Excel Yourself 2019

By Neale Blackwood

www.a4accounting.com.au

a4@iinet.net.au

A4 ACCOUNTING

EXCEL CONSULTING, TRAINING AND WEBINARS

Contents

Introduction 3
February 2019 – The new IFS Function 4
May 2019 – Allocations using the MOD function 6
June 2019 – Complex allocations 9
September 2019 - Conditional Format with helper cells 11

Introduction

This session will cover some of my Excel Yourself articles that were first published in the INTHEBLACK magazine during 2019. We will examine four of the 11 articles in the hour allotted. This is my eighth webinar covering my INTHEBLACK articles. The other seven webinar recordings and materials are available for free on my website.

The companion Excel sheets are arranged in date order starting with the February article on the left and working across to the right to finish with the September article. The sheet tabs are grouped and coloured. There is an INDEX sheet to assist in navigation.

The content of this manual has been re-written and adapted from the original articles and includes some extra content.

This session will focus mainly on conditional calculations in Excel. We'll look at a brand-new function to handle multiple conditions. We'll also look at how you can use the IF function as well as the logic calculations. There is also a focus on budgeting structures. We will also look at Conditional Formatting.

INTHEBLACK Website

Note: I record a companion video for each month's article. You can see them all at this page

https://www.intheblack.com/topics/excel

Most of the articles on the website also include a companion file.

I have recently started updating my older articles that weren't on the website. My articles started in 2002. The series is called the Excel Vault and I review multiple articles and provide a video and updates on techniques.

CPA Australia Podcasts

I also have a page on my website for my Excel podcasts with CPA Australia.

https://a4accounting.com.au/podcasts/

February 2019 – The new IFS Function

You will need the latest version of Excel to work with this function.

The IF function has three arguments

1. Logical Test
2. TRUE action
3. FALSE action

IF(Logical test, TRUE action, FALSE action)

When you have a more complex problem you can use IF functions within IF functions, which is called nesting the IF functions. This can create a complex formula that is hard to understand.

Nested IF function structure

=IF(Logical test,IF(Logical test, TRUE action, FALSE action),IF(Logical test, TRUE action, FALSE action))

Brackets (parentheses) are use around each IF function. Each IF function is treated as a level within the formula. The more levels, the harder it can be to understand the formula. Excel uses different coloured brackets at each level to help identify the levels. The more brackets you have in a formula the harder it is to follow. To reduce the number of brackets the IFS function was created.

The new IFS function has two arguments

1. Logical Test
2. TRUE Action

IFS(Logical test#1, TRUE action#1, Logical test#2, TRUE action#2,…)

With the IFS function you only handle the TRUE action of each Logical Test. There is no need to nest the functions. You can handle multiple Logical Tests in the same function, no extra levels. You pair up each Logical Test with its TRUE Action. You can perform other calculations in the TRUE action argument as you would an IF function.

This reduces the number of brackets and makes the formula easier to read. There is a trick to using it though and I will explain with an example. You may have figured out the issue if you are a frequent user of the IF function.

What if all the logical tests return FALSE?

This is the problem and luckily there is an easy solution.

Worked Example

There is an easier way to handle this solution, but this example is for demonstration purposes.

The image on the following page has three columns of figures, one for each state. Based on the entry in cell B2 we want to add up the correct state's values. Cells B4, B6, B8 and B10 all have different formulas to solve the same problem.

The formula in cell B4 uses the IF function and is the standard approach to the problem.

The formula in cell B8 also uses an IF function but uses a slightly non-standard approach.

The formula in cell B4 is shown below.

	A	B	C	D	E	F
1				WA	NSW	VIC
2	Region	NSW		2,689	1,037	2,570
3				1,840	2,642	1,231
4	Total IF	19,281		2,273	2,225	1,483
5				1,865	1,817	1,566
6	Total IFS	19,281		2,315	2,842	1,721
7				2,609	2,219	2,964
8	Total IF	19,281		1,985	1,830	2,814
9				1,154	2,583	1,336
10	Total IFS	19,281		1,892	2,086	1,338

=IF(B2=D1,SUM(D2:D10),IF(B2=E1,SUM(E2:E10),IF(B2=F1,SUM(F2:F10),0)))

This is a nested IF function using three IF functions. Each IF function tests for a match of the State in cell B2 with the header row for each state in columns D, E and F. If a match is found then a SUM is performed. The nested IF functions are in the FALSE action argument of the preceding IF function.

