Excel – Quick Reference Guide
The Most Important Functions and Formulas for Finance (IB, PE, HF/AM, ER, CF, etc.)

http://breakingintowallstreet.com

What This Guide Covers and How to Use It

Rather than re-explaining every single function here, we're mostly going to give you solid examples that demonstrate how to use each function. Note that this guide is **NOT** about keyboard shortcuts – the focus is on how to use the functions themselves.

Text and Formatting Functions

You will use these functions a moderate amount, much more so for due diligence and analysis of order/customer/other data than in financial models.

Function	What It Does
=LEFT(Text, # Characters)	Gives leftmost portion of text for # characters
=RIGHT(Text, # Characters)	Gives rightmost portion of text for # characters
=MID(Text, Start #, # Characters)	Gives text from Start # for # characters
=TRIM(Text)	Removes extra spaces
=PROPER(Text)	Makes first letter in each word of text uppercase
=CLEAN(Text)	Removes all non-printable characters from text
=UPPER(Text)	Makes text ALL UPPERCASE
=LOWER(Text)	Makes text all lowercase
=FIND(Text to Find, Within Text)	Finds starting position of text within text; case sensitive
=SEARCH(Text to Find, Within Text)	Same as above, but not case sensitive
=LEN(Text)	Gives # of characters in text
=SUBSTITUTE(Within Text, Text to Replace, Text to Swap In)	Replace text within text based on search for "Text to Replace"
=REPLACE(Within Text, Start #, # Characters, Text to Swap In)	Replace text within text based on character position at "Start #"
=VALUE(Text)	Converts text to number
=TEXT(Value, Format Text)	Shows number or text in different format

Here are examples for all these functions:

	A	B	C	D	E	F	G	H	I
1									
2		**Examples of these formulas in action:**							
3									
4		**Formula:**			**Text:**			**Result:**	
5									
6		=LEFT(E6,3)			Test text 123 1/1/2023			Tes	
7		=RIGHT(E7,3)			Test text 123 1/1/2023			023	
8		=MID(E8,6,8)			Test text 123 1/1/2023			text 123	
9									
10		=TRIM(E10)			too many spaces x			too many spaces x	
11		=PROPER(E11)			bob jones sr.			Bob Jones Sr.	
12		=CLEAN(E12)			•text•			text	
13									
14		=UPPER(E14)			Bob James Jr.			BOB JAMES JR.	
15		=LOWER(E15)			Bob James Jr.			bob james jr.	
16									
17		=FIND("123",E17)			Test test 123 1/1/2023			6	
18		=SEARCH("test",E18)			Test test 123 1/1/2023			1	
19		=LEN(E19)			Test test 123 1/1/2023			22	
20									
21		=SUBSTITUTE(E21,"123","456")			Test test 123 1/1/2023			Test test 456 1/1/2023	
22		=REPLACE(E22,11,3,"456")			Test test 123 1/1/2023			Test test 456 1/1/2023	
23									
24		=VALUE(E24)			3428			3428	
25		=TEXT(E25,"mmm dd , yyyy")			1/23/2021			Jan 23 , 2021	

The Text to Columns Function

This function (shortcut key: Alt + A + E or Alt + D + E, no shortcut on the Mac) can be **very** helpful if you've imported data separated by spaces, commas, periods, or other characters and you need to turn it into columns (see the example to the right).

Before:	**After:**		
Test1, Test2, Test3	Test1	Test2	Test3
Item1, Item2, Item 3	Item1	Item2	Item 3
List1, List2, List 3	List1	List2	List 3

However, there are some limitations: for one, all the data must be separated by the SAME character, so it couldn't be a mix of both commas and periods. Also, you still have to fix the data with TRIM, PROPER, and other functions even after splitting it apart.

Date and Time Functions

Date and time functions are not super-important in Excel, but you will use them when formatting the years, quarters, or months in financial models and calculating the next and previous periods.

