics makes your head spin after a while, so that'll do for now.

#76. Control Who Can Edit Tabs

Here's how you can control who can edit individual sheets (tabs) within your Google Sheet with the "Protect sheet" menu.

One of Google Sheets superpowers is its collaborative features.

You can work on a single spreadsheet with many others simultaneously.

It's magical, especially if you're used to working with spreadsheets asynchronously.

Control Access

If you share a Google Sheet with many people, but don't necessarily want everyone to be able to edit everything, then you have plenty of options to control access.

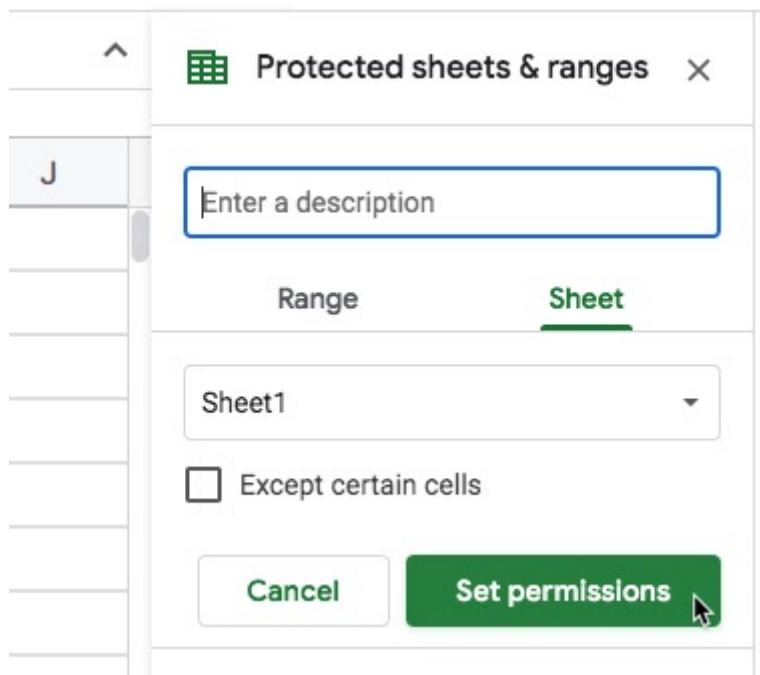
Today, we'll see how to restrict who can edit individual sheets (tabs) within your Google Sheet.

For example, you might have a tab with business data in (e.g. annual department goals) that you don't want anyone to change (accidentally or intentionally).

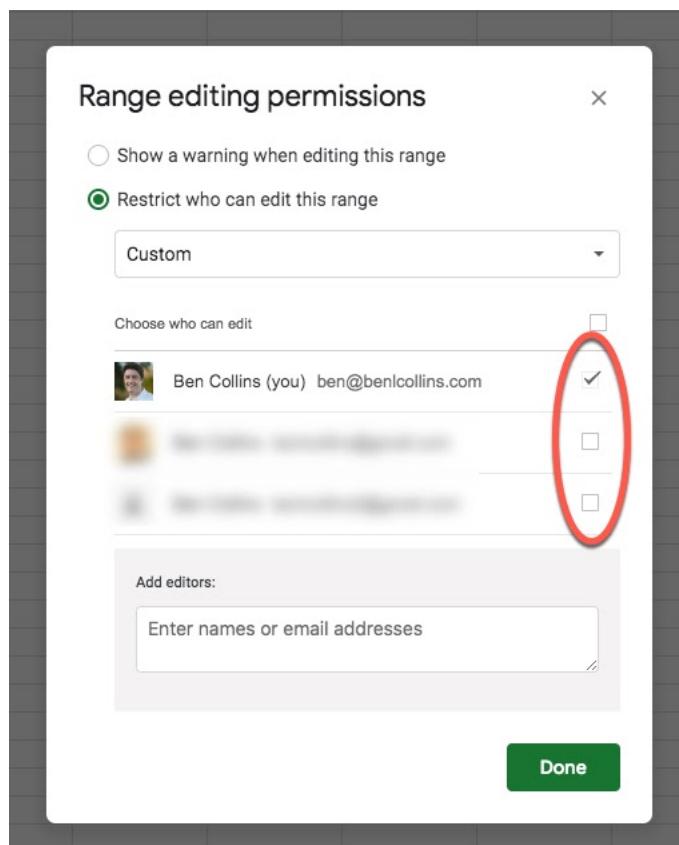
To protect a sheet (tab), right click it and choose "**Protect sheet**"

Or access it through the menu: **Data > Protect sheet**

This opens a popup in the right pane, where you can "**Set permissions**" for your chosen sheet.



Click that and then select the individuals you want to be able to edit the sheet:



In this case, my name is the only one checked, so I'm the only one who can edit the sheet called Sheet1. You could choose multiple people if you wish.

Protected sheets are marked with a small padlock symbol. Others will still be able to see this sheet, just not edit it.

A screenshot of a Google Sheets spreadsheet. The top row has headers A, B, C, and D. Row 1 contains the text "This Sheet is only editable by me, not anyone I've shared it with." Row 3 contains the text "However, those folks can still view it." Row 6 is empty. The bottom navigation bar shows tabs for "Sheet1" and "Sheet2". A red arrow points to the lock icon next to "Sheet1", indicating it is a protected sheet. The formula bar at the top left shows "fx".

	A	B	C	D
1	This Sheet is only editable by me, not anyone I've shared it with.			
2				
3	However, those folks can still view it.			
4				
5				
6				

You can always change or remove Protected ranges and sheets too.

#77. Custom Formulas Data Validation

Here's how to use Custom Formulas in Data Validation to restrict what types of data can be entered into cells.

Data Validation is a technique you can use to improve the quality of data entered into your Sheets.

For example, you could use Data Validation to ensure that only numbers are entered into a column. This way, you won't get text or dates or other funny things going on, which would cause issues with calculations.

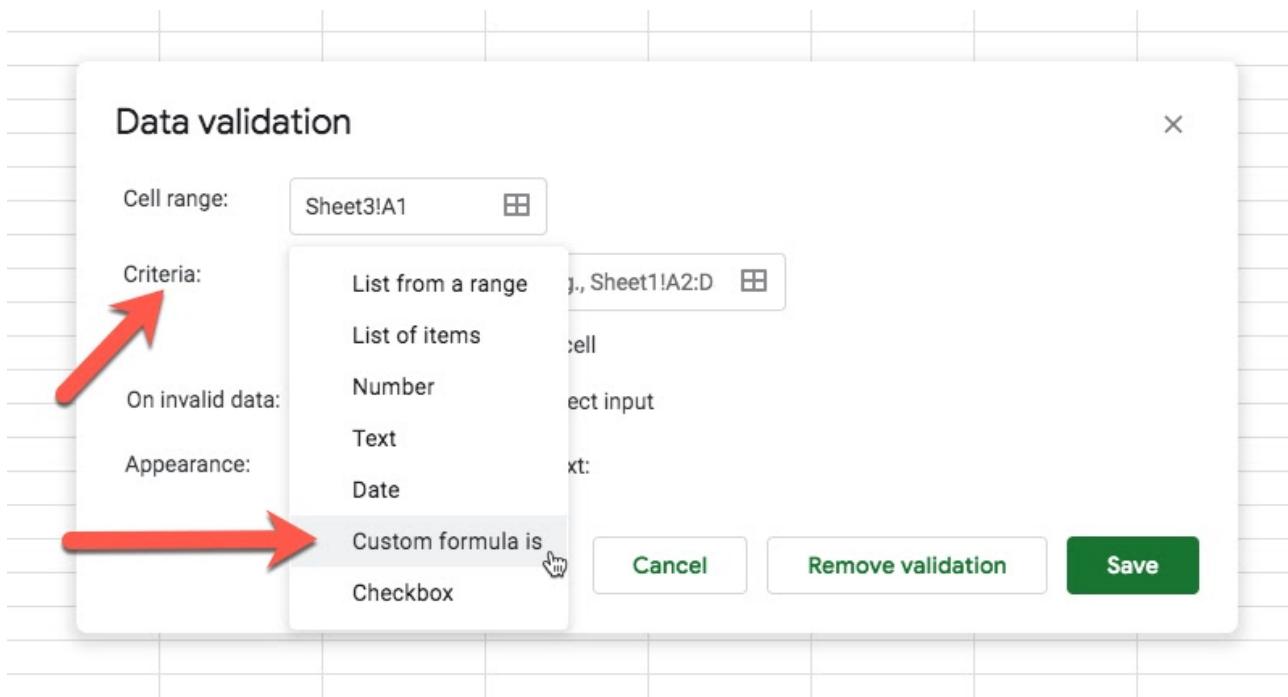
Standard data validation options include numbers, text, dates, lists of items or lists from ranges in your Sheet.

However, you can take it one step further and create your own custom data validation rules using formulas.

Let's see some examples.

Select the cell you want to apply the data validation too. **Right click** to bring up the data validation menu, or choose **Data > Data Validation** from the main menu.

Select the “**Custom formula is**” option from the “**Criteria**” drop down in the data validation editor.