The image below is in colour. It shows the colour coding used to match the brackets at each level of the formula.

The 0 on the end is the FALSE action argument of the last IF function. It represents not finding a match. That may happen if the cell is blank or there is an invalid entry in the cell.

The IFS formula in cell B6 is

=IFS(B2=D1,SUM(D2:D10),B2=E1,SUM(E2:E10),B2=F1,SUM(F2:F10),TRUE,0)

This has the same logical tests and TRUE actions as the IF solution. As was mentioned earlier it needs to handle the case when no match is found. That is what the TRUE,0 on the end achieves. This is a hard-keyed TRUE result that ensures that something will be returned if all the other logical test return FALSE. If you omit The TRUE,0 then #N/A is returned if no TRUE result is returned. The IFS function processes the logical tests from left to right. The order you enter them may be important. It depends on the logic you are using.

Alternative solutions

Both the IF and IFS can return ranges. This is not widely known. Two alternative solutions are shown below.

Cell B8

=SUM(IF(B2=D1,D2:D10,IF(B2=E1,E2:E10,IF(B2=F1,F2:F10,0))))

Cell B10

=SUM(IFS(B2=D1,D2:D10,B2=E1,E2:E10,B2=F1,F2:F10,TRUE,0))

Both these have a single SUM function. Both are shorter than their original formula. The IF and IFS functions are returning a range (or the value 0) for the SUM to use.

May 2019 – Allocations using the MOD function

The MOD function is not widely used. This is a pity as it offers an easy way to identify regular patterns. In this month's article I focused on how you can allocate across months using a reasonably simple formula to identify the correct months to allocate.

We will also examine how you can change the timing of the allocations. There are four separate sheets used to demonstrate these techniques.

The allocations performed are monthly, bi-monthly, quarterly and six-monthly.

In the next article (June 2019) wee examine creating a formula to identify specific months for allocations.

In the image below columns A to D are used for inputs. Column E is a helper column and calculates the amount to allocate.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1						1	2	3	4	5	6	7	8	9	10	11	12		
2	Dept	Account	Annual $	Number of allocations	Allocation Amount	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Total	Valid Allocation
3	ADMIN	1234	1,000	6	167	0	167	0	167	0	167	0	167	0	167	0	167	1,000	TRUE
4	ADMIN	1236	3,000	4	750	0	0	750	0	0	750	0	0	750	0	0	750	3,000	TRUE
5	ADMIN	1238	5,000	3	1,667	0	0	0	1,667	0	0	0	1,667	0	0	0	1,667	5,000	TRUE
6	ADMIN	1240	7,000	1	7,000	0	0	0	0	0	0	0	0	0	0	0	7,000	7,000	TRUE

Row 1 contains the numbers from 1 to 12 to simplify the final formula. In general, when I'm working with our financial year I use a row that contains the month numbers. This can simplify many calculations.

In column D you can enter 12, 6, 4, 3, 2, 1 as entries. The numbers entered need to be evenly divisible into 12.

The formula in column E simply divides the annual amount by the number of allocations.

There is a formula in cell F3 that calculates the amount to allocate full each month. That formula is copied down and across to populate all of the month columns. The formula in cell F3 is

=IF(MOD(F$1,12/$D3)=0,$E3,0)

The MOD function performs an unusual calculation. It divides one number by another number and returns the remainder. This sounds like a useless calculation but when used correctly it can be used to identify patterns. In our case, when an allocation should be made.

We need to identify which month to allocate the amount in column E. In row 3, because we need to make six allocations, we need to make an allocation every two months. By dividing 12 by the number of allocations 12/$D3 we identify the number of months between allocations. That is why the number entered in column D needs to be evenly divisible into 12.

The MOD function divides the month number in row 1 by the number of months between allocations and returns the remainder. When the remainder equals zero then it is an allocation

month. In the IF function above if the MOD function equals zero then the allocation amount from column E is returned. Otherwise zero is returned.

