



Happy New Year!

Welcome to 2018. Not just the New Year – we have another event to celebrate: a second Excel MVP in the ranks. Congratulations to our training director Tim Heng! To celebrate, we decided to have a 10% discount on any training ordered in January too.

We've made our New Year's Resolution to continue with the usual regular features too, such as Power Query Pointers, the A to Z of Excel Functions (you can **CHOOSE** which ones you want to read) and more Keyboard Shortcuts. There's also exciting news about our annual Excel Summit – it's coming back to Australia and New Zealand in April...

Until next month.

Liam Bastick, Managing Director, SumProduct



Time for Tim

SumProduct is pleased to announce that our Sydney director, **Tim Heng**, has been presented with Microsoft's Most Valuable Professional (MVP) award for Excel for 2018-19 (*he's always been a little ahead of himself!*). This award recognises exceptional technical community leaders from around the world who voluntarily share their high quality, real world expertise with others. Microsoft MVPs are a highly select group of experts representing technology's best and brightest who share a deep commitment to community and a willingness to help others.



Worldwide, there are over 100 million participants in technical communities; of these participants, there are fewer than 4,000 active Microsoft MVPs. In Excel, c.100 people have received this award.

Microsoft's MVP Award evaluates technical expertise and voluntary community contributions for the past year, considering the quality, quantity and level of impact of contributions. It's a difficult award to attain and here at SumProduct, we are pleased his efforts have finally been recognised.

Together with Liam Bastick, there are now two MVPs in our ranks; at SumProduct, you can rely on our experience and willingness to help.

SumProduct New York Presentations and a New Modelling World Champion

Every year, the good folks at ModelOff run the World Financial Modelling Championships over three rounds, culminating in the final round being held to great pomp and fanfare at a major city around the world. Having run the finals in London in 2015 and 2016, with Joseph Lau winning the major prize, 2017 saw the return back to New York, and the crowning of a new World Champion, Alvin Woon.

In conjunction with the finals, ModelOff held one of their Global Training Camps in New York. With 150 attendees over the two-day conference,

it was a great opportunity to hear from a range of great speakers from Microsoft, former ModelOff champions and finalists, and experienced professionals including our own Sydney Director, Tim Heng, who flew over to run his first sessions as a freshly minted Microsoft MVP.

His sessions went down well apparently and Tim would like to say a heartfelt "thank you" to Microsoft and ModelOff for having SumProduct there, and congratulations to Alvin and all of the other finalists who made it to New York. Looking forward to the next year!

Unlock Excel 2018



CPA AUSTRALIA PROFESSIONAL DEVELOPMENT
INVEST IN YOURSELF

UNLOCK EXCEL
IMPROVE | DISCOVER | STREAMLINE

Advance your Excel knowledge with Microsoft's awarded MVPs – the world's leading Excel gurus – and learn how to improve your efficiency, discover new skills and streamline your processes.

APRIL 2018
BRISBANE
MELBOURNE
SYDNEY
WELLINGTON

EARLY BIRD PRICES AVAILABLE!
View program and book now
cpaaustralia.com.au/unlockexcel

BE HEARD.
BE RECOGNISED.

CPA AUSTRALIA

You may recall, together with CPA Australia, SumProduct hosted the Excel MVP conference in 2017 for Australia and New Zealand. Well, we're back!! Hot off the press, we thought we'd give our readers the good news.

There will be more emphasis on financial modelling than last year, but the usual topics (charting, dashboards, Power BI / Power Pivot / Power Query, PivotTables and Excel Tricks & Tips) will all be covered too. Speakers are still being finalised at this stage, but MVPs Gašper Kamenšek, Inge Hawighorst, Jon Peltier, Ken Puls, Mynda Treacy, as well as our own Liam Bastick and Tim Heng are confirmed as at the time of writing. The event is aimed at all abilities and we hope you'll find there's something for everybody.

The events are as follows:

- **MELBOURNE** 9-10 April
- **SYDNEY** 12-13 April
- **BRISBANE** 16-17 April
- **WELLINGTON** 19-20 April

For more details, you may view the full program and book (very shortly!) at www.cpaaustralia.com.au/training-and-events/conferences/unlock-excel. Hope to see you there!