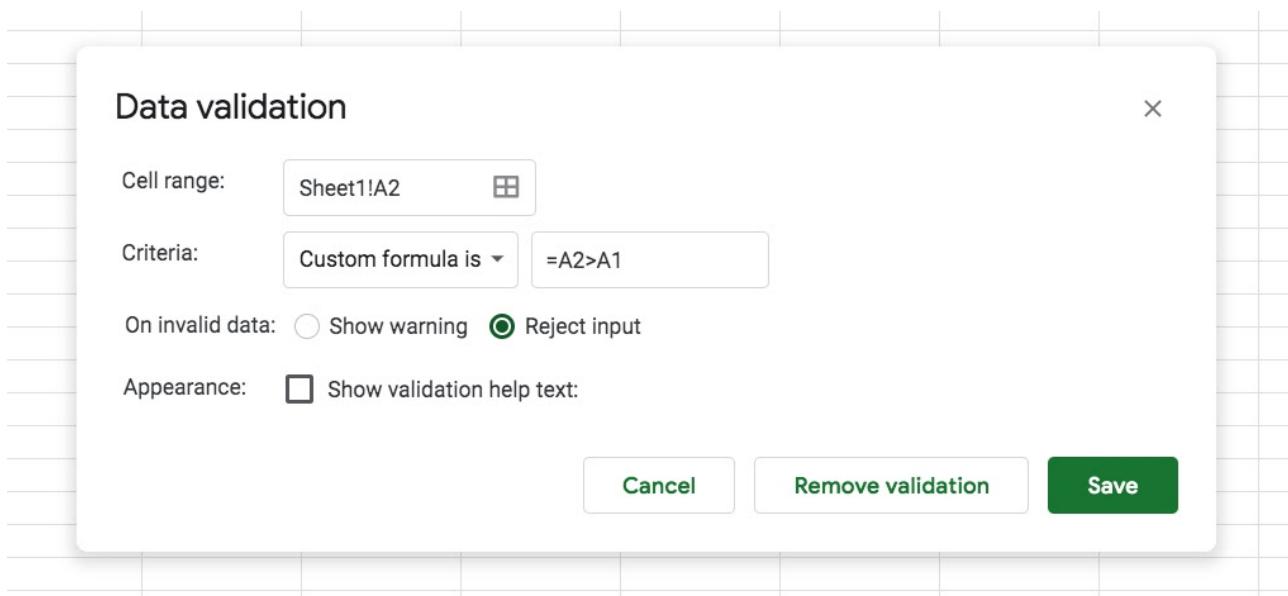


Example 1

Suppose you want each row to have a number entered that is greater than the preceding row.

Use this formula, starting in cell A2:

= A2 > A1

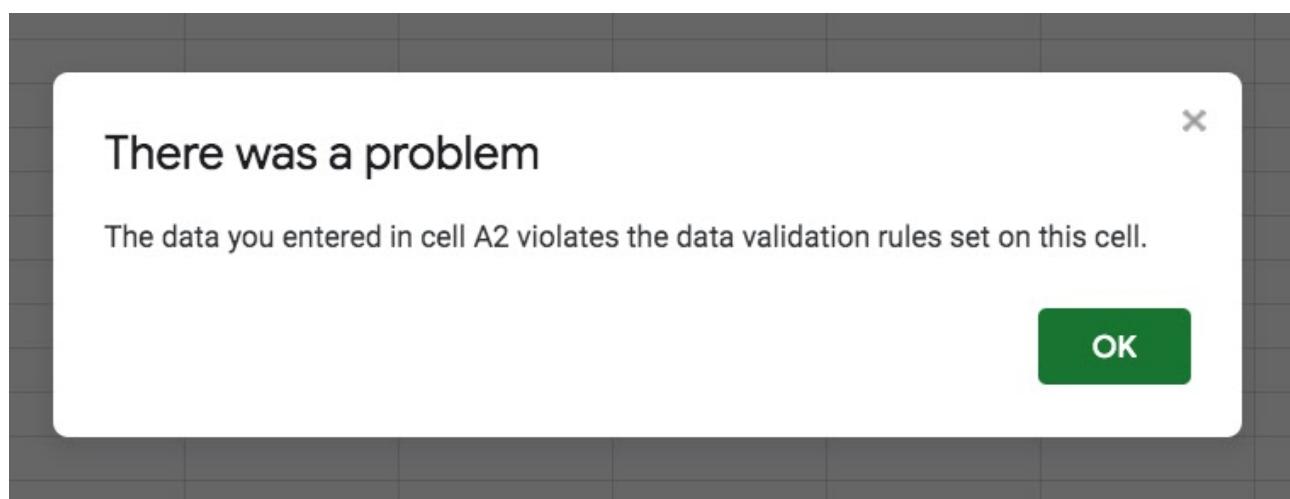


To also ensure that the data is a number, expand the formula to include the ISNUMBER function:

```
= AND( A2 > A1 , ISNUMBER( A2 ) )
```

The value in A2 must be a number that is greater than the value in cell A1. Copy the cell down as far as you need to apply to more cells.

If you try to enter text or a smaller number into cell A2, you'll see this error message:



Example 2

Use this custom formula in the criteria of the data validation to ensure that only a formula can be added into the cell:

```
= ISFORMULA( A1 )
```

Example 3

Here's an interesting one!

Restrict anybody from entering data into a cell until the column width is greater than a specific number of characters. Perhaps you know that this column will contain long text values and you don't want people making it smaller.

Use this formula as your custom formula in the criteria of the data validation:

```
= CELL( "width" , A1 ) > 25
```

Example 4

Check for a valid email address in cell A1 with this formula:

```
= ISEMAIL( A1 )
```

Show Warning or Reject Input?

You can choose to show a warning if the data entered does not pass the validation rule or you can flat out reject it in which case the cell contents won't change.

Validation Help Text

Additionally you can choose to show validation help text when someone clicks on the cell. For example, you might add “Please enter a valid email” to give users some information about what data can be entered in that cell.

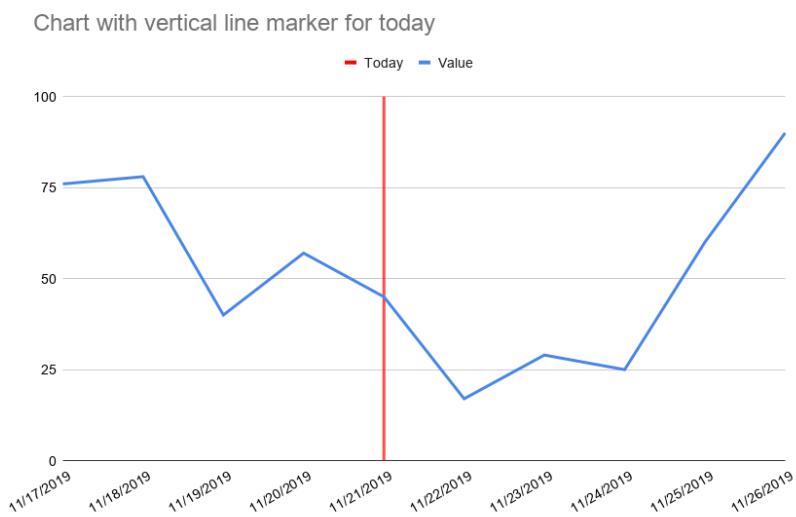
The screenshot shows a Google Sheets interface with a grid of cells labeled A through C and 1 through 6. Cell A1 is highlighted with a blue border. A cursor is positioned over the cell. To the right of the grid, a vertical orange bar indicates the validation status. Below the grid, the following text is displayed:

Validation:
Please enter a valid email.

#78. Line Chart Trick

In this tip, we'll look at how to dynamically annotate a line chart with a vertical line to highlight today's value. Here's how to use formulas to create a helper column AND a helper row. Then use a trendline to annotate line charts in Google Sheets.

We're aiming to add a vertical line (the red one in this image) to a line chart to show where today's value is relative to the rest of the data.



In tip #71, I showed you a basic method for annotating today's value on a line chart, so that it stands out. We did that by adding a dummy series with just a single value for today.

Let's take that a step further and add a full line to highlight today.

Suppose you have this dataset:

	A	B	
1	Date	Value	
2	11/17/2019	76	
3	11/18/2019	78	
4	11/19/2019	40	
5	11/20/2019	57	
6	11/21/2019	45	
7	11/22/2019	17	
8	11/23/2019	29	
9	11/24/2019	25	
10	11/25/2019	60	
11	11/26/2019	90	
12			
13			

To get the line effect we need to display the row for today twice, so that we can trick the chart into letting us use a trendline for that vertical line.

Use this formula in cell D2 to duplicate the row with today's date:

```
= SORT( { A2:A17 ; TODAY() } )
```

Then in cell E2, add this formula and copy it down:

```
= IF( D2 = TODAY() , 0 , "" )
```

And then in cell F2, use a VLOOKUP formula to retrieve the original data from the dataset:

```
= VLOOKUP( D2 , $A$2:$B$17 , 2 , false )
```

Your new dataset will look like this in columns D, E and F:

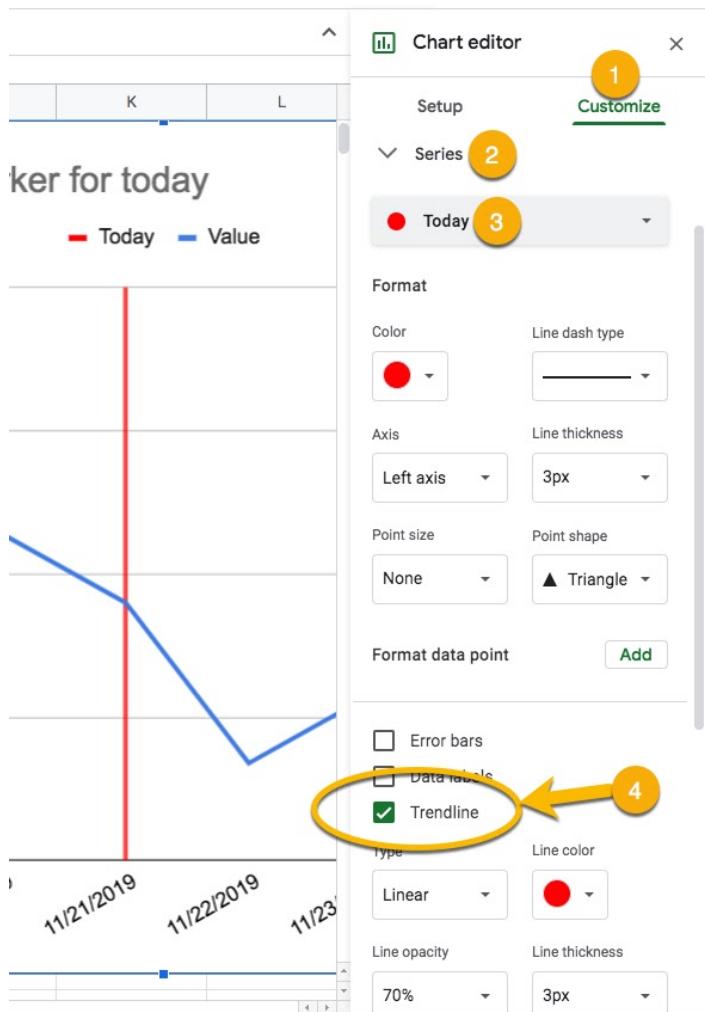
	A	B	C	D	E	F	
1	Date	Value		Extra Date	Today	Value	
2	11/17/2019	76		11/17/2019		76	
3	11/18/2019	78		11/18/2019		78	
4	11/19/2019	40		11/19/2019		40	
5	11/20/2019	57		11/20/2019		57	
6	11/21/2019	45		11/21/2019	0	45	
7	11/22/2019	17		11/21/2019	0	45	
8	11/23/2019	29		11/22/2019		17	
9	11/24/2019	25		11/23/2019		29	
10	11/25/2019	60		11/24/2019		25	
11	11/26/2019	90		11/25/2019		60	
12				11/26/2019		90	
13							

Look closely and you'll see the row for today's date is now repeated twice (courtesy of the { A2:A17 ; TODAY() } array construction. The SORT wrapper ensures they're in the correct order.).

