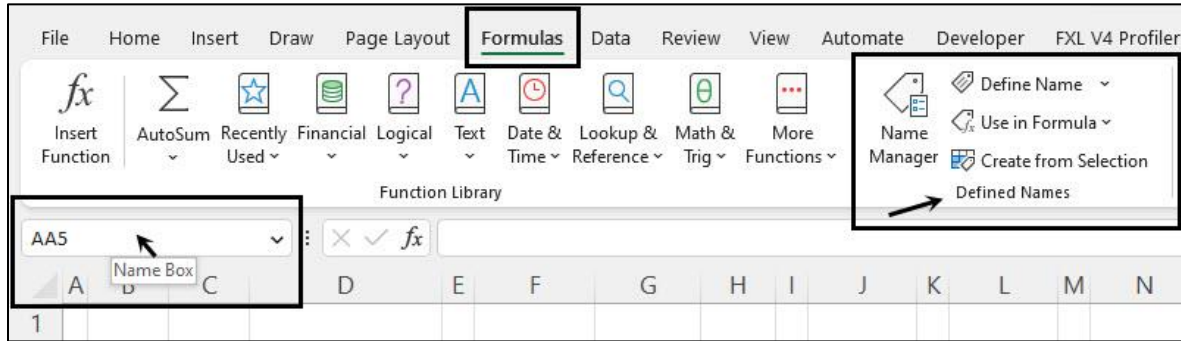# M 365 Excel Class Video 10: LET & LAMBDA Functions and Single Cell Reports
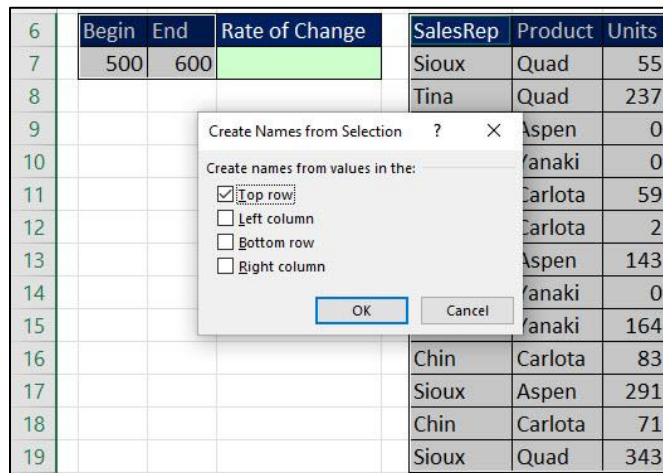
Defined Names

- Picture of Defined Names group in Formula tab in Excel Ribbon:

- Defined Names allow you to assign a name to:
    1. A cell
    2. A range of cells
    3. Non-contiguous ranges of cells
    4. Worksheet Formulas
    5. LAMBDA functions and then use the Defined Names in Formulas.
    6. Excel Table Names, Print Areas and Criteria and Extract Ranges for Advanced Filter for automatically created and listed with all Defined Names.

- Create Defined Names:
    1. You can create a Defined Name for a ***Cell, Range, or Non-Contiguous Ranges of Cells*** by:
        - Selecting cell, range or ranges
        - Type Name in Name Box (shown in picture above)
            - Rules for Defined Names:
                - Must begin with text, not number
                - No spaces
                - No cell references
                - Max of 255 characters in name
                - Cannot use the characters: */+-()^<>+&%~`|][}{@";:,'$#!
        - Press Enter.
    2. You can create a Defined Name for a ***Cell or Range*** when the name you want to use is above the cell or range, use the keyboard Ctrl + Shift + F3 to open the Create Names From Selection Dialog Box, as shown here:

3. You can create a Defined Name for a *Formula or LAMBDA Function* using the New Name Dialog Box by:
   - Open Name Manager Dialog Box:
     - Formulas tab in Excel Ribbon, Defined Name group, Name Manager button (shown in above figure).
       or
     - Use the keyboard to open the Name Manager = **Ctrl + F3**
     - Name Manager Dialog Box looks like this:

   - To create a new Defined Name, click the New… button. This opens the New Name Dialog Box as shown below
     - This Defined Name **defines a worksheet formula** that can be called with an equal sign and the name "ZeroToTen":
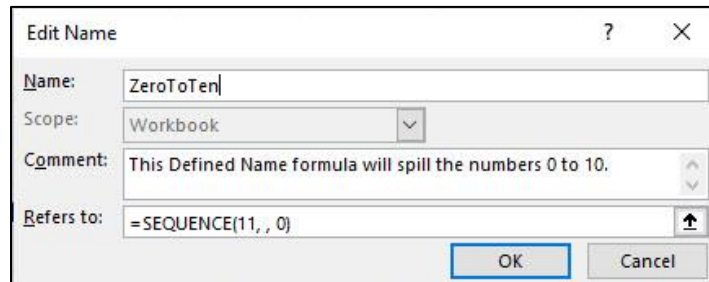
     - This Defined Name **defines a lookup table** based on an Array Constant:
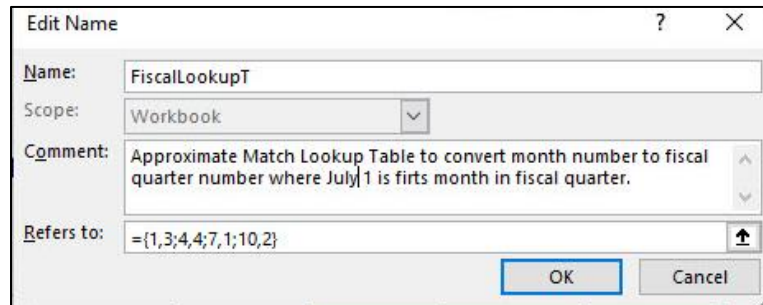
     - This Defined Name **defines a LAMBDA re-useable custom function** that will create a new function available in the workbook that will calculate fiscal quarter from a serial number date:
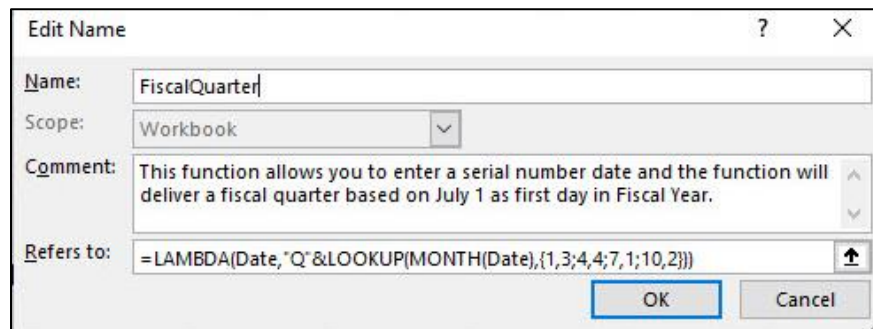
- **Use Defined Names**:
    1. To Jump to a Defined Name location in workbook, use Name Box dropdown to select name, as shown here:
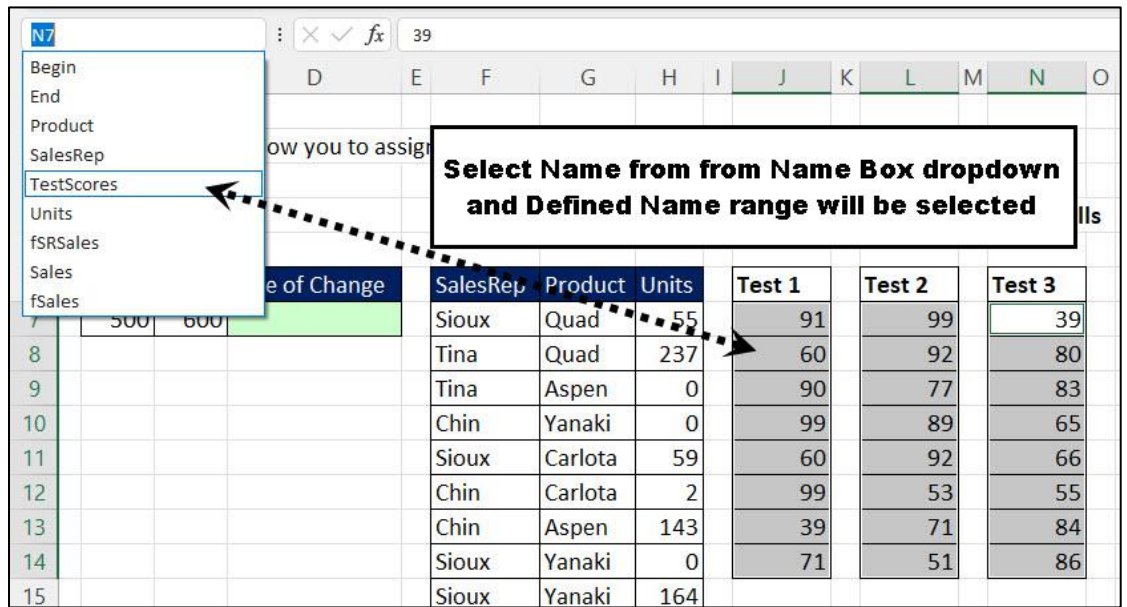
    2. To use a Defined Name that is a worksheet formula:
        i. Type an equal sign and then the first few letters in name, as shown here:
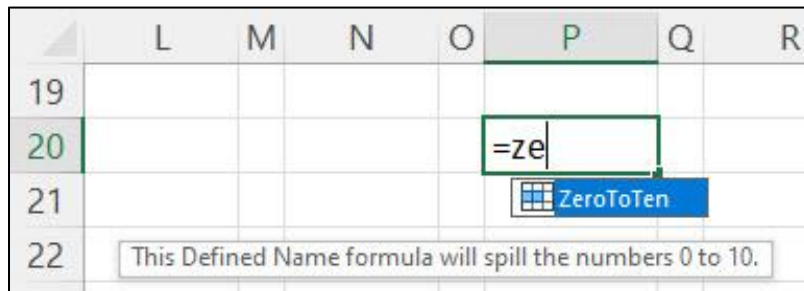
        ii. After entering Defined Name worksheet formula, this is what you see:
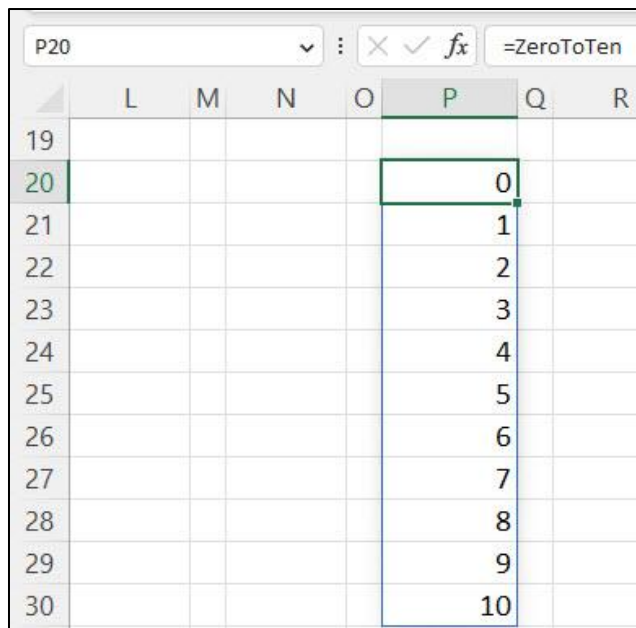
3. To use a Defined Name that is a lookup table:
   i. In lookup formula, type first few letters in name, as shown here:

U	V	W	X	Y	Z	AA	AB	AC
Date	Fiscal Q	Fiscal Q	Fiscal Q					
1/4/24	="Q"&LOOKUP(MONTH(U9),FiscalL							
5/19/24	LOOKUP(