Function	What It Does
=DATE (Year, Month, Day)	Creates date in Excel
=DATEVALUE (Text)	Converts text to real date
=YEAR (Date)	Returns year of date
=MONTH (Date)	Returns month of date
=DAY (Date)	Returns day of date
=NETWORKDAYS (Start Date, End Date)	Calculates # of business days between two dates
=EOMONTH (Start Date, # Months)	Last day of month after # months

And here are examples and use cases:

	A	B	C	D	E	F	G	H
1								
2		**Examples of these formulas in action:**						
3								
4		**Formula:**			**Date or Number:**			**Result:**
5								
6		=DATE(2023,1,1)			2023, 1, 1			1/1/2023
7		=DATEVALUE("12/31/2020")			"12/31/2020"			12/31/2020
8								
9		=YEAR(E9)			12/31/2020			2020
10		=MONTH(E10)			12/31/2020			12
11		=DAY(E11)			12/31/2020			31
12								
13		=NETWORKDAYS(E11,E13)			1/31/2021			22
14								
15		=EOMONTH(E13,E15)			1			2/28/2021
16		=EOMONTH(E13,E16)			5			6/30/2021
17		=EOMONTH(E13,E17)			15			4/30/2022
18		=EOMONTH(E13,E18)			-4			9/30/2020

Arithmetic and Logical Functions

You can already figure out what functions like SUM, AVERAGE, MAX, and MIN do simply by looking at them. The best way to learn / review these is with examples:

	A	B	C	D	E
1					
2				2	6
3				3	7
4				4	8
5				5	9
6		**Formula:**	**Result:**		
7		=SUM(D2:D5)	14		
8		Alt + = or ⌘ + Shift + T (Mac)		14	=SUM(E2:E7)
9					SUM(**number**
10		=COUNT(D2:D5)	4		
11		=AVERAGE(D2:D5)	3.5		
12		=MAX(D2:D5)	5		
13		=MIN(D2:D5)	2		
14					
15		=SUMIF(D2:D5,">=4",D2:D5)	9		
16		=SUMIFS(D2:D5,D2:D5,">=4",D2:D5,"<5")	4		
17					
18		=COUNTIF(D2:D5,">=3")	3		
19		=COUNTIFS(D2:D5,">=3",D2:D5,"<5")	2		
20					
21		=SUMPRODUCT(D2:D5,E2:E5)	110		
22					
23		=ABS(-4)	4		
24					
25		=IF(C11>3,"Avg. above 3", "Avg. below 3")	Avg. above 3		
26		=OR(C7>10,C7<12)	TRUE		
27		=AND(C7>10,C7<12)	FALSE		

While SUMIFS and COUNTIFS are very useful (since they let you do conditional sums and counts based on *multiple* criteria), they only work in versions of Excel from 2007 onward – which should be 99% of versions in use these days, but you never know…

Financial Functions

We cover many financial functions in the course, but you use 2 functions more than any others: **IRR** and **NPV** (and their close relatives, such as XIRR, MIRR, and XNPV).

You can think of IRR (Internal Rate of Return) as the "effective compounded interest rate on an investment."

It assumes that you invest some amount in the beginning (an outflow of cash), receive money back over the years (cash inflows), and then re-invest those funds at the same rate of return:

Demonstration of IRR Function:					
Initial Investment:					$ 1,000
Annual Cash Flow:					-
Investment Selling Value:					2,000
Year 0	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
$ (1,000)	$ -	$ -	$ -	$ -	$ 2,000
Invested Amount:					1,000
What We Get Back (Proceeds):					2,000
IRR:					**14.9 %**

This 14.9% number tells us: had we invested that $1,000 in the beginning and then earned 14.9% on it each year, added the interest to the principal (compounding), and then kept earning 14.9% on the entire amount, we'd end up with $2,000 at the end of 5 years.

If you don't receive the cash flows at regular intervals, XIRR (which accepts both the values and the dates of the cash flows) handles that case.

MIRR is a variation on IRR. We didn't cover it explicitly in the course, but it allows you to separate the **cost of the investment** from the **interest rate on re-investment**. So if there's no way to actually earn 14.9% on reinvested proceeds above, we could specify a different rate with MIRR.

NPV (Net Present Value) is another related concept. Instead of saying, "What is the effective compounded interest rate on an investment?" you're saying, "Let's say that we **know in advance**

what that rate is, *and* what the cash flows look like. How much could we pay upfront and still achieve the return that we're targeting?"

Demonstration of NPV Function:					
Initial Investment:					????
Annual Cash Flow:					100
Investment Selling Value:					2,000
What Return Are We Looking For?					15.0 %
Year 0	**Year 1**	**Year 2**	**Year 3**	**Year 4**	**Year 5**
$ -	$ 100	$ 100	$ 100	$ 100	$ 2,000
How much can we pay upfront for the discount rate (required rate of return) to be 0.0%?					$ 1,280

Often, you will use both of these functions together to estimate the viability of projects, new investments, and acquisitions. An investment can have a high IRR but low NPV, and vice versa.