Now you can highlight columns D, E and F and create a line chart with this data.

Now this is where the trick comes in.

Under the **Customize menu** (1), choose **Series** (2) and then the "**Today**" series (3). Add a **trendline** (4) to that series and it'll give you the vertical line:



The nice thing about this method is that it's dynamic, so the vertical line will update as today becomes tomorrow, i.e. it'll keep pace with the correct date.

Feel free to make your own copy of the [Google Sheet Tip 78 template](#).

(Note: I've set my file sharings to allow anyone with the link to view this file. You may not be able to open this file because it's from an outside organization, and my G Suite domain is not whitelisted at your organization. You may be able to ask your G Suite administrator about this.)

In the meantime, feel free to open in an incognito window and you should be able to view it.)

#79. Find Every Formula

Excel has a nifty feature called GoTo, which lets you easily jump around your spreadsheets and find specific information or items.

Google Sheets does not have this functionality, but we can use alternative techniques to achieve the same goal.

Let's look at how to use conditional formatting to highlight every formula in your sheet.

Suppose your Google Sheet is full of formulas and you want to see where they all are, so you can check them. You don't want to simply look through the Sheet because you might miss some.

Instead, here's a method to highlight them all for you.

Click on the box in the top left corner, between the column A and the row 1 headings. This will highlight your entire Sheet.

	Sector	Region	Market
1	Sector	Region	Market
2	Utilities	Midwest	Cleveland
3	Training	Northeast	New York
4	Training	South	Houston
5	Training	Midwest	Louisville
6	Consulting	Midwest	Cleveland

Then choose **Format > Conditional Formatting** from the menu.

Select "**Custom formula is**" under the Format rules.

Set the formula to:

```
=ISFORMULA( A1 )
```

Change the formatting style if you wish. I like to use a bright yellow.

Click Done and all the formulas in your Sheet will be highlighted, making them much easier to see.

	<i>fx</i> =sum(A2:A8)			
	A	B	C	D
1	Quantity	Revenue	Profit	Cost
2	\$300	\$6,267	\$2,541	\$3,726
3	\$100	\$2,401	\$1,022	\$1,379
4	\$500	\$9,345	\$4,235	\$5,110
5	\$600	\$11,628	\$5,082	\$6,546
6	\$100	\$2,042	\$984	\$1,058
7	\$200	\$3,552	\$1,694	\$1,858
8	\$800	\$14,440	\$6,776	\$7,664
9	\$2,600	\$49,675	\$22,334	\$27,341
10				
11				

You can use this same technique with other conditional formatting rules like ISERROR to find all your errors.

Thanks to reader Robert B. who shared this idea with me!

#80. Simple Formula Clock

In this tip, we'll build a simple digital clock in a Google Sheet. We'll subtract the TODAY function from the NOW function to get the current time.

In a blank Google Sheet, add this formula to cell A1:

```
=TEXT( NOW() , "hh:mm am/pm" )
```

The NOW() part returns the date and time.

The TEXT function formats this answer as hours and minutes with an AM/PM designation.

To make it update, go to **File > Spreadsheet** settings, and set the spreadsheet **calculation** settings to be **“On change and every minute”**.

#81. Hyperlinked Image

Here's how to combine the IMAGE and HYPERLINK functions to create clickable images.

This is a fun tip that came up at a in-person workshop I taught recently.

The question was: can I create a hyperlinked image?

Answer: yes, you can!

Simply drop the IMAGE function into the link label argument of the HYPERLINK function.

Here's an example of the National Christmas tree in the White House garden:

	<i>fx</i>	=HYPERLINK("https://www.nps.gov/whho/planyourvisit/national-christmas-tree.htm",IMAGE("https://live.staticflickr.com/4901/32235994688_11c5129ed5_m.jpg"))
1	A	
2	B	https://www.nps.gov/whho/planyourvisit/national-christmas-tree.htm 

Clicking the image link will take you to the National Park Service info page about the tree.

Here's another, pointing to Google's homepage:

fx	=HYPERLINK("https://www.google.com/", IMAGE("https://www.google.com/favicon.ico"))	
A	B	C
1		
2	https://www.google.com/	
3		

It's a little easier to see the formula because the URLs are shorter.

```
=HYPERLINK( "https://www.google.com/" , IMAGE( "https://www.google.com/favicon.ico" ) )
```

Could be a nice way to add some visual elements to your Sheets!

#82. Dates Are Really Just Numbers

Dates in spreadsheets -- and this applies in Excel as well as in Google Sheets -- are actually just numeric values with a special date format applied.

So today, 12/30/2019, is actually stored by Google Sheets as the number 43,829.

Try it:

Put today's date in a cell (shortcut: Ctrl + ;) and change the formatting of the cell to a number format.

These numbers are known as **serial numbers**. Each day is represented by one whole number.

The number 1 corresponds to 12/31/1899 in Google Sheets (but 1/1/1900 in Excel). So today's number tells us that 43,829 days have elapsed since then.

And if I add the time, then 12/30/19 at 11am is 43,829.45833, where the decimal places represent the time, a percent value of how far through this 24 hour period we are.

Here are two columns of identical numbers, one formatted as numbers, the other formatted as dates:

	A	B
1	Number Format	Corresponding Date
2	-1	12/29/1899
3	0	12/30/1899
4	1	12/31/1899
5	2	1/1/1900
6	43829	12/30/2019
7	43829.25235	12/30/2019 6:03:23
8		

Once you understand this fact about dates and how spreadsheets really "see" dates, it makes it much easier to work with them. When you subtract two dates, all you're really doing is subtracting two numbers.

#83. Working With Excel Files

Did you know you can work with Excel files directly in Google Sheets without converting them?

Have you ever been sent Excel files from colleagues or external partners?

I'm sure you have.

Perhaps you imported the data into your Google Sheet, or opened the Excel and simply copy-pasted the data directly.

Then you did your work, slicing and dicing that data in Google Sheets. Finally you downloaded your Sheet as a new Excel file to send back to your contact (who doesn't use Sheets).

However, there's a better workflow when you know that the file begins and ends in Excel.

You can open the Excel file in Google Sheets and work on it, without converting it into a Google Sheets file.

That's right, it's still an Excel file but it looks and feels like a Google Sheet. So you have all the functionality you're familiar with. And it auto-saves as the original Excel.

How to open an Excel file but not convert it:

This works for Excel files saved in your Google Drive.

From your Drive folder, double-click the Excel file you want to open.

It will open in Preview mode. At the top of the preview window, there's a button called "Open with Google Sheets". Click it.

It opens in Sheets and looks like a regular Google Sheet but you'll notice an XLSX flag at the top to tell you that you're working on the Excel file.

The screenshot shows the Microsoft Excel interface with the file 'exampleData.xlsx' open. A red arrow points to the file extension '.xlsx' in the title bar. The spreadsheet displays the following data:

	A	B	C	D	E
1	userid	lastlogin	joined_at	course_count	src
2	148377	11/27/2019	12/9/2018		8
3	582020	11/26/2019	7/10/2019		9
4	144464	11/19/2019	2/14/2018		8

Your changes are autosaved directly into the original Excel.

Note that not all the functionality transfers from Google Sheets to Excel, and vice versa. For example, slicers and scripts are both implemented differently and won't transfer between the tools.

#84. Colored Checkboxes

Here's how you can use Checkboxes and Conditional Formatting to change the color of "checked" rows.

Checkboxes are an easy way to add some interactivity to your Google Sheets.

Users of the Sheet can check or uncheck the box to cause an action to happen.

A classic example would be a To-Do list, where the check indicates that a task is complete. You can improve this visually by combining the checkbox with conditional formatting to change the color of rows that have been checked.

Suppose you have this basic To-Do list:

	<i>fx</i>	FALSE	
	A	B	
1	Task	Complete	
2	Task 1	<input type="checkbox"/>	
3	Task 2	<input type="checkbox"/>	
4	Task 3	<input type="checkbox"/>	
5	Task 4	<input type="checkbox"/>	
6	Task 5	<input type="checkbox"/>	
7			

You add checkboxes by either **right-clicking** on a cell and selecting **Data Validation**, or from the menu **Data > Data Validation**. Then choose **Checkbox** from the Criteria drop-down menu.

Now you can add conditional formatting to highlight completed rows, and make comprehension of the data quicker:

fx | TRUE

	A	B	
1	Task	Complete	
2	Task 1	✓	
3	Task 2	✓	
4	Task 3	<input type="checkbox"/>	
5	Task 4	<input type="checkbox"/>	
6	Task 5	<input type="checkbox"/>	
7			

To achieve this, go to the menu **Format > Conditional formatting**

Set the Format Rules to be **Custom formula is**

And enter this formula:

```
=${B2}=TRUE
```

In this example, I've set the format to red background color, dark red text and strikethrough.

This will apply the conditional formatting to the entire row.

There are a couple of important things to note:

We refer to column "B" because the checkboxes are in column B. We use row 2 because our data starts on row 2 (we omit the headers in row 1). The crucial detail is the "\$" before the B, which locks the conditional test to column B for each row. So the conditional formatting only tests the custom formula for column B, but applies it to the whole of each row.

The test is whether the value equals TRUE, which corresponds to the checkbox being checked. If it's checked then the formatting is applied.

Unchecked checkboxes have a value of FALSE.

#85. TRANSPOSE Function: Part I

This is a nice, easy tip - plus here's another Formula Challenge. (You might find this tip useful for this challenge!)