Training in January and February

The start of the year generally brings about renewal – both in that new staff are joining teams, and existing staff take a breather after a long year past. The relative quiet that comes in January is often a good time to refresh your skills and ensure that you are equipped for the challenges in the year ahead.



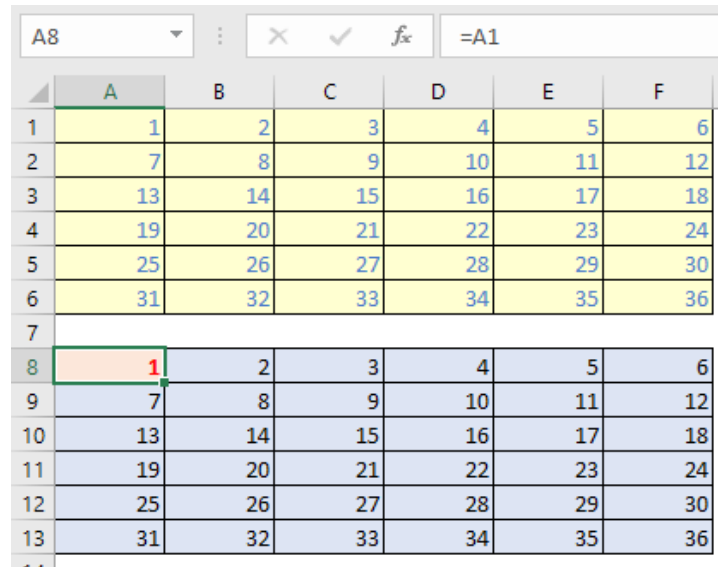
With that in mind, to help you in this process, SumProduct is offering a 10% discount on all course registrations made during January, for courses being held in January or February, with further discounts if you want to register for multiple courses or multiple attendees. Why not take the opportunity to train up new team members and pick up some tips and tricks for yourself? Email us at training@sumproduct.com or register your interest at www.sumproduct.com/courses and mention the code **JAN2018**.

Common Referencing Mistakes

A new year, a fresh start. Here's a resolution for you: consider cell referencing a little more carefully. We thought this month we'd bring up what should be a relatively straightforward area of modelling: referencing cells. However, as very experienced model auditors, it no longer surprises us how frequently modellers get this wrong...

Many model formulae are copied across rows and down columns. Some of these calculations are quite complex. It's important that the references in these calculations refer to the correct cells – and sometimes they do not. Some modellers seem to be perennially confused between relative, absolute and semi-absolute references.

Consider the following example:



	A	B	C	D	E	F
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
4	19	20	21	22	23	24
5	25	26	27	28	29	30
6	31	32	33	34	35	36
7						
8	1	2	3	4	5	6
9	7	8	9	10	11	12
10	13	14	15	16	17	18
11	19	20	21	22	23	24
12	25	26	27	28	29	30
13	31	32	33	34	35	36

Cell **A8** contains the formula **=A1**. This is what is known as a **relative** reference. As the formula is copied across and down through the range **A8:F13** the formulae reference the corresponding values in cells **A1:F6**, *i.e.* as I move across one column the reference moves one column to the right, as I move down one row, so does the reference and so on.

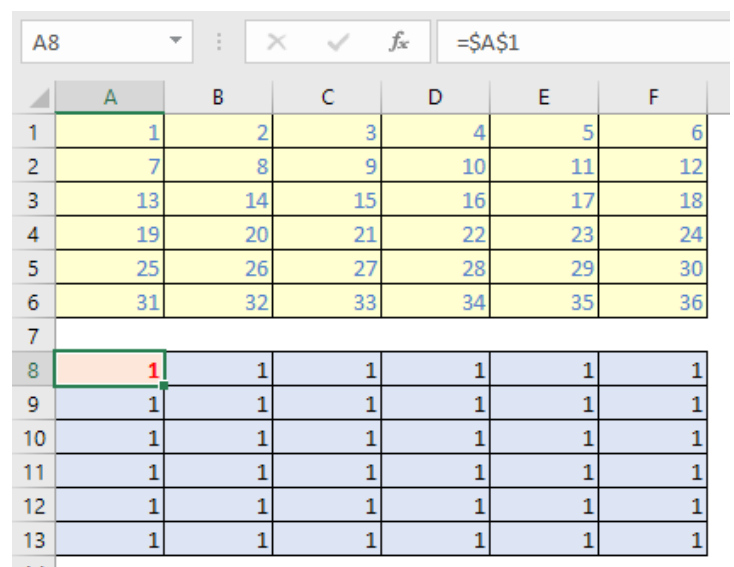
If I am in 'Edit' mode in Excel (*e.g.* by clicking in the formula bar or pressing the **F2** function key), the **F4** function key will toggle cell references as follows:

Pressing the F4 key repeatedly:

=A1
↓
=\$A\$1
↓
=A\$1
↓
=\$A1

Pressing **F4** whilst in 'Edit' mode for a formula (or part of a formula) toggles the dollar (\$) signs in the cell references. These signs anchor the row and column references. A dollar sign before a column reference anchors the column (that is, the column reference will not move when the formula is copied across a row); a dollar sign before a row reference anchors the row (that is, the row reference will not change when the formula is copied down a column). These dollar signs may simply be typed in; the function key entry is not necessary.

To show the effect, **=\$A\$1** produces →



	A	B	C	D	E	F
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
4	19	20	21	22	23	24
5	25	26	27	28	29	30
6	31	32	33	34	35	36
7						
8	1	1	1	1	1	1
9	1	1	1	1	1	1
10	1	1	1	1	1	1
11	1	1	1	1	1	1
12	1	1	1	1	1	1
13	1	1	1	1	1	1

This is known as an **absolute** reference. All cells point to the same reference: **A1**. The other two options produce slightly more variety though. For example, **=A\$1** produces

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
4	19	20	21	22	23	24
5	25	26	27	28	29	30
6	31	32	33	34	35	36
7						
8	1	2	3	4	5	6
9	1	2	3	4	5	6
10	1	2	3	4	5	6
11	1	2	3	4	5	6
12	1	2	3	4	5	6
13	1	2	3	4	5	6

In this example, the formulae allow cell references to refer to different values in row 1, but other rows may not be varied. This type of reference is known as a **semi-absolute** reference.

Finally, **=\$A1** produces

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
4	19	20	21	22	23	24
5	25	26	27	28	29	30
6	31	32	33	34	35	36
7						
8	1	1	1	1	1	1
9	7	7	7	7	7	7
10	13	13	13	13	13	13
11	19	19	19	19	19	19
12	25	25	25	25	25	25
13	31	31	31	31	31	31

In this final example, the column (A) is anchored instead – another example of a semi-absolute referencing. It is possible to have a formula that has all types of referencing in one calculation. The trick is to get the referencing right first time every time – and this comes with practice.

Talking of which, you may have read all of this so far and thought this article seems trivial. We wish it were. However, many modellers get referencing wrong all too often. Want to see how you fare?

Try the following. Give yourself, say, 30 seconds to attempt the following question: →

Referencing Example

Before

Multiplier 2.0x

x	1	2	3	4	5
1	2				
2					
3					
4					
5					

Consider cell **G16**. This cell contains the formula

$$=F16*G15*G12$$

This is producing a simple 'times table' grid, multiplied by a factor stated in cell **G12**. The challenge is simple: can you put the necessary dollar signs in the formula highlighted (cell **G16**) such that the formula may then be copied correctly into cells **G16:K20** as shown (below)?

x	1	2	3	4	5
1	2	4	6	8	10
2	4	8	12	16	20
3	6	12	18	24	30
4	8	16	24	32	40
5	10	20	30	40	50

Simple, yes? Try it now and see if you get it right. It is amazing how often people make mistakes and / or cannot do it quickly. This is a core skill in modelling and comes with practice. If you found it trickier than you thought you would, don't worry, I won't tell anyone but may I suggest you *practice, practice, practice*.

There's more though.

We get so used to linking cells from elsewhere in the same worksheet or a different sheet in the same workbook that it becomes second nature to expect relative cell referencing, e.g.

$$=A1$$

$$=Sheet1!A1$$

$$='Another Sheet'!A1$$

The problem is, this is not always the case. If you link from another *workbook*, whether it is open or not, the reference will be absolute, e.g.

$$=[Book2]Sheet1!A1$$

Take care as often modellers do not notice this and then formulae produce incorrect results when copied.