XNPV allows you to account for irregular time intervals with the NPV function.

Other Functions

We also covered IPMT and PPMT for mortgage interest and principal repayment, and SLN, DDB, and SYD for calculating depreciation in the lesson on financial functions. Those are far less important than IRR and NPV, so we're not going through them here – take a look at the lesson and notes in the Excel files for more on those.

Lookups and Scenario Functions

You'll use these functions most often when manipulating, compiling, and summarizing data, but a few of them also have uses in financial models. Common use cases:

1. **VLOOKUP:** General data lookup; often used to find data for a public comp by its ticker.
2. **HLOOKUP:** Same, but often used to pull in source data for public comps.
3. **INDEX / MATCH:** Used if you don't know the row # and column # of what you're looking for… very common when you've pasted in disorganized data from PDFs or websites, which is what the REIT case study in this course is based on.

4. **INDIRECT:** Used to find data when you have lots of spreadsheets with very similar titles and you want to refer to a similar area or search for something similar across *all* those spreadsheets.
5. **ADDRESS:** Not as common as the others – might be helpful if you have a data set where the entire area is constantly changing or if you need to look up a value in a table, and then use *that* value that you've retrieved for other purposes (see the examples below).
6. **CHOOSE / OFFSET:** Used most frequently for scenarios in financial models, but OFFSET is very powerful since it returns a **range**. You will see it used with dynamic charts as well.

Function	What It Does
=VLOOKUP (Value, Table, Column #)	Looks up Value in Left Column and Returns cell in specified Column # and Row # that contains Value
=HLOOKUP (Value, Table, Row #)	Looks up Value in Top Row and Returns cell in specified Row # and Column # that contains Value
=MATCH (Value, Row or Column Range)	Finds Item's Position in Row/Column
=INDEX (Table, Row #, Col #)	Retrieves cell value at Row # and Column #
=INDIRECT (Reference)	Returns cell at reference given by Reference text
=ADDRESS (Row #, Col #)	Creates cell reference text
=CHOOSE (Number, Item1, Item2…)	Selects Item from List based on Number
=OFFSET(Cell, # Rows, # Cols)	Move # of Rows and Columns from Cell

First, let's look at a very simple example of how VLOOKUP and HLOOKUP work:

	A	B	C	D	E	F	G
1							
2		Sales Rep ID	Jan	Feb	Mar	Apr	
3		1	$ 159,182	$ -	$ 247,262	$ 156,746	
4		2	230,254	432,621	-	461,898	
5		3	1,135,702	1,274,247	763,525	441,812	
6		4	-	-	223,495	-	
7		5	384,465	272,842	284,841	-	
8		6	226,509	189,041	554,264	280,018	
9		7	162,928	551,062	254,536	652,300	
10		8	225,276	635,752	252,118	452,591	
11		9	464,931	-	537,280	165,772	
12							
13		Formula:			Result:		
14		=VLOOKUP(5,B2:F11,4,FALSE)			284,841		
15							
16		=HLOOKUP("Mar",B2:F11,6,FALSE)			284,841		

Both functions retrieve the same value from the table – the only difference is that with VLOOKUP, we're looking up the row # based on the values in the left-hand side of the table, whereas with HLOOKUP we're looking up the column # based on the values in the top row.

You should almost **always** use "FALSE" for the final input into the function because in 99% of cases you want an exact match. VLOOKUP and HLOOKUP have 2 big drawbacks:

1. You must know the row # or column # of the value you're looking for in the first place.
2. You can only go from left to right or from top to bottom, so you're out of luck if the value you're looking up is **NOT** in the left column of the table or in the top row of the table.