The TRANSPOSE function is a handy function that changes data from rows into columns, or vice versa.

Suppose you have a column of data in the range A1:A10. This formula (in cell B1) will show that data as a row:

```
=TRANSPOSE(A1:A10)
```

	A	B	C	D	E
1	1	1	2	3	4
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				

It works with tables of data too, so it's a quick way to re-orientate your datasets. You can also nest the TRANSPOSE function inside of other functions, which is useful when you're doing data wrangling.

All in all, it's a really useful function to know about!

Formula Challenge #3

Ok, are you ready for this?

Suppose you have a list of words in a single cell, separated by commas and **not** in alphabetical order, like so:

Epsilon,Alpha,Gamma,Delta,Beta

	A	B
1	Epsilon,Alpha,Gamma,Delta,Beta	
2		
3		

Your challenge is to create a single formula (i.e. in a single cell) that reorders this list into alphabetical order, to give an output like this (in the yellow cell):

Alpha,Beta,Delta,Epsilon,Gamma

	A	B
1	Epsilon,Alpha,Gamma,Delta,Beta	
2		
3	Alpha,Beta,Delta,Epsilon,Gamma	
4		
5		

I'll share answers in the next tip!

#86. TRANSPOSE Function: Part II

So, in tip #85 I challenged you to find a single formula that could alphabetically sort a list of words in a single cell.

We'll build this solution in steps, and you'll see that it's not too difficult!

Start with a list of words in a single cell, separated by commas and not in alphabetical order, like so:

Epsilon,Alpha,Gamma,Delta,Beta

	A	B
1	Epsilon,Alpha,Gamma,Delta,Beta	
2		
3		

Your challenge was to create a single formula (i.e. in a single cell) that reorders this list into alphabetical order.

Formula Challenge #3 Solution

Step 1

Use the SPLIT function to separate the comma-delimited string into separate cells.

```
=SPLIT(A1, ", ")
```

(Split has two additional arguments and you have to be precise with your delimiter. In this simple example, we can omit the two additional arguments.)

Step 2

Use the TRANSPOSE function from last week's tip to change from row orientation to a column orientation, so that we can sort in Step 3.

```
=TRANSPOSE(SPLIT(A1, ", "))
```

Step 3

Sort the data with the SORT function!

You don't need to specify a column or direction, because we only have 1 column and we want ascending order, which is the default direction. This keeps our formula brief.

```
=SORT(TRANSPOSE(SPLIT(A1, ", ")))
```

Step 4

Finally, join the column back together with the JOIN function, again using a comma as the delimiter. There's no need to use a second transpose because the JOIN function works equally well with a column of data as it does with a row of data!

```
=JOIN(", ", SORT(TRANSPOSE(SPLIT(A1, ", "))))
```

Bingo!

	A	B
1	Epsilon,Alpha,Gamma,Delta,Beta	
2		
3	Alpha,Beta,Delta,Epsilon,Gamma	
4		

#87. Create Data With Randarray

Here's how to use the RANDARRAY function to generate data automatically.

Have you ever needed to create a quick table of values? Perhaps to test out your formulas or pivot tables, or maybe to teach a concept for which you need fictitious data.

It's tedious to create manually.

Instead, you can create it with the RANDARRAY function.

Step 1: Create a random array of numbers in your Google Sheet with the RANDARRAY function. Specify the size of your dataset with a number of rows and columns:

```
= RANDARRAY( 3 , 4 )
```

Step 2: To get whole numbers multiply by 100 (or some other scaling factor) and use the ROUND function to turn into whole numbers:

```
= ROUND( RANDARRAY( 3 , 4 ) * 100 )
```

Step 3: Turn into an Array Formula to fully expand again, by pressing Ctrl + Shift + Enter (or Cmd + Shift + Enter on a Mac):

```
= ArrayFormula( ROUND( RANDARRAY( 3 , 4 ) * 100 ) )
```

The output of this formula is:

<i>fx</i>	<code>=ArrayFormula(round(randarray(3,4)*100))</code>					
	A	B	C	D	E	
1		Customer 1	Customer 2	Customer 3	Customer 4	
2	Product 1	80	39	38	40	
3	Product 2	12	38	98	16	
4	Product 3	50	84	42	87	
5						

The RANDARRAY function is a volatile function, meaning it will recalculate every time you make a change in your Sheet.

To avoid this, highlight the whole dataset created by the RANDARRAY, copy it (Ctrl + C or Cmd + C) and paste-special as values (Ctrl + Shift + V or Cmd + Shift + V) over the top. This converts the array formula into a dataset of hard-coded values, which don't change.

#88. SWITCH Function

The SWITCH function is a useful tool for categorizing data. In the right circumstances, it can save you from messy, nested IF statements.

The SWITCH function is used to test an expression against a list of cases. It returns a value when the expression is EQUAL to one of the cases.

It has some similarities to the IFS function, but differs because SWITCH tests for exact matching rather than whether a condition is true (e.g. $X > Y$). SWITCH also has a default option to return a value if no match is found.

Let's see an example. Suppose we have this data set of student grades and we want to add context to each grade level.

	A	B	
1	Student	Grade	
2	Student 1	A	
3	Student 2	A	
4	Student 3	C	
5	Student 4	B	
6	Student 5	B	
7	Student 6	A	
8	Student 7	E	
9	Student 8	E	
10	Student 9	A	
11	Student 10	C	
12			
13			

Use this SWITCH function to categorize these grades:

```
=SWITCH (
    B2,
    "A", "Top marks! Great job!",
    "B", "Keep up the good work",
    "C", "Could do better",
    "Failed. Remedial study needed"
)
```

Let's break it down:

B2 is the condition we're going to test. It's the grade letter from column B and it's the input to our SWITCH function.

We check the value of B2 against the first case "A". If they match (i.e. the grade in B2 was also "A") then SWITCH returns the string "Top marks! Great job!".

If the value from B2 doesn't match "A", we move on and test it against "B" and if that fails, test against "C".

If the condition doesn't match any of the cases, the last string is returned: "Failed. Remedial study needed". It's our catch-all solution. Anything that's not "A", "B" and "C" will return this answer.

This is what the solution looks like:

	A	B	C
1	Student	Grade	Outcome
2	Student 1	A	Top marks! Great job!
3	Student 2	A	Top marks! Great job!
4	Student 3	C	Could do better
5	Student 4	B	Keep up the good work
6	Student 5	B	Keep up the good work
7	Student 6	A	Top marks! Great job!

Give it a try and SWITCH things up!

#89. Upgrade Your Apps Script!

In early 2020, Google announced the launch of the V8 runtime for Apps Script, which is the same runtime environment that powers Chrome.

(A runtime environment is the engine that interprets your code and executes the instructions.)

This V8 engine means our scripts run much faster and allow us to take advantage of all the modern JavaScript features. Here's how to switch your Apps Script to V8,

When you open the Apps Script editor (**Tools > Script Editor**), you'll see a yellow notification bar at the top of your editor window prompting you to enable the new V8 runtime.

You can also select from the menu: **Run > Enable new Apps Script runtime powered by V8**

With V8 enabled, your Apps Script code will run much faster.

In addition, you can use modern JavaScript syntax in your code, like:

- Default parameters
- Better multi-line strings
- **let** and **const** keywords
- Arrow functions
- [and much, much more...!](#)

Here's one example, showing how you can now specify default parameters when defining functions (the bits highlighted in yellow are new in V8):

```
function addNumbers(x = 1, y = 2) {
  Logger.log(x + y);
}
```

The function **addNumbers** simply logs the value of $x + y$.

If we don't tell the function what the values of x and y are, it uses the defaults we've set (x is 1 and y is 2) and outputs an answer of 3.

#90. Rolling Average Formula

The INDEX function is one of the most useful and versatile spreadsheet functions. It's a powerful way to access your data.

And you know what's even better than an index function? Two index functions!

Here's how to use the INDEX():INDEX() formula trick to get data dynamically.

This tip is one of the most challenging tips I've shared but it's worth the effort to learn.

The index function has a curious, hidden property that's extremely handy.

If you put the index function into a range reference it returns a valid cell address, which means you can build dynamic ranges of data.

Let's see an example.

Suppose we have a column of values -- perhaps it's recent transaction values or new leads each day -- and we want to compute a rolling 7-day average.

Double Index Trick

We start by using this double index trick to extract the last 7 values from the list.

Whenever new data is added, the double index formula includes the new data and "rolls" down the range.

Here's the formula to extract the last 7 values from column A:

```
=INDEX(A2:A, COUNTA(A2:A) - 6) : INDEX(A2:A, COUNTA(A2:A))
```

	A	B	C	D	
1	Values				
2	0		7		
3	16		20		
4	91		37		
5	73		46		
6	63		204		
7	50		78		
8	20		57		
9	25				
10	7				
11	20				
12	37				
13	46				
14	204				
15	78				
16	57				
17					
18					
19					

In the image you can see that the double index formula in cell C2 has returned the last 7 values from column A. If we add additional data to column A then the formula will update to show the last 7 values.

(You can change the number 6 to something else if you want a different period.)

Can you see how it works?

The COUNTA() simply counts how many values we have in column A.

The INDEX(A2:A,COUNTA(A2:A)) returns the value at the position we specify. The first INDEX gets the value 7th from the bottom and the second index gets the bottom value.

However, per the index trick above, when we put an INDEX function next to a colon ":" in a range reference it returns the cell address instead of the value! Wow!!

So the two index functions return cell address A10 and A16 respectively, instead of the values in those cells! This gives the range reference A10:A16 which returns the 7 values for us.

This formula is equivalent to saying:

```
=ArrayFormula (A10:A16)
```

Except that the double index keeps "rolling" with the data to get the last 7 values as new data is added.