Using this technique allocates the amount at the end of the period. If you wanted to allocate the amount at the beginning of the period you would use a slightly different formula. The formula for cell F3 which allocates at the beginning of the period is

=IF(MOD(F$1,12/$D3)-1=0,$E3,0)

The problem with this formula is that there will be an allocation in July for all the different allocation timings – see image below.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1						1	2	3	4	5	6	7	8	9	10	11	12		
2	Dept	Account	Annual $	Number of allocations	Allocation Amount	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Total	Valid Allocation
3	ADMIN	1234	1,000	6	167	167	0	167	0	167	0	167	0	167	0	167	0	1,000	TRUE
4	ADMIN	1236	3,000	4	750	750	0	0	750	0	0	750	0	0	750	0	0	3,000	TRUE
5	ADMIN	1238	5,000	3	1,667	1,667	0	0	0	1,667	0	0	0	1,667	0	0	0	5,000	TRUE
6	ADMIN	1240	7,000	1	7,000	7,000	0	0	0	0	0	0	0	0	0	0	0	7,000	TRUE

Logic Calculation

Even though the original formula was quite short it can be made even shorter by using logic calculations.

Logic calculations involve TRUE and FALSE. In Excel TRUE equals one and FALSE equals zero. We can use those values in our calculations if we need to zero an amount. Obviously if you multiply any value by zero you zero it. We can use that fact in our calculations.

The alternative formula for cell F3 using logic calculations is

=(MOD(F$1,12/$D3)=0)*$E3

When using logic calculations you need to enclose the logical test between brackets (parentheses). This captures the result and when you multiply by the result you convert the TRUE or FALSE into a one or zero. When you multiply by the result you will leave any values unchanged if the result was TRUE. And you will zero any amounts if the result was FALSE.

Logic calculations are useful when you are creating a formula that either displays a value or zeroes a value based on a condition.

Start or End

As I mentioned above, if you want to allocate the amounts at the start of the period it ends up loading July with all of the start amounts. You may want the ability to decide whether to allocate at the start or the end of the period on each row. The image below shows the adjusted structure that allows that.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1							1	2	3	4	5	6	7	8	9	10	11	12		
2	Dept	Account	Annual $	Number of allocations	Start/End	Allocation Amount	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Total	Valid Allocation
3	ADMIN	1234	1,000	6	Start	167	167	0	167	0	167	0	167	0	167	0	167	0	1,000	TRUE
4	ADMIN	1235	2,000	6	End	333	0	333	0	333	0	333	0	333	0	333	0	333	2,000	TRUE
5	ADMIN	1236	3,000	4	Start	750	750	0	0	750	0	0	750	0	0	750	0	0	3,000	TRUE
6	ADMIN	1237	4,000	4	End	1,000	0	0	1,000	0	0	1,000	0	0	1,000	0	0	1,000	4,000	TRUE
7	ADMIN	1238	5,000	3	Start	1,667	1,667	0	0	0	1,667	0	0	0	1,667	0	0	0	5,000	TRUE
8	ADMIN	1239	6,000	3	End	2,000	0	0	0	2,000	0	0	0	2,000	0	0	0	2,000	6,000	TRUE
9	ADMIN	1240	7,000	2	Start	3,500	3,500	0	0	0	0	0	3,500	0	0	0	0	0	7,000	TRUE
10	ADMIN	1241	8,000	2	End	4,000	0	0	0	0	0	4,000	0	0	0	0	0	4,000	8,000	TRUE

The formula in cell F3 in the above image is

```
=IF(MOD(G$1,12/$D3)-IF($E3="Start",1,0)=0,$F3,0)
```

The modification to the formula inserts an extra IF function to determine whether to deduct one or zero from the result of the MOD function. By deducting one from the result of the MOD function we allocate at the start of the period.

This formula can also be converted into a logical calculation which creates a shorter formula. The alternative formula the cell F3 is

```
=(MOD(G$1,12/$D3)-($E3="Start")=0)*$F3
```

In this formula the result of the logical test is being deducted from the result of the MOD function. This works because we need to deduct one from the MOD function to adjust it to allocate at the start of the period.

Validation

In all of these examples the final column is used to validate the amount being allocated. It displays TRUE if the row is valid and FALSE if it isn't. This can be used with a conditional format to identify the row as invalid.

June 2019 – Complex allocations

Last month's allocation article looked at how you can allocate regular amounts over the year. A common request is the ability to allocate to specific months over the year.