It's no safer if you link to other sheets from the same workbook! Believe it or not, this is the most subtle issue of all. Often, we link to cells on other worksheets and then refer back to a cell on the same worksheet which causes the sheet reference to be referred to unnecessarily.

For example, imagine we were in **Sheet1** cell **A1** and wrote the following formula:

$$=E7+(Sheet2!C4+Sheet2!C6)/Sheet1!E3$$

Do you see the **Sheet1!** reference is superfluous? Instead, I could have written the following:

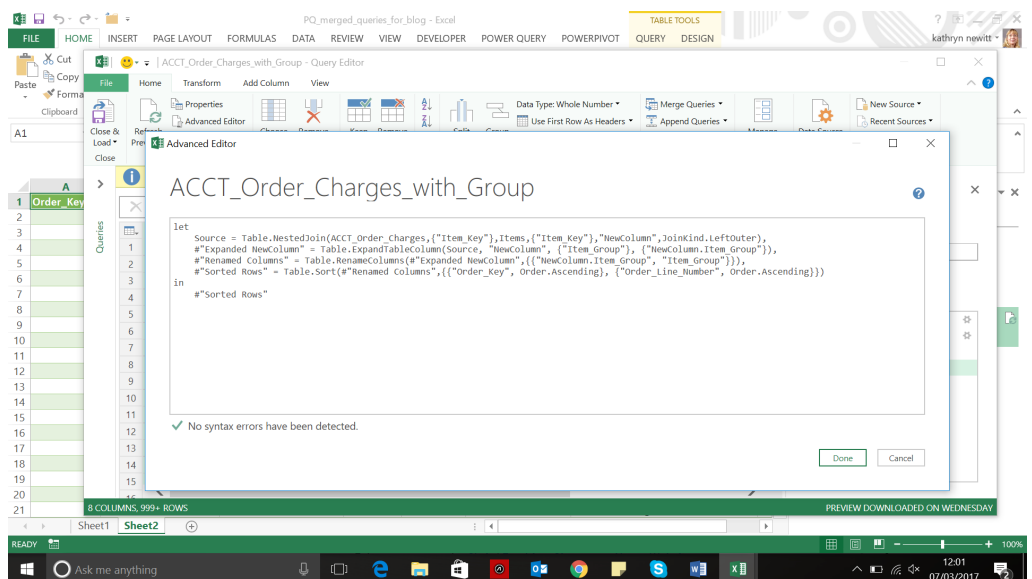
$$=E7+(Sheet2!C4+Sheet2!C6)/E3$$

Not only is this slightly easier to read, it's not as dangerous either. If this formula were on a worksheet that was copied, depending upon how the sheet was copied, the formula may always refer back to cell **E3** on **Sheet1** which may not be what was intended. Therefore, we strongly recommend that superfluous references are always removed.

Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Excel 2016) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month we continue to look at **M**, the programming language behind Power Query.

Following on from previous examples in this section, we are going to continue with **ACCT_Order_Charges_with_Group**. Having opened the Query Editor for **ACCT_Order_Charges_with_Group**, this time let's choose the 'Advanced Editor' from the 'Home' tab: →



This allows us to view a short section of **M** language, which demonstrates some of the **M** language syntax rules:

- Since this is a query and not a function, the first line of every query must begin with **'let'**, e.g.

```
'let
    Source = Table.NestedJoin( ...
    #"Expanded NewColumn" = Table...
    #"Renamed Columns" = Table...
    #"Sorted Rows" = Table...'
```

In this case I have four lines which refer to steps defining the source of the query and data manipulation (some of which are preceded by a **'#'** – but more on this in a minute). The lines are separated by a comma (',')