Rolling Average

The double index formula can be used to create a rolling average, by wrapping this double index with a standard AVERAGE function:

```
=AVERAGE (INDEX (A2:A, COUNTA (A2:A) - 6) : INDEX (A2:A, COUNTA (A2:A) ))
```

	A	B	C	D
1	Values	Rolling 7-day average		
2	0		81	
3	16			
4	91			
5	73			
6	63			
7	50			
8	20			
9	25			
10	7			
11	20			
12	37			
13	46			
14	204			
15	78			
16	57			
17	36			
18	41			
19	78			
20	75			
21				
22				

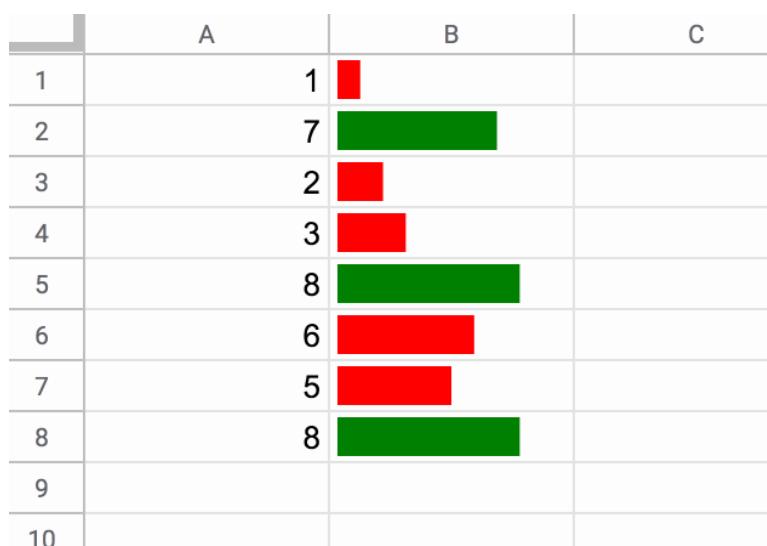
#91. Conditional Sparklines

The SPARKLINE function is one of my favorite functions in Google Sheets, so I'm excited to share this tip. I'll show you how to nest an IF function inside the SPARKLINE to create a conditional sparkline that changes color based on data.

[Sparklines](#) are small charts that exist inside a single cell. They're created with the SPARKLINE function in Google Sheets.

Let's see how to create a sparkline that changes color based on the underlying data.

In this scenario, the values in column A vary between 1 and 10 and I want values greater than a threshold value of 6 to turn green:



The formula in column B to create these charts is:

```
=SPARKLINE( A1 , {"charttype","bar" ; "max",10 ; "color1",
IF(A1<7,"red","green") })
```

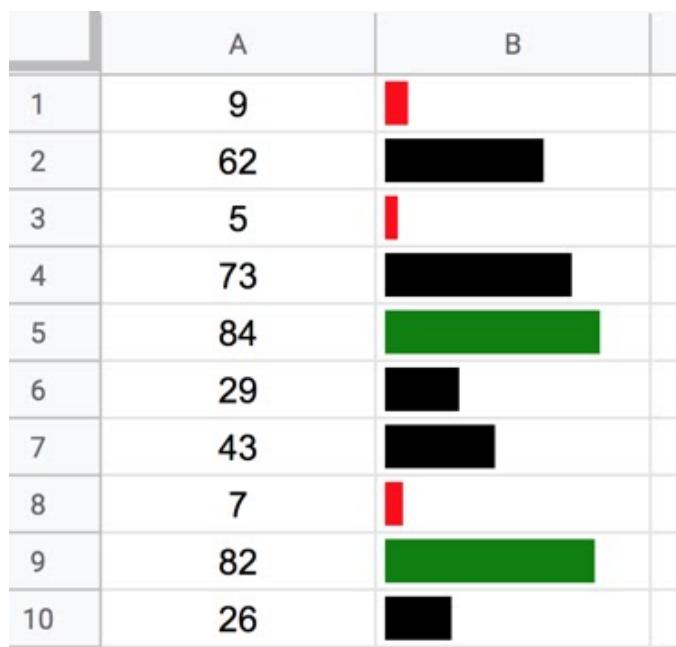
You can see the IF function inserted as the option for the "color1" setting of the sparkline bar chart.

Feel free to modify this IF statement with different threshold values and/or colors (you may need to adjust the "max" setting as well).

If you want to have multiple tiers of colors, you can nest an IFS function instead of a single IF, for example:

```
=SPARKLINE(A1, {"charttype", "bar" ; "max", 100 ; "color1", IFS(A1<25 , "red" , A1<75 , "black" , A1<=100 , "green") })
```

This formula, applied to numbers between 1 and 100, gives an output like this:



#92. Trace Precedents With F2

The F2 key can be used to trace precedents in formulas.

Press the F2 key to enter the "formula view" of a cell with a formula in. You're probably familiar with that shortcut key.

Google Sheets highlights ranges in your formula expression and in your actual Sheet with matching colors. It applies different colors to each unique range in your formula.

	A	B	C	D	E	F
1	A	1				
2	B	2				
3	C	3				
4	D	4				
5	E	5				
6	F	6				
7	G	7				
8	H	8				
9						
10						

However, it has another useful property too.

If you position your cursor over a range of data in your formula and then press the F2 key again, it will highlight that specific range of data for you (even if it's in a different tab):

A screenshot of a Google Sheets interface. The formula bar at the top shows =vlookup(E7,Sheet1!A1:D5,2,false). The main area of the sheet has rows 4 through 12. Row 7 contains the formula =vlookup(E7,Sheet1!A1:D5,2,false) in cell E7. To the right of this formula, under the column headers 'Lookup Value' and 'Result', there is a table with one row. The 'Lookup Value' cell contains 'West' and the 'Result' cell contains '11'. A blue rectangular selection highlights the entire range from E7 to F11. Below the sheet area, the navigation bar shows 'Sheet1' and 'Sheet2'.

In this example, the VLOOKUP was on Sheet2 but the data was on Sheet1. When I press F2 the first time it highlights the ranges in the formula but I still don't know where the lookup table is. So I hover over the lookup table reference and press F2 again. Voila! It takes me right there.

It's a handy feature when you're auditing Sheets or working with someone else's formulas. It helps you understand where the underlying data is.

#93. Group Columns Trick

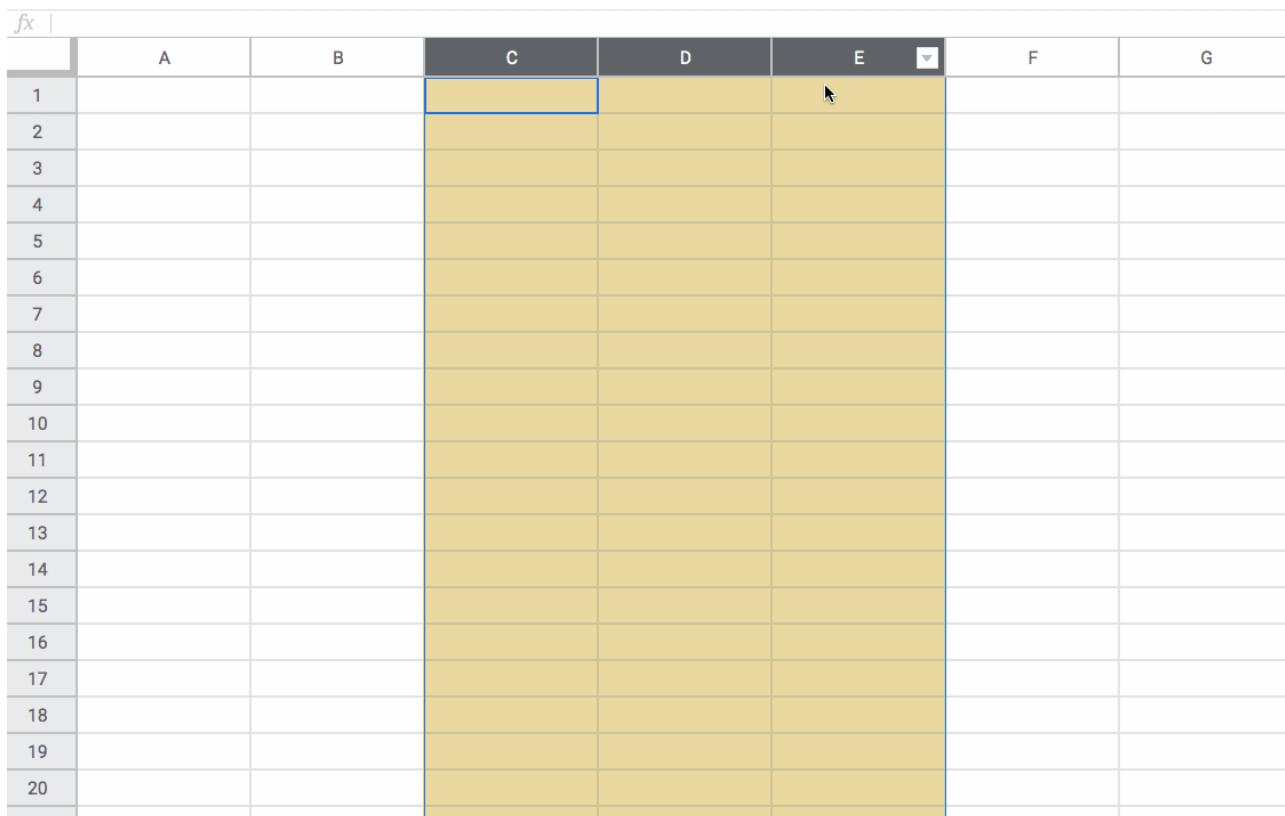
Use the group columns feature to build "sidebars" in your Google Sheets.

I'm sure most of you know that you can group rows or columns in Google Sheets. (It was featured way back as tip #11!)

If you highlight several columns, you can right click and choose to Group columns. This adds a button above the columns, which you can toggle to show/hide these grouped columns. (It works for rows as well.)

You can set the +/- toggle button to be on the left or right of the group by right-clicking on the +/- button and selecting left or right.

Here's an example:



The screenshot shows a Google Sheets interface with a light gray background. At the top, there is a dark header bar with the letters A through G. Below this, the first two rows are labeled with numbers 1 and 2. Rows 3 through 21 are labeled with numbers 3 through 20. The columns are labeled A, B, C, D, E, F, and G. Columns C and D are highlighted with a yellow background, indicating they are selected. A blue border surrounds the cells in columns C and D from row 1 to row 21. To the right of column E, there is a small dropdown arrow icon. The rest of the spreadsheet is empty with no visible data.

Knowing this, let's see how you can add a sidebar to your Google Sheets.