In the image below the structure allows us to enter amounts either by month or annually. We also have the ability to specify the individual months for the allocation. This has been slightly amended from the original article as I have deleted a column.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Department	Account	Month $	Annual $	Allocate Months	Number of Months	Valid Input	Allocation Amount	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Total	Valid Allocation
2	ADMIN	1234	100		Jul,Sep,Jan,Feb,Jun	5	TRUE	100	100	0	100	0	0	0	100	100	0	0	0	100	500	TRUE
3	ADMIN	1235		1,000	Sep,Dec,Mar,Jun	4	TRUE	250	0	0	250	0	0	250	0	0	250	0	0	250	1,000	TRUE
4	ADMIN	1236	200		Nov,Mar,Apr	3	TRUE	200	0	0	0	0	200	0	0	0	200	200	0	0	600	TRUE
5	ADMIN	1237		2,000	Aug,Dec	2	TRUE	1,000	0	1,000	0	0	0	1,000	0	0	0	0	0	0	2,000	TRUE
6	ADMIN	1238	300		Jul,Sep	2	TRUE	300	300	0	300	0	0	0	0	0	0	0	0	0	600	TRUE
7	ADMIN	1239		3,000	Jul,Sep	2	TRUE	1,500	1,500	0	1,500	0	0	0	0	0	0	0	0	0	3,000	TRUE
8	ADMIN	1240		4,000	Sep	1	TRUE	4,000	0	0	4,000	0	0	0	0	0	0	0	0	0	4,000	TRUE
9	ADMIN	1241	400			12	TRUE	400	400	400	400	400	400	400	400	400	400	400	400	400	4,800	TRUE
10	ADMIN	1242		4,000		12	TRUE	333	333	333	333	333	333	333	333	333	333	333	333	333	4,000	TRUE

This example has a number of helper cells to simplify the final formula. Columns F, G and H are all used to break down the calculation into smaller steps.

There should only be one entry per row between columns C and D. A validation will confirm this.

In column E you enter the three-character month names separated by commas. A single month name does not require a comma. If no months are entered in column E then it is assumed that the allocation is done evenly across all the months.

Column F - Number of Months

The first thing we need to do is to figure out how many allocations are to be made. We need to figure out how many months are listed in column E. One way to do that is to count how many commas there are in the cell and add 1 to it. Unfortunately Excel does not have a function that counts specific characters in a cell. There is a formula that counts specific characters in a cell.

The formula in cell F2 begins with this formula

```
=LEN(E2)-LEN(SUBSTITUTE(E2,",",""))
```

The LEN function returns the number of characters in a cell. The SUBSTITUTE function replaces one text string with another text string in a cell. By substituting a blank for a comma you remove the commas from the cell. In Excel two double quotation marks together represents a blank. By subtracting the number of characters without the commas from the count of characters with the commas you calculate the number of commas in the cell.

The number of commas in the cell does not determine how many months there are though. We need to add one to the result to calculate how many months are entered. Also if the cell is blank we need to allocate every month so we need to adjust the result of the above formula to handle blanks as well.

The final formula in cell F2 is

```
=LEN(E2)-LEN(SUBSTITUTE(E2,",",""))+IF(E2="",12,1)
```

The IF function that has been added at the end will add 12 to the results of the initial calculation if the cell is blank. It will add one if the cell isn't blank. Remember if the cell in column E is blank we need to return the number 12.

Column G - Valid Input

This column is used to ensure that there is only one entry in the row in between: C and D. The formula is simple and it returns TRUE if there is only one entry and FALSE if there isn't.

```
=COUNTA(C2:D2)=1
```

If the result is FALSE then no allocation will be made since the result is used in column H to zero the allocation amount if column G is FALSE.

Column H – Allocation Amount

The formula in this column relies on the rule that on each row there can only be one entry in columns C and D. The formula adds the month amount to the annual amount divided by the number of months. There can only be one amount hence this formula works. The formula in cell H2 is

```
=(C2+(D2/F2))*G2
```

Multiplying the result from column G zeroes the value if two amount entries have been made.