The INDEX and MATCH functions allow you to get around those limitations by finding the *position* of an item in a range (with MATCH) and then locating its precise location with the INDEX function, which accepts a table, row #, and column # as input:

Sales Rep ID	Jan	Feb	Mar	Apr
1	$ 159,182	$ -	$ 247,262	$ 156,746
2	230,254	432,621	-	461,898
3	1,135,702	1,274,247	763,525	441,812
4	-	-	223,495	-
5	384,465	272,842	284,841	-
6	226,509	189,041	554,264	280,018
7	162,928	551,062	254,536	652,300
8	225,276	635,752	252,118	452,591
9	464,931	-	537,280	165,772

Formula:	Result:
=VLOOKUP(5,B2:F11,4,FALSE)	284,841
=HLOOKUP("Mar",B2:F11,6,FALSE)	284,841
=MATCH("Mar",B2:F2,0)	4
=MATCH(5,B2:B11,0)	6
=INDEX(B2:F11,6,4)	284,841
=INDEX(B2:F11,MATCH(5,B2:B11,0),MATCH("Mar",B2:F2,0)	284,841

At first glance, you may not even see the advantage of INDEX/MATCH or how they're different.

But there is a BIG difference: with INDEX and MATCH, you are *searching* for the position of specific values in the table and retrieving what's at that intersection – you do NOT need to know the row # and column # of what you're looking for in advance.

Yes, it is more complicated to write this function with the INDEX/MATCH method rather than with VLOOKUP or HLOOKUP, so for simple lookups you're better off with one of those. For anything complex or subject to change, though, INDEX and MATCH are a must.

The other functions here are more straightforward / less important, but let's look at a few simple examples to see them in action as well:

	A	B	C	D	E	F	G
1							
2		Sales Rep ID	Jan	Feb	Mar	Apr	
3		1	$ 159,182	$ -	$ 247,262	$ 156,746	
4		2	230,254	432,621	-	461,898	
5		3	1,135,702	1,274,247	763,525	441,812	
6		4	-	-	223,495	-	
7		5	384,465	272,842	284,841	-	
8		6	226,509	189,041	554,264	280,018	
9		7	162,928	551,062	254,536	652,300	
10		8	225,276	635,752	252,118	452,591	
11		9	464,931	-	537,280	165,772	
12							
13		**Formula:**		**Result:**			
25		=INDIRECT("C4")		230,254			
26							
27		=ADDRESS(4,3)		$C$4			
28							
29		=INDIRECT(D27)		230,254			
30							
31		=CHOOSE(3,C2,D2,E2,F2)		Mar			
32							
33		=OFFSET($B$2,2,1)		230,254			
34							
35		=OFFSET($B$2,2,1,2,2)		230,254	432,621		
36				1,135,702	1,274,247		

These are all different ways to reference cells – but is there any practical value to these functions?

INDIRECT is helpful when you want to refer to the *same* or a *similar* position across multiple spreadsheets, and you can assemble the reference inside the function (e.g. "Sheet1!C4").

ADDRESS is less useful in traditional finance roles, but it could be helpful when you use it in conjunction with MATCH to locate the positions of items and then retrieve the references for other purposes.

CHOOSE and OFFSET are mostly used for scenarios, so please see that lesson on the site if you need a refresher. Some people don't know that OFFSET actually returns a **range** if you specify a width and height – this can be useful when you're extracting a smaller data set from a larger one, or when you're creating a dynamic chart.

Database and Array Functions

Database functions are useful for summarizing and aggregating data across large tables.

They are **not** as useful for simple lookups because they require "setup" in the beginning, but it is much easier to do complex SUMIFs, SUMPRODUCTs, COUNTIFs, and so on with database functions than with those functions plus complex lookup functions. Let's go to the example:

	A	B	C	D	E	F	G	H
1								
2		Order Date	Order Dollar Amount	State	Region	Sales Rep ID		
3		9/1/2020	$ 289,469	AZ	Southwest	5		
4		11/22/2020	288,875	GA	Southeast	9		
5		12/18/2020	288,652	FL	Southeast	2		
6		12/6/2020	287,356	RI	Northeast	2		
7		10/15/2020	286,721	OH	Midwest	6		
8		6/4/2020	286,017	NE	Midwest	8		
9		2/15/2020	285,938	RI	Northeast	7		
10		7/20/2020	285,239	NY	Northeast	3		
11		3/28/2020	284,841	MS	Southeast	5		
12		8/29/2020	283,025	AL	Southeast	1		
13								
14		Order Date	Order Dollar Amount	State	Region	Sales Rep ID		
15		>2/1/2020	>50000		Midwest			
16								
17		Order Total:				572,738	=DSUM(B2:F12,C2,B14:F15)	
18		Order Count:				2	'=DCOUNT(B2:F12,C2,B14:F15)	

The field names MUST match what's in the table or this will NOT work. Also, your conditions must make logical sense; to exclude a condition, just leave the field (the B15:F15 range) blank.