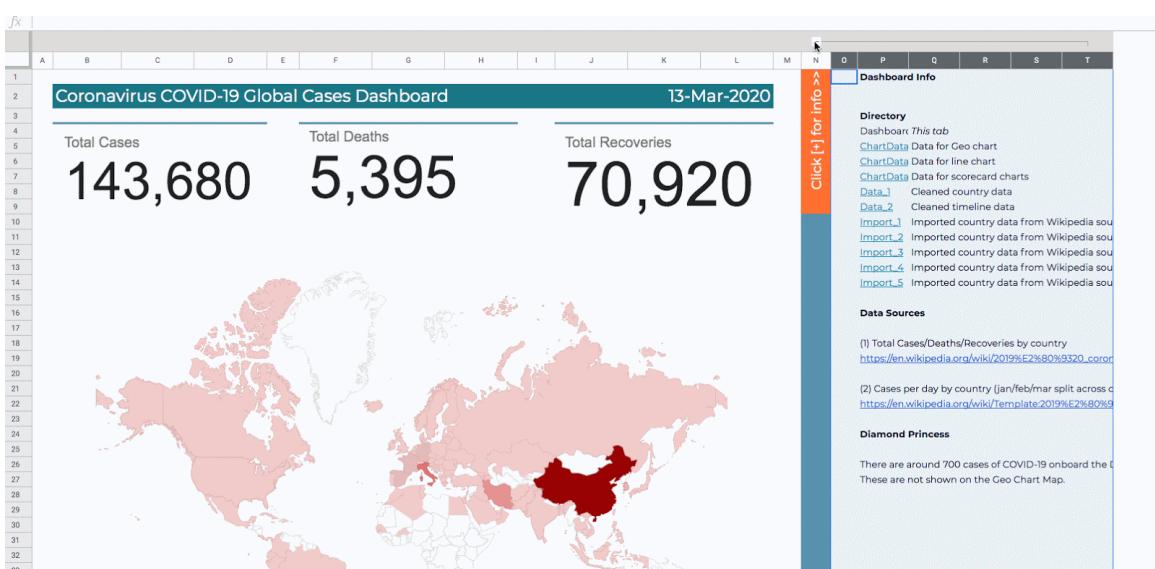
Add or select columns to the right (or left) side of your Sheets. Group them.

Next, add the text "Click [+] for info" into the top cell next to your grouped columns.

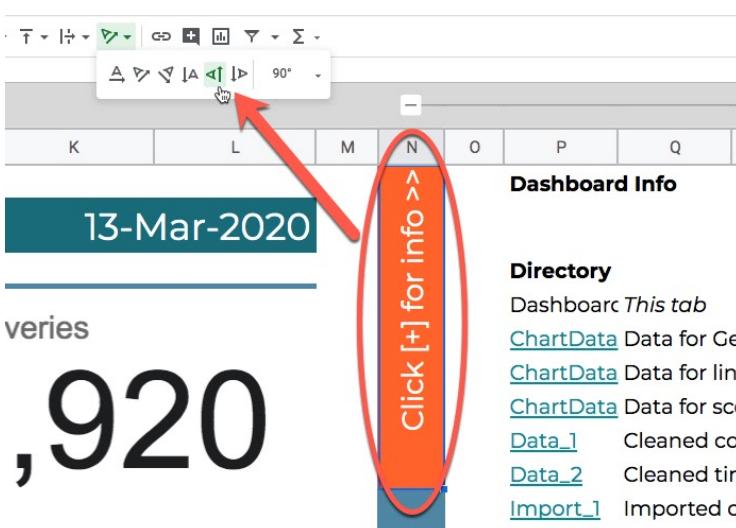
Rotate that text to make it vertical and then merge cells with the rows beneath so it has space to be fully shown.

Change the background color of this column so that it stands out.

And voilà! It's a really neat way to tidy up your Google Sheets!



Here's a zoomed in view of the vertical text bar:



#94. Organize Tabs

Use separator tabs to keep your Google Sheets organized.

I had a question recently about whether it's possible to group tabs in a Google Sheet file, to keep things organized when you have a lot of them.

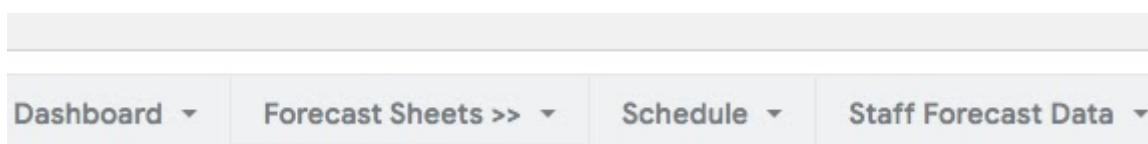
(A "tab" in an individual Google Sheet file is also called a "Sheet", but here I've used the term tabs to avoid confusion.)

Unfortunately it's not possible to group tabs, but that doesn't mean you can't do anything to organize them.

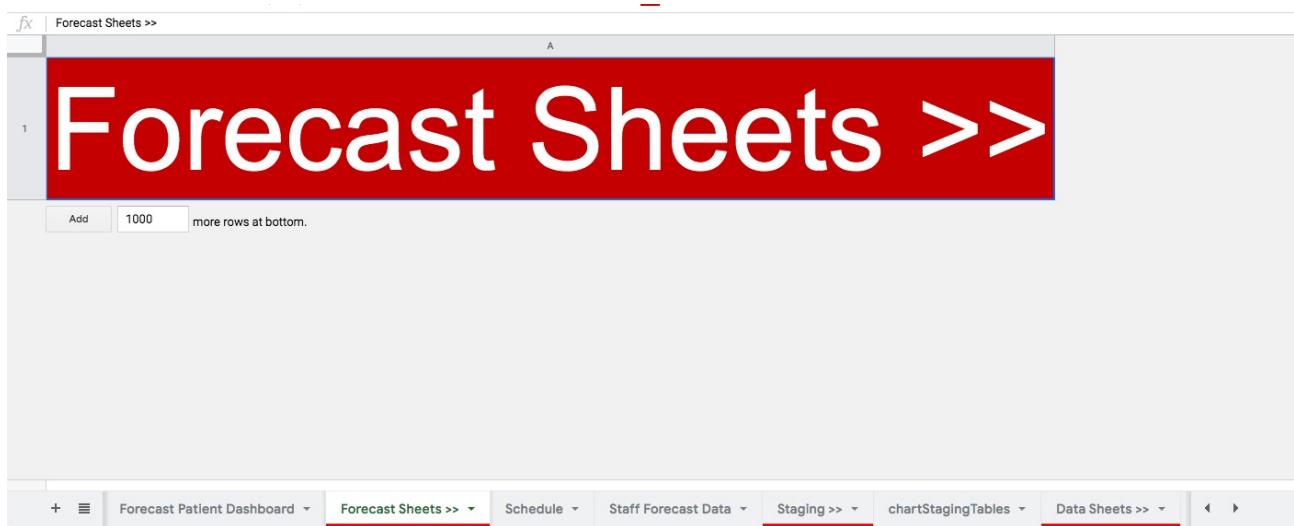
Here's a trick I use to keep multiple tabs organized:

1. I insert a tab between the "groups" of tabs to create a breakpoint that's easy to find.
2. Name this tab with a reference to the "group" of tabs that follow. My convention is to add some arrows to indicate it's an organizer tab e.g. **Forecast Sheets >>**
3. Add a color to that tab.

It looks something like this in practice:



For the tab itself, I delete all rows and columns except A1. In cell A1, I add information about the group that matches the tab name. I also use the same color scheme (click to enlarge):



It's simple but it works surprisingly well to keep your larger Google Sheets organized.

#95. Pivot Date Trick

Here's how to create pivot date groups from dates in your pivot tables.

For years, Google has been adding features to pivot tables in Google Sheets to increase their functionality and bring them closer to their Excel counterparts.

In this tip, we'll look at date grouping, which is hugely useful and a valuable addition to Sheets pivot tables.

Suppose you have a table of data with a column of dates and you're interested in summarizing that daily data at a higher level, e.g. by month.

In this example, we have website pageviews on a daily basis (with the dates in US format dd/mm/yy):

	A	B
1	Date	Pageviews
2	1/1/20	1419
3	1/2/20	2,379
4	1/3/20	1,956
5	1/4/20	1,048
6	1/5/20	1,338
7	1/6/20	2,493
8	1/7/20	2,617
9	1/8/20	2,777
10	1/9/20	2,822
11	1/10/20	2,427

It's rather cumbersome to summarize with formulas but it's just a few clicks in a pivot table.

Step 1: create your pivot table (menu: **Data > Pivot table**).

Step 2: add the date column to the rows section of your pivot table and add a value (pageviews in this case)

Step 3: right click on any date in the pivot table and choose "Create pivot date group" and select the grouping level you want

The screenshot shows a Google Sheets interface with a pivot table. The table has two columns: 'Date' (A) and 'SUM of Pageviews' (B). Row 4 is selected, showing the date '1/3/20'. A context menu is open over this cell, listing various date grouping options. The 'Day of the week' option is highlighted.

	A	B
1	Date	SUM of Pageviews
2	1/1/20	1419
3	1/2/20	2,379
4	1/3/20	1,956
5	1/4	Cut ⌘X
6	1/5	Copy ⌘C
7	1/6	Paste ⌘V
8	1/7	Paste special ▶
9	1/8	Insert column
10	1/9	Create pivot date group ▶
11	1/10	Get link to this cell
12	1/11	
13	1/12	

The context menu options include:

- Second
- Minute
- Hour
- Hour-Minute (24 hour)
- Hour-Minute (12 hour)
- Day of the week** (highlighted)
- Day of the year
- Day of the month
- Day-Month
- Month
- Quarter
- Year
- Year-Month
- Year-Quarter
- Year-Month-Day

The individual dates will collapse down to the date groups you've chosen.

For example, here I've chosen the "Day of the week" group and elected to show an average pageview for each day of the week:

	A	B	
1	<i>Date - Day of the week</i>	AVERAGE of Pageviews	
2	Sunday	1,397	
3	Monday	2,735	
4	Tuesday	2,941	
5	Wednesday	2,766	
6	Thursday	2,769	
7	Friday	2,412	
8	Saturday	1,303	
9			

It's as easy as that!