Columns I to T – Monthly Allocations

The formula in cell I2, which will be copied across and down, is not as complex as you might think. We can use one of Excel's functions to determine if a particular month appears in column E. If it does then we can allocate the amount from column H. The formula in cell I2 is

```
=IFERROR((SEARCH(I$1,$E2)>0)*$H2,($F2=12)*$H2)
```

The SEARCH function returns the character position of one string (text) within another string. Unfortunately the SEARCH function returns an error if it can't find a string. By using the IFERROR function we can identify when the three-character month name is not in column E. If the month name is not in column E then we either return a zero (no allocation) or we return the allocation amount if column F contains 12 (allocate every month).

The above formula is another example of logic calculations which we looked at in the previous month's article. If you are not comfortable with logic calculation you can convert the above formula using IF functions. The above formula using IF functions would be

```
=IFERROR(IF(SEARCH(I$1,$E2)>0,$H2),IF($F2=12,$H2,0))
```

Column V contains a Validation formula that returns TRUE if the monthly allocation is valid.

```
=((C2*F2)+D2)=U2
```

To allocate differing amounts per month you would use more than one row for a Dept/Account combination.

September 2019 - Conditional Format with helper cells

Excel's data validation system has a problem. The problem is you can use paste values to enter a value into a data validation cell. This means you can enter an invalid entry even though a data validation is in place. This will be demonstrated in the webinar.

Excel has a system that helps you identify invalid entries but unfortunately the system requires you to physically view all of the data validation cells to see if there are invalid entries.

If it is important to make sure that you don't have any invalid entries you need to add some extra validation cells to your spreadsheet to confirm entries are valid.

The validation cells can be used to apply conditional formats to make it obvious to users that something is wrong.

The example we will use is a little bit of overkill but it demonstrates how you can add validations to your spreadsheets to ensure valid entries.

In the example below we have three input columns H, I and J. The columns on the left are for validation. Column G displays a tick (valid); a cross (invalid) or nothing (no entry made).

	A	B	C	D	E	F	G	H	I	J
1	All Blanks	Dept	Account	Three Entries	ISNUMBER	Valid		Department	Account	Amount
2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	✓	DISTRIBUTION	12346	6000
3	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	✓	ADMINISTRATION	12350	2000
4	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	✗	ADMINISTRATION		
5	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	✗		12348	
6	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	✗			9000
7	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	✗	PRODUCTION		10000
8	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	✗		12346	30000
9	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	✗	MARKETING	12350	
10	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE				

Grouping (icons above the columns letters) has been added to the validation columns so they can be easily hidden and unhidden.

Columns H and I both contain drop-down lists. The rules we need to apply are

1. On each row all three input cells need an entry for a valid row.
2. The entry in columns H and I must be from the drop-down list.
3. The entry in column J must be a value.

The columns on the left-hand side perform validation checks to ensure that these rules are met.

We will examine all the formulas in row 2 of columns A to G. Note all the formulas have been copied down. These formulas are all reasonably straightforward and, with the exception of column G, all return TRUE or FALSE.

There is a Tables sheet that holds the two lists for the Department and the Account drop down lists.

Column A – All Blanks

The formula in this column is reasonably simple. It confirms that there are no entries in columns H, I and J. The formula in cell A2 is

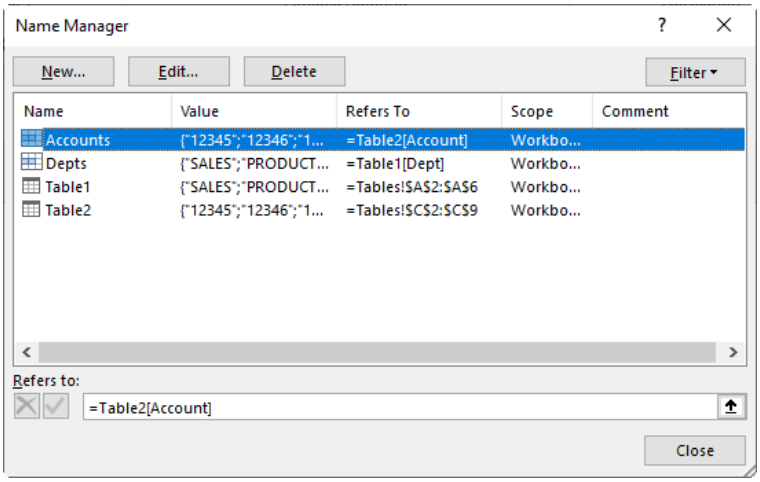
=COUNTA(H2:J2)=0

This displays TRUE if there are three blank cells in the input area. Otherwise it displays FALSE. This column will be used as an override to display a blank cell in column G.