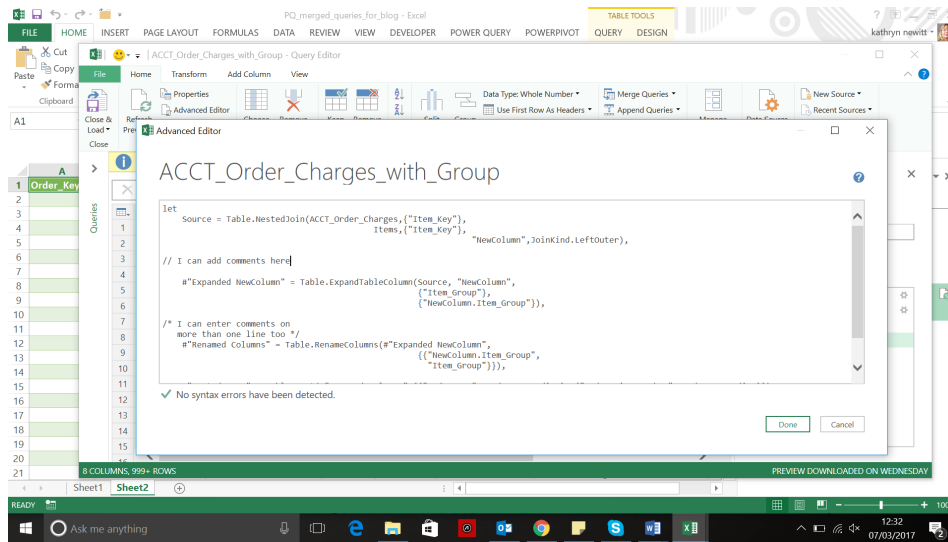
- The end of the query is indicated by **'in'** and a further step

'in

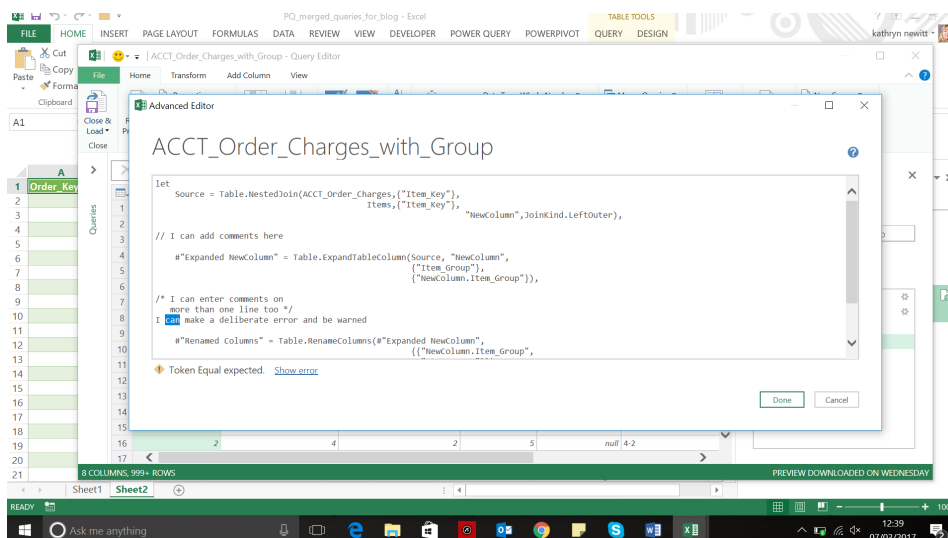
```
#"Sorted Rows" '
```

This tells Power Query which step to return to once the query has been executed, in this case **'Sorted Rows'**

- The **'#'** at the beginning of some of the lines is a message to Power Query to ignore the spaces in the names (e.g. in **'Sorted Rows'**). This is useful to know when creating steps manually
- The steps are sequential, and need to be linked to the previous step. Hence, the **'Sorted Rows'** step refers to the **'Renamed Columns'** step and the **'Renamed Columns'** step refers to the **'Expanded NewColumn'** and so on. When inserting lines, you must adjust the surrounding lines to make sure the sequence is intact, otherwise you will generate an error
- The lines are quite long and can get quite convoluted, so we may edit the format to make it easier to read:



Here, the lines have been split up to make them less of a long list of code – this can be useful when trying to keep track of whether all brackets have been closed (and closed in the right place). Also, a single comment has been inserted by preceding the line with **'//'**, and a section of comments by preceding with **'/*'** and ending with **'*/'**. There is a check at the bottom of the screen which will warn you if the syntax is wrong and allow you to jump to the error.



- Users of earlier versions of Power Query may notice that the **'token comma'** is now not needed after a comment section – be guided by the syntax checker.