Lastly, there's one nuance to keep in mind. Toggling the "Show totals" checkbox for the rows of dates will undo the grouping you've set. You can add it right back, but it's easier to remove the totals before grouping to avoid this behavior.

#96. Pivot Custom Text Groups

Here's how to create custom text groups in your pivot tables.

Following on from the last tip about custom date grouping in pivot tables, let's look at another custom grouping option available in pivot tables.

This time, suppose you have a table of data and you're interested in summarizing that data in a highly specific way.

For example, you might want to group transaction data into geographic areas for particular sales reps.

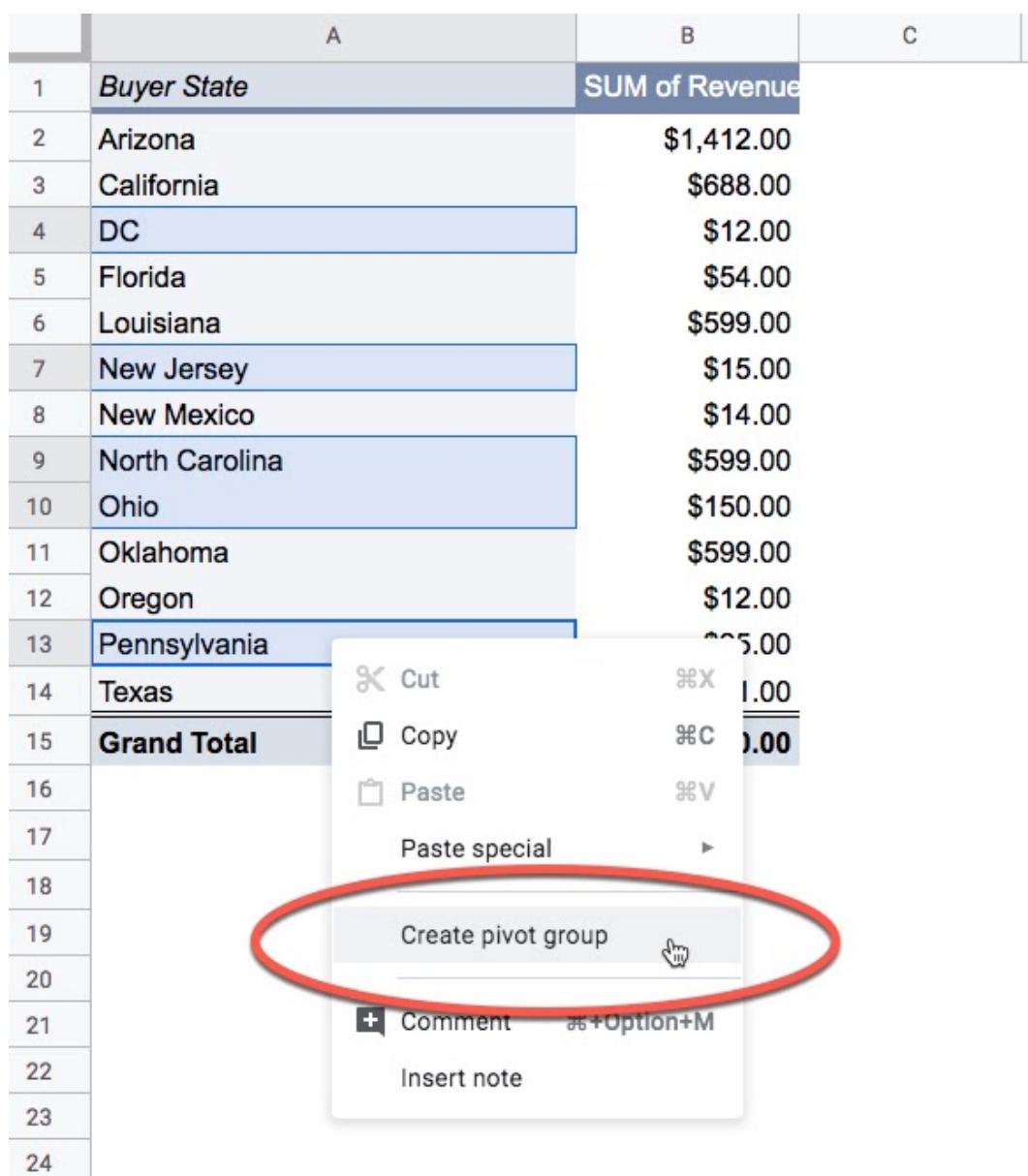
Starting with the following simple pivot table:

	A	B	C
1	Buyer State	SUM of Revenue	
2	Arizona	\$1,412.00	
3	California	\$688.00	
4	DC	\$12.00	
5	Florida	\$54.00	
6	Louisiana	\$599.00	
7	New Jersey	\$15.00	
8	New Mexico	\$14.00	
9	North Carolina	\$599.00	
10	Ohio	\$150.00	
11	Oklahoma	\$599.00	
12	Oregon	\$12.00	
13	Pennsylvania	\$25.00	
14	Texas	\$1,481.00	
15	Grand Total	\$5,660.00	
16			
17			
18			

Imagine we want to group the state results for each different Sales Rep.

Step 1: select the items you want to group by clicking them and holding down the Ctrl key (PC/Chromebook) or Cmd key (Mac)

Step 2: right click and select "Create pivot group"



	A	B	C
1	Buyer State	SUM of Revenue	
2	Arizona	\$1,412.00	
3	California	\$688.00	
4	DC	\$12.00	
5	Florida	\$54.00	
6	Louisiana	\$599.00	
7	New Jersey	\$15.00	
8	New Mexico	\$14.00	
9	North Carolina	\$599.00	
10	Ohio	\$150.00	
11	Oklahoma	\$599.00	
12	Oregon	\$12.00	
13	Pennsylvania	\$5.00	
14	Texas	1.00	
15	Grand Total	0.00	
16			
17			
18			
19			
20			
21			
22			
23			
24			

This will create a custom group in your pivot table labeled with the group items.

Step 3: To rename the group, simply type over the top of the default label.

	A	B	C	D
1	Grouped Buyer State	Buyer State	SUM of Revenue	
2	- Arizona	Arizona	\$1,412.00	
3	Arizona Total		\$1,412.00	
4	- California	California	\$688.00	
5	California Total		\$688.00	
6	- DC, New Jersey, North Carolina, ...	DC	\$12.00	
7		New Jersey	\$15.00	
8		North Carolina	\$599.00	
9		Ohio	\$150.00	
10		Pennsylvania	\$25.00	
11	DC, New Jersey, North Carolina, ... Total		\$801.00	
12	- Florida	Florida	\$54.00	
13	Florida Total		\$54.00	
14	- Louisiana	Louisiana	\$599.00	
15	Louisiana Total		\$599.00	
16	- New Mexico	New Mexico	\$14.00	
17	New Mexico Total		\$14.00	
18	- Oklahoma	Oklahoma	\$599.00	
19	Oklahoma Total		\$599.00	
20	- Oregon	Oregon	\$12.00	
21	Oregon Total		\$12.00	
22	- Texas	Texas	\$1,481.00	
23	Texas Total		\$1,481.00	
24	Grand Total		\$5,660.00	
25				
26				
27				

To create more groups, repeat the steps above.

When our grouping is complete, our new pivot table with custom grouping might look something like this:

	A	B	C	D
1	Grouped Buyer State	Buyer State	SUM of Revenue	
2	- Sales Rep: Jon Smith	DC	\$12.00	
3		New Jersey	\$15.00	
4		North Carolina	\$599.00	
5		Ohio	\$150.00	
6		Pennsylvania	\$25.00	
7	Sales Rep: Jon Smith Total		\$801.00	
8	- Sales Rep: Karen Bradley	Arizona	\$1,412.00	
9		Florida	\$54.00	
10		Louisiana	\$599.00	
11		New Mexico	\$14.00	
12		Oklahoma	\$599.00	
13		Texas	\$1,481.00	
14	Sales Rep: Karen Bradley Total		\$4,159.00	
15	- Sales Rep: Noah Williams	California	\$688.00	
16		Oregon	\$12.00	
17	Sales Rep: Noah Williams Total		\$700.00	
18	Grand Total		\$5,660.00	
19				
20				
21				
22				

It's as easy as that!

(If you plan to use these groups frequently, you might want to create a column in your dataset with each row categorized into its group. That way you can use them directly in your pivot tables.)

#97. Pivot Table Bins

Following on from tip 95 (date grouping in pivot tables) and tip 96 (custom text grouping in pivot tables), this tip looks at how to group values in pivot tables into groups, or bins. Think histograms.

Suppose you have a column containing ages of individuals in your dataset, and you want to know how many fall into the 20 - 29 age bracket, the 30 - 39 bracket, the 40 - 49 bracket etc.

You can easily do this with value grouping in your pivot table.

Step 1: Add the ages into the rows section of your pivot table.

Step 2: Add a COUNT metric into the values section, in this example a count of the ages column.

Step 3: Right click on one of the ages in the pivot table and select "Create pivot group rule"

Step 4: Choose appropriate bounds and interval size. Use the minimum and maximum values of your dataset to help you determine these values. Experiment with the interval sizes to see what works for you.

	A	B	C	D	E
1	Ages	COUNT of Ages			
2		20	1		
3		21	2		
4		22	3		
5		23			
6		25			
7		26			
8		27			
9		28			
10		29			
11		30			
12		31			
13		32			
14		33			
15		34			
16		35			
17		38			
18		39			
19		41			
20		44	1		
21		45	1		
22		46	1		

Grouping rule X

Minimum value

Maximum value

Interval size

CANCEL OK

The finished pivot table will look something like this:

	A	B	
1	Grouped Ages	COUNT of Ages	
2	20 - 29	14	
3	30 - 39	9	
4	40 - 49	15	
5	50 - 59	12	
6	60 - 69	13	
7	70 - 79	18	
8	80 - 89	18	
9	90 - 100	1	
10	Grand Total	100	
11			

What it's telling us is that there are 14 records in our dataset that had an age value between 20 and 29 inclusive. Then there are 9 records between 30 and 39, etc.

It's a really useful feature for data analysis. You can create charts from these pivot tables, which will help you ascertain things like whether the distribution of your data is normal.

(The data used in the example above was generated by the RANDBETWEEN function so is definitely not normally distributed!)