Column B – Dept

A range name has been used to hold the list for the Department drop-down.

This will be shown during the webinar. The image on the right shows the range names, as well as the two formatted table names that have been used in the file.



Formatted tables are a powerful feature and you can check out a free webinar on the topic on my website.

The formula in cell B2 is

=COUNTIF(Depts,H2)>0

This counts the number of times the entry in cell H2 is in the Depts range name. It compares the result to 0. If the entry is valid TRUE will be displayed. If the entry is invalid FALSE will be displayed. FALSE will also be displayed if the entry is blank.

Column C - Account

The same technique used for column B is also used in C, but with the Accounts range name. The formula in cell C2 is.

=COUNTIF(Accounts,I2)>0

Column D – Three Entries

A valid entry will contain three entries. The formula in cell D2 confirms there are three entries. The formula in cell D2 is.

=COUNTA(H2:J2)=3

This displays TRUE if there are three entries and FALSE if there are not. **Note:** the COUNT function only counts numbers and dates. COUNTA count all entries.

Column E – ISNUMBER

We need to confirm that column J contains a number. Excel has a function that returns TRUE if a cell contains a number. The formula for cell E2 is.

=ISNUMBER(J2)

Column F – Valid

This column has the overall validation check. Based on the results in columns A to E we can determine if the input cells are valid. The formula in cell F2 is

=OR(A2,AND(B2:E2))

This formula will display TRUE if the input cells are valid or blank. The OR function returns TRUE if any of the logical tests are TRUE. The AND function only returns TRUE when every single logical test is TRUE.

Referring to A2 in the OR function means A2 becomes an override. If there is a TRUE in column A then that makes the overall validation TRUE. If there is a FALSE in column A then all of the other validations from column B to E must be TRUE for the validation to return TRUE.

Column G – Ticks, crosses and blanks

We require three different entries in column G. We need a tick for a valid entry. A cross for an invalid entry. And a blank cell if there are no entries.

The simple formula in column G will display one of three entries. One will represent a valid entry. Zero will represent an invalid entry. And two will represent a blank entry. The formula in cell G2 is.

=A2+F2

That's right, by simply adding up columns A and F you will end up with the values required. Remember that TRUE equals one and FALSE equals zero. Two TRUE entries will equal two.

The formula results are shown over the page where you can see the ones, zeroes and twos in column G.

	A	B	C	D	E	F	G	H	I	J
1	All Blanks	Dept	Account	Three Entries	ISNUMBER	Valid		Department	Account	Amount
2	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	1	DISTRIBUTION	12346	6000
3	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	1	ADMINISTRATION	12350	2000
4	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	0	ADMINISTRATION		
5	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	0		12348	
6	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	0			9000
7	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	0	PRODUCTION		10000
8	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	0		12346	30000
9	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	0	MARKETING	12350	
10	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	2			

These numbers make it very easy to create a conditional format to display the ticks and crosses.

Select the range G2:G10 and click the Conditional Formatting drop-down on the Home ribbon and select Icon sets.

Select the tick , exclamation mark and cross as shown on the right.

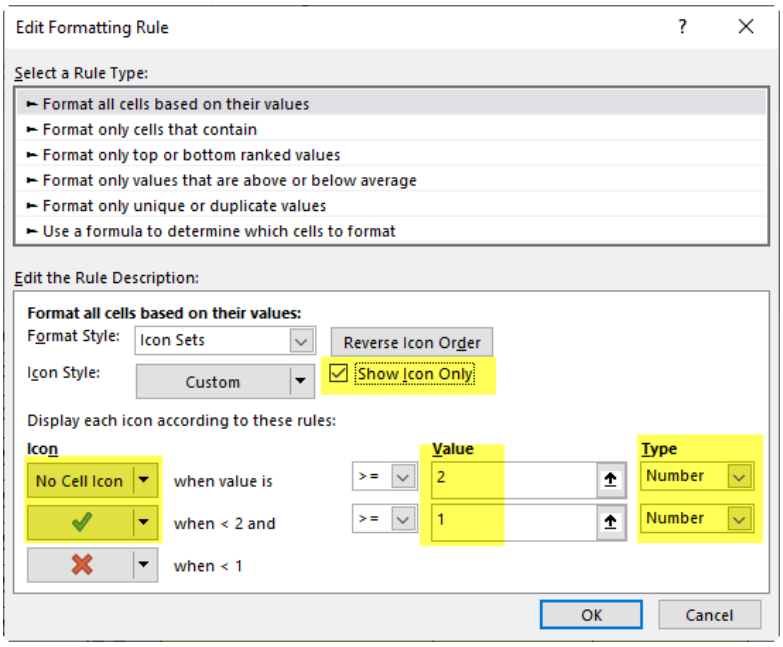
The default settings are not what we need so we must edit the format.