In addition to DCOUNT and DSUM, there's also DCOUNTA (only counts non-blank cells) and DGET for extracting a single row that matches specific criteria, but those are both less useful.

These functions will **not** work properly if you have blanks or other irregular data in your table, so make sure that the underlying data is good first.

Array Functions

You use **array functions** when you want to perform an operation on *multiple* cells and then return *multiple* values.

Array functions are useful for transposing data (using the built-in TRANSPOSE function) to convert rows to columns or vice versa, and also for writing functions like MINIF and MAXIF that don't exist as built-in functions in Excel.

You **MUST** press Ctrl + Shift + Enter after entering an array function or it won't work correctly. It's represented by the "{" and "}" in the screenshot below (just entering those symbols won't make it work – you **NEED** to press Ctrl + Shift + Enter). Let's look at two examples:

	A	B	C	D	E	F	G
1							
2		Sales Rep ID	Jan	Feb	Mar	Apr	
3		1	$ 159,182	$ -	$ 247,262	$ 156,746	
4		2	230,254	432,621	-	461,898	
5		3	1,135,702	1,274,247	763,525	441,812	
6		4	-	-	223,495	-	
7		5	384,465	272,842	284,841	-	
8		6	226,509	189,041	554,264	280,018	
9		7	162,928	551,062	254,536	652,300	
10		8	225,276	635,752	252,118	452,591	
11		9	464,931	-	537,280	165,772	
12							
13		Formula:				Result:	
14		{=MIN(IF($B$3:$B$11>5,$C$3:$C$11,""))}				162,928	
15							
16		{=MAX(IF($B$3:$B$11>5,$C$3:$C$11,""))}				464,931	
17							
18		{=TRANSPOSE(C2:F2)}				Jan	
19						Feb	
20						Mar	
21						Apr	

These are fairly simple examples – all we're doing in the first two is calculating the minimum order dollar amount for January for all sales reps with IDs *higher* than 6. But you can see how even that simple function requires an array function in Excel if you want to make it truly flexible.

The second case is far more common: we're using TRANSPOSE to flip a row into a column. You could also go to Paste Special and select Transpose from that menu, but this method is better because it directly links the row and flipped column.

How to Audit Formulas and Leave Comments

We'll conclude this quick reference guide with an important topic that is not *exactly* a new function, but which is very relevant to the topic: **how to audit formulas and find errors.**

In Excel 2007+, you can use the IFERROR function to check for errors like #REF! and #N/A and then return a message like "Error" if the calculation produces one of those errors:

- =IFERROR(2/0,"Error!")

Be careful when using this function because you do *not* necessarily want to "handle" all errors in your models. Sometimes, you want to **see** these errors so that you can fix mistakes as you move along.

This function is **BEST** in cases where you can't predict in advance whether or not there will be an error – for example, if you're using VLOOKUP or HLOOKUP and there's a chance that the value you're looking up may not actually exist in the range.

If you are somehow using Excel 2003 or another ancient version, IFERROR doesn't exist so you'll have to use IF(ISERROR(Expression), "Error Message", Expression) instead.

"Auditing" your formulas just means quickly highlighting **immediate** precedents and dependents, as well as **all** precedents and dependents, tracing the formulas, and seeing where everything is coming from.

These are not even true "functions" like everything else on here.

Instead, they are just normal keyboard shortcuts:

Function	PC Excel 2007+	Mac Excel	PC Excel 2003
Immediate Precedents	Ctrl + [	Ctrl + [	Ctrl + [
Immediate Dependents	Ctrl +]	Ctrl +]	Ctrl +]
Trace Precedents	Alt + M + P	*No equivalent*	Alt + T + U + T
Trace Dependents	Alt + M + D	*No equivalent*	Alt + T + U + D
Erase Traces	Alt + M + A + A	*No equivalent*	Alt + T + U + A
All Precedents	Shift + Ctrl + {	Shift + Ctrl + {	Shift + Ctrl + {
All Dependents	Shift + Ctrl + }	Shift + Ctrl + }	Shift + Ctrl + }
Jump to Original Cell	F5 + Enter	F5 + Enter	F5 + Enter