There's one nuance to keep in mind. Toggling the "Show totals" checkbox for the rows of dates will undo the grouping you've set. You can add it right back, but it's easier to remove the totals before grouping to avoid this behavior.

#98. Use the Combo chart to create bullet charts

Bullet charts are variations of bar charts that include a primary measure against a range of other measures for comparative purposes.

For example, you might show your revenue against low and high forecast values to gauge performance.

There is no native bullet chart option in Google Sheets, so we need to use the Combo chart option to create one. And here's how we do that.

Bullet Chart Dataset

	A	B	C	D	E	F	
1	Bullet Chart Data						
2							
3		Available	Low	Low Label	High	High Label	
4	Available	35,000	32,000	Low	38,000	High	
5							
6							

Column B has the primary measure (called "Available"). Columns C and E have the low and high range values. Columns D and F are optional labels.

Bullet Chart Setup

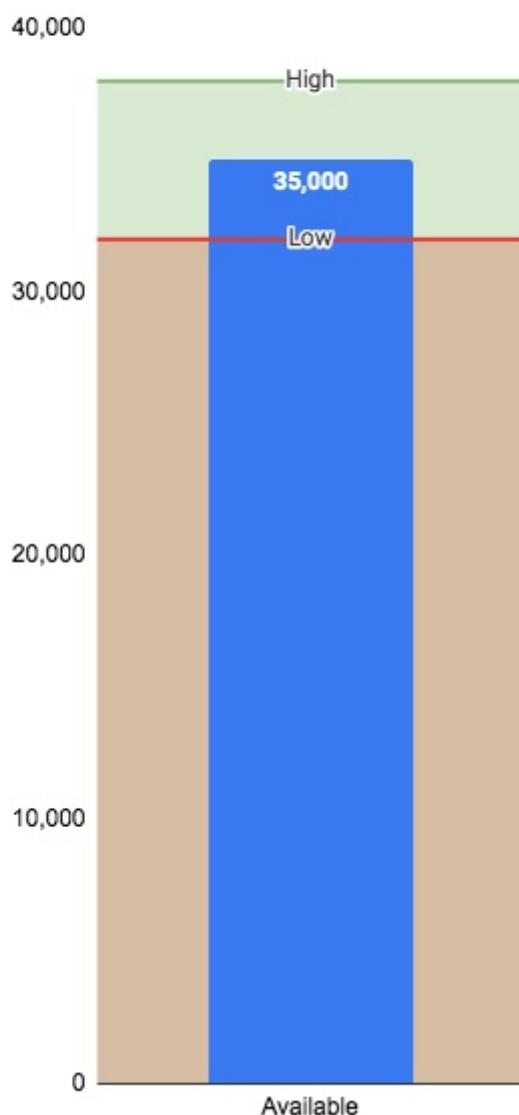
1. Highlight the whole dataset
2. Insert > Chart
3. Under Setup option, choose Chart type to be Combo chart
4. Under Customize, set the Series to the following values:
5. Available should be a Column
6. Low should be Stepped area
7. High should be Stepped area

8. For the Low and High series, add Data labels and set Type to Custom and the Position to Center
9. Feel free to remove the Legend (Position > None)
10. Experiment with the colors of the series.

Your final chart should look something like this:

H	I	J	K	L
---	---	---	---	---

Bullet Chart Example



#99. Use Array Formulas to check whether cells are blank

Add some data to column A. It could be a list of numbers, or a list of names, or dates, or whatever you want really. Doesn't matter if you include blank cells or not either.

Perhaps it looks something like this:

	A	
1	1	
2	2	
3	3	
4		
5	5	
6	6	
7		
8	text value	
9	9	
10	4/24/2020	
11		
12		

In cell C1, add the following formula:

```
=ArrayFormula(A:A<>"")
```

This checks every row in column A to see whether or not it's blank. The output is a vector of TRUE/FALSE values in column C, with TRUE showing for any rows that are non-blank in column A.

Add a ROW(A:A) multiplier to the formula:

```
=ArrayFormula(ROW(A:A)*(A:A<>""))
```

This takes the TRUE/FALSE values, converts them into 1s and 0s and multiplies against the row number.

So you're left with some ROW values and lots of zeros.

Now the data looks like this:

	<i>fx</i>	=ArrayFormula(ROW(A:A)*(A:A<>""))	
	A	B	C
1	1		1
2	2		2
3	3		3
4			0
5	5		5
6	6		6
7			0
8	text value		8
9	9		9
10	4/30/2020		10
11			0
12			0
13			0
14			0
15			0

All that remains is to take the max of this set:

```
=ArrayFormula(MAX(ROW(A:A)*(A:A<>"")))
```

which gives the answer 10 in this example.

MATCH Alternative

Another alternative to find the last non-blank row is to use the MATCH function, like so:

```
=ArrayFormula(MATCH(2,1/(A:A<>"") ,1))
```

FILTER Alternative

Yet another alternative to find the last non-blank row is to use the FILTER function, like so:

```
=MAX(FILTER(ROW(A:A), A:A<>""))
```

Which has the nice property of not needing to be an Array Formula.

LOOKUP to retrieve value

And to retrieve the last value itself, not the row number, you can use the LOOKUP function:

```
=ArrayFormula(LOOKUP(2,1/(A:A<>"") ,A:A))
```

The key with both of these formulas is that the search value "2" is bigger than any value in the search range, which consists of 1's or errors. Try building up the formula in pieces to see how it works, as we did in the first example.

#100. Use the Hyperlink function to create a link to the next blank row

We'll use the formula from the previous tip as part of a hyperlink to jump us to the next blank row. It could be useful if you have big data tables that you use for data entry.

Step 1

Right click on cell A1 in a Sheet, select the "Get link to this cell" option and then paste the url into a blank cell somewhere in your Sheet. You'll notice it takes the form:

`https://docs.google.com/spreadsheets/d/XXXXXXXXXXXXXX/edit#gid=0&range=A1`

where the "XXXXXXXX" is your unique spreadsheet ID.

Step 2

In a blank cell, create a hyperlink with this link to A1, as follows:

```
=HYPERLINK("https://docs.google.com/spreadsheets/d/XXXXXXXXXXXXXX/edit#gid=0&range=A1", "Add Data")
```

remembering to swap out the "XXXXXXXX" for your unique spreadsheet ID.

Step 3

Now, use the formula from last week to change the range reference from A1 to the next blank row.

Remove the "1" at the end of the range reference (the "...range=A1" part) and replace with an ampersand and the formula to find the last row number (the green section in the following formula):

```
=HYPERLINK("https://docs.google.com/spreadsheets/d/XXXXXXXXXXXXXX/edit#gid=0&range=A"&ArrayFormula(MAX(ROW(A:A)*(B:A<>""))), "Add Data")
```

Clicking this hyperlink will take you to the last non-blank entry in column A.

Step 4

To get the hyperlink to take you to the next blank row, just add 1 to the Array Formula:

```
=HYPERLINK("https://docs.google.com/spreadsheets/d/XXXXXXXXXXXXXX/edit#gid=0&range=A"&ArrayFormula(MAX(ROW(A:A)*(A:A<>""))+1), "Add Data")
```

Here's this formula in action on a larger dataset:

	Add Data	Property Type	Address	Projected	Closed	Paid
1						
2	1	Apartment	1653 First Avenue	11/5/2019	11/10/2019	Yes
3	2	House	476 Oak Close, A	11/16/2019	11/16/2019	Yes
4	3	Apartment	1199 Washington	12/3/2019	12/8/2019	Yes
5	4	Apartment	1714 Oak Street,	12/28/2019	12/29/2019	Yes
6	5	Apartment	1665 Cedar Clos	1/1/2020	1/5/2020	Yes
7	6	Condo	942 Seventh Roa	1/1/2020	1/1/2020	Yes
8	7	Apartment	656 Fifth Road, A	1/22/2020	1/24/2020	Yes
9	8	Townhouse	1979 Maple Roa	1/23/2020	1/27/2020	Yes
10	9	Condo	1200 Washington	1/24/2020	1/28/2020	Yes
11	10	Townhouse	725 Fourth Aven	1/31/2020	2/3/2020	Yes
12	11	Apartment	625 Elm Close, A	2/12/2020	2/14/2020	Yes
13	12	Condo	112 Ninth Close,	2/20/2020	2/21/2020	Yes
14	13	Apartment	1149 Seventh St	2/22/2020	2/27/2020	Yes
15	14	Townhouse	1394 Sixth Road	2/26/2020	2/27/2020	Yes
16	15	Apartment	1876 Sixth Road	3/1/2020	3/1/2020	Yes
17	16	Studio	307 Cedar Close	3/25/2020	3/27/2020	Yes
18	17	Condo	391 Oak Avenue	3/29/2020	4/2/2020	Yes
19	18	Condo	1581 View Close	3/29/2020	4/2/2020	Yes
20	19	Studio	755 Pine Avenue	4/1/2020	4/6/2020	Yes
21	20	House	1960 Third Road	4/1/2020	4/4/2020	Yes
22	21	Townhouse	55 View Close, A	4/24/2020	4/26/2020	Yes
23	22	Townhouse	193 Fifth Road, A	5/1/2020	5/1/2020	Yes
24	23	Condo	1176 First Street,	5/1/2020	5/3/2020	Yes
25	24	House	362 Park Close,	5/1/2020	5/6/2020	Yes

Closing Comments

Thanks for downloading and reading this ebook!

It was great fun to put together so I hope you've enjoyed it and found it helpful.

Corrections, comments or suggestions are encouraged. Let me know what you think.

Got any weird and wacky spreadsheet tips or tricks? I'm working on the next 100 tips and will credit anyone who contributes.

Remember, keep summing, sorting, filtering, pivoting, charting and analyzing, but above all keep having fun!

Thanks!

Ben Collins

May 2020

Contact Details

Sign up for the Google Sheets Tips email here:

<https://www.benlcollins.com/google-sheets-tips/>

Contact me here:

<https://www.benlcollins.com/contact/>

Full archive of Google Sheet tutorials here:

<https://www.benlcollins.com/>

The Collins School of Data (online training courses for Google Sheets):

<https://courses.benlcollins.com/courses>

Twitter:

<https://twitter.com/benlcollins>