Next month, we'll make more changes to the query in order to show how **M** language can be configured to operate on each row in a table...

INDEX, LOOKUP, VLOOKUP and **HLOOKUP** all require contiguous references. They refer to lists, row vectors, column vectors and / or arrays. **CHOOSE** is different:

=CHOOSE(index_num, value1, [value2]...)

As explained above, this function allows references to different calculations, workbook / worksheet references, etc. Try to use the function appropriately. For instance, a well-known Excel website proposes the following formula for calculating the US Thanksgiving date. Assuming cell **A1** has the year:

=DATE(A1,11,CHOOSE(WEEKDAY(DATE(A1,11,1)),26,25,24,23,22,28,27))

To understand this formula, note that **DATE(Year,Month,Day)** returns a date and **WEEKDAY(Date)** returns a number 1 (Sunday) through 7 (Saturday). But doesn't this formula look horrible? It is full of hard code and it contains an unnecessary number of arguments. The formula could exclude **CHOOSE** viz.

=DATE(A1,11,28-MOD(WEEKDAY(DATE(A1,11,1))+1,7))

Now let's be clear here. We are not saying this is a simple, transparent formula. Test it. They both provide the same answer. **CHOOSE** – and plenty of additional hard code – has been used unnecessarily.

That's not to say there isn't a time and a place for **CHOOSE**. It is useful when you need to refer to cells on different worksheets or in other workbooks. Some argue that it is useful when a calculation needs to be computed using different methods, e.g.

=CHOOSE(index_num, calculation1, calculation2, calculation3, calculation4)

It could be more transparent. Let me explain. In the example below, I have created a lookup table in cells **E10:E13** which I have called **Data** (I will explain how to create range names later). The calculations are all visible on the worksheet, rather than hidden away in the formula bar. The **index_num** selection, here referred to as **Selection_Number**, is input in cell **E2**. The result?

	A	B	C	D	E	F	G
1							
2				Selection	3		
3							
4				Result	88		
5							
6							
7				Calculations			
8							
9					Data		
10				Calculation 1	17		
11				Calculation 2	94		
12				Calculation 3	88		
13				Calculation 4	53		
14							
15							

It's identical, but easier to follow

=INDEX(Data,Selection_Number)

We have taught financial modelling to many gifted analysts over the years and a common mistake made by many is that they build *models that are easy to build rather than models that are easy to understand*. The end user is the customer. It should be simple to use: taking shortcuts invariably only helps the modeller – and even then, more often than not, shortcuts will backfire.

CHOOSE can lead to opaque models that need to be rebuilt and are often less flexible to use. You have been warned!

The A to Z of Excel Functions: CLEAN

Need to make a **CLEAN** getaway? This function removes all nonprintable characters from text. And we're not talking about expletives. You can use **CLEAN** on text imported from other applications that contains characters that may not print with your operating system. For example, you can use **CLEAN** to remove some low-level computer code that is frequently positioned at the beginning and end of data files and cannot be printed.

The **CLEAN** function employs the following syntax to operate:

CLEAN(text)

The **CLEAN** function has the following arguments:

- **text**: this is required and represents any worksheet information from which you want to remove nonprintable characters.

Please see our example below:

	A	B	C
1	Data		
2	=CHAR(9)&"Monthly Report"&CHAR(10)		
3			
4			
5	Formula	Description	Result
6	=CLEAN(A2)	Removes the nonprintable characters CHAR(9) and CHAR(10) from the text string in cell A2	Monthly Report
7			

The A to Z of Excel Functions: CODE

If you find it **CODE** outside, this function may assist you. **CODE** returns a numeric code (great dictionary definition) for the first character in a text string. The returned code corresponds to the character set used by your computer.

Operating Environment	Character Set
Macintosh	Macintosh character set
Windows	ANSI

The **CODE** function employs the following syntax to operate:

CODE(text)

The **CODE** function has the following arguments:

- **text**: this is required and represents the text for which you want the code of the first character.