Here's what it looks like when you do this and highlight (all) precedents:

Membership Fees ($MM):	$ 2,953	$ 2,897	$ 3,096	$ 2,982	$ 2,982	$ 2,982	$ 2,982	$ 2,982

Income Statement - Wal-Mart Stores, Inc.							
($ in Millions, Except per Share Figures)	Historical		Projected				
	1/31/2011	1/31/2012	1/31/2013	1/31/2014	1/31/2015	1/31/2016	1/31/2017
Revenue:	$ 421,849	$ 446,950	$ 471,317	$ 493,709	$ 515,675	$ 534,773	$ 551,404
Revenue Growth:		*6.0 %*	*5.5 %*	*4.8 %*	*4.4 %*	*3.7 %*	*3.1 %*
Cost of Revenue:	314,946	335,127	352,638	369,391	385,826	400,116	412,559
Cost of Revenue % Revenue:	*74.7 %*	*75.0 %*	*74.8 %*	*74.8 %*	*74.8 %*	*74.8 %*	*74.8 %*
Gross Profit:	106,903	111,823	118,679	124,318	129,849	134,658	138,845
Gross Margin:	*25.3 %*	*25.0 %*	*25.2 %*	*25.2 %*	*25.2 %*	*25.2 %*	*25.2 %*

And then when you trace precedents:

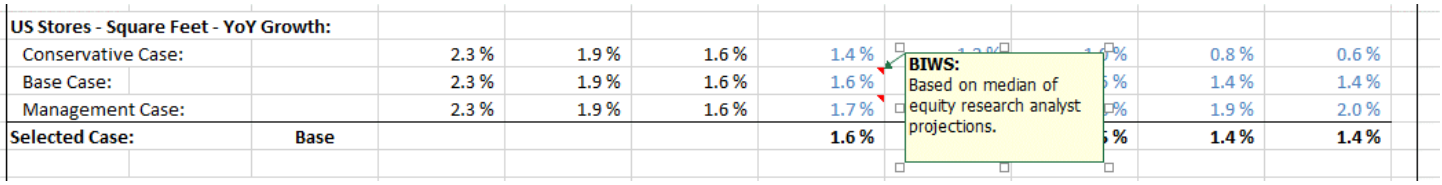
Income Statement - Wal-Mart Stores, Inc.							
($ in Millions, Except per Share Figures)	Historical		Projected				
	1/31/2011	1/31/2012	1/31/2013	1/31/2014	1/31/2015	1/31/2016	1/31/2017
Revenue:	$ 421,849	$ 446,950	$ 471,317	$ 493,709	$ 515,675	$ 534,773	$ 551,404
Revenue Growth:		*6.0 %*	*5.5 %*	*4.8 %*	*4.4 %*	*3.7 %*	*3.1 %*
Cost of Revenue:	314,946	335,127	352,638	369,391	385,826	400,116	412,559
Cost of Revenue % Revenue:	*74.7 %*	*75.0 %*	*74.8 %*	*74.8 %*	*74.8 %*	*74.8 %*	*74.8 %*
Gross Profit:	106,903	111,823	118,679	124,318	129,849	134,658	138,845
Gross Margin:	*25.3 %*	*25.0 %*	*25.2 %*	*25.2 %*	*25.2 %*	*25.2 %*	*25.2 %*
Sales, General & Administrative Expenses:	81,361	85,265	90,408	94,703	98,916	102,580	105,770
% Revenue:	*19.3 %*	*19.1 %*	*19.2 %*	*19.2 %*	*19.2 %*	*19.2 %*	*19.2 %*
Operating Income:	25,542	26,558	28,272	29,615	30,932	32,078	33,076

These functions *can* be useful, but they are not always as useful as you might think.

They can help you track down simple errors, but unless your error is using the wrong number or missing a number in a formula, you won't find the root cause of problems very quickly with these.

Ctrl + [, the simplest shortcut here, is actually the most useful one for real error-tracing.

Finally, you can also add **comments** to key cells (best used to explain assumptions and/or complicated formulas) with Shift + F2 in any version of Excel:



To delete comments, use Alt + R + D (2007+), Alt + E + A + M (2003), or Shift + F2 + Del (Mac).

Leaving comments is not directly related to auditing formulas, but we're still inserting it here because comments can be very helpful in explaining your model and digging into other peoples' models and detecting errors there.