Click the Conditional Formatting drop-down again and choose Manage Rules and double click the Icon Set rule that is listed.

Change the settings as per the image on the right.

The entries highlighted will need to be changed.

Click OK and OK again.



Bonus – getting invalid cells to go red

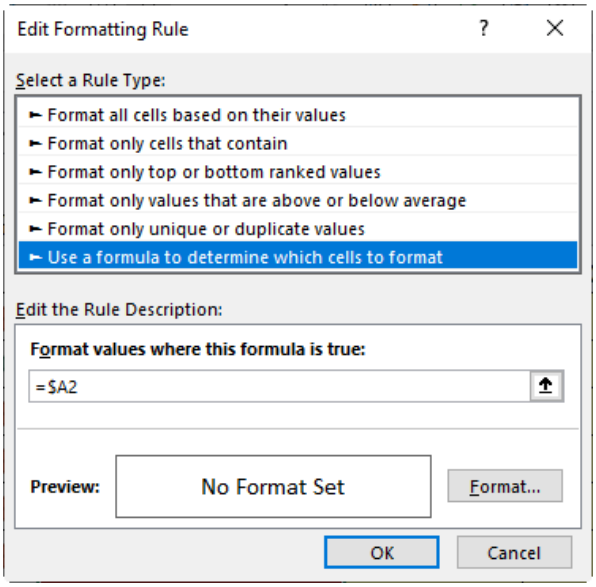
Column G displays icons to show if a row is valid or invalid. It doesn't tell us which cells are valid or invalid on the row. We can use the validations on the left-hand side to change the format of the individual input cells.

We will use the Department input column as an example.

We only want the cell in an invalid row to change colour. So in our example row 10 should not change colour because there are no entries.

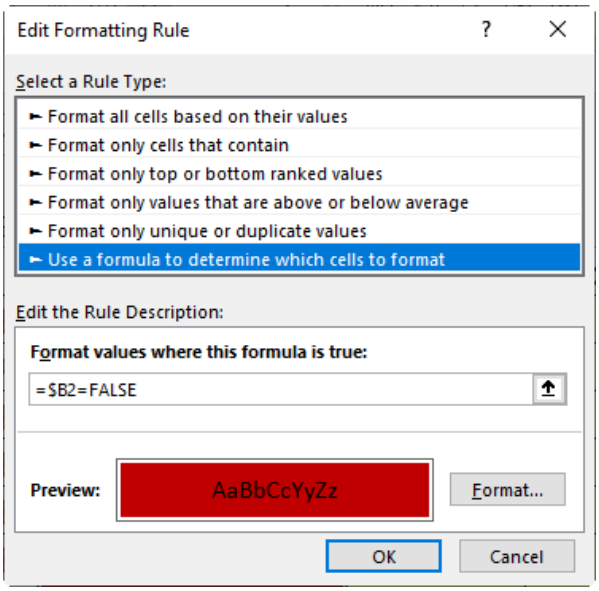
Select the range H2:H2. Click the Conditional Formatting drop-down and select New Rule.

Use the settings in the dialog on the right and click OK.



Click the Conditional Formatting drop-down and select New Rule again.

Use the settings in the dialog on the right.



Finally click the Conditional Formatting drop-down and select Manage Rules.

Make sure the dialog looks like the dialog below and click OK.

Ensure the No Formula Set rule is first and the red format rule is second.

Make sure the Stop If TRUE option is ticked for the first rule.

You can use the arrow icons on the right of the Delete Rule button to move the selected rule up or down.