Please our final example for this month below:

Formula	Description	Result
=CODE("A")	Displays the numeric code for A	65
=CODE("a")	Displays the numeric code for a	97
=CODE("!")	Displays the numeric code for !	33

Upcoming SumProduct Training Courses

Location	Course	Date	Duration
Sydney	Excel Tips & Tricks	15th Jan 2018	1 day
Sydney	Financial Modelling	16th - 17th Jan 2018	2 days
Sydney	Power Pivot, Power Query and Power BI	22nd - 24th Jan 2018	3 days
Melbourne	Power Pivot, Power Query and Power BI	29th - 31st Jan 2018	3 days
Hobart	Financial Modelling	5th - 6th Feb 2018	2 days
Melbourne	Power Pivot, Power Query and Power BI	4th - 6th Dec 2017	3 days
Brisbane	Excel Tips & Tricks	12th Feb 2018	1 days
Brisbane	Financial Modelling	13th - 14th Feb 2018	2 days
Melbourne	Excel Tips & Tricks	19th Feb 2018	1 day
Melbourne	Financial Modelling	20th - 21st Feb 2018	2 days
Perth	Financial Modelling	26th - 27th Feb 2018	2 days
Sydney	Visual Basic for Applications	12th Mar 2018	1 day
Sydney	Power Pivot, Power Query and Power BI	19th - 21st Mar 2018	3 days
Adelaide	Financial Modelling	5th - 6th Apr 2018	2 days
Sydney	Excel Tips & Tricks	23rd Apr 2018	1 day
Sydney	Financial Modelling	26th - 27th Apr 2018	2 days
Melbourne	Excel Tips & Tricks	30th Apr 2018	1 day
Melbourne	Financial Modelling	1st - 2nd May 2018	2 day
Melbourne	Power Pivot, Power Query and Power BI	14th - 16th May 2018	2 day
Auckland	Financial Modelling	21st - 22nd May 2018	2 days
Wellington	Financial Modelling	24th - 25th May 2018	2 days
Darwin	Financial Modelling	4th - 5th Jun 2018	2 days
Sydney	Power Pivot, Power Query and Power BI	18th - 20th Jun 2018	1 day
Sydney	Financial Modelling	25th - 26th Jun 2018	2 days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. Time for a New Year's resolution? Here are some we use all of the time:

Keystroke	What it does
ALT + F2	Save As
CTRL + 1	Format cells
CTRL + B	Embolden
CTRL + C	Copy
CTRL + F	Find dialog box
CTRL + F4	Close window
CTRL + G	Go To
CTRL + H	Replace dialog box
CTRL + HOME	Reset the sheet
CTRL + K	Insert hyperlink
CTRL + N	New workbook
CTRL + U	Underline (toggle)
CTRL + V	Paste
CTRL + X	Cut
CTRL + Y	Redo
CTRL + Z	Undo

There are over 540 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file a www.sumproduct.com/thought/keyboard-shortcuts. Also, check out our new daily **Excel Tip of the Day** feature on the www.sumproduct.com homepage.

Our Services

We have undertaken a vast array of assignments over the years, including:

- **Business planning**
- **Building three-way integrated financial statement projections**
- **Independent expert reviews**
- **Key driver analysis**
- **Model reviews / audits for internal and external purposes**
- **M&A work**
- **Model scoping**
- **Power BI, Power Query & Power Pivot**
- **Project finance**
- **Real options analysis**
- **Refinancing / restructuring**
- **Strategic modelling**
- **Valuations**
- **Working capital management**

If you require modelling assistance of any kind, please do not hesitate to contact us at contact@sumproduct.com.

Link to Others

These newsletters are not intended to be closely guarded secrets. Please feel free to forward this newsletter to anyone you think might be interested in converting to "the SumProduct way".

If you have received a forwarded newsletter and would like to receive future editions automatically, please subscribe by completing our newsletter registration process found at the foot of any www.sumproduct.com web page.

Any Questions?

If you have any tips, comments or queries for future newsletters, we'd be delighted to hear from you. Please drop us a line at newsletter@sumproduct.com.

Training

SumProduct offers a wide range of training courses, aimed at finance professionals and budding Excel experts. Courses include Excel Tricks & Tips, Financial Modelling 101, Introduction to Forecasting and M&A Modelling.

Check out our more popular courses in our training brochure:



Drop us a line at training@sumproduct.com for a copy of the brochure or download it directly from <http://www.sumproduct.com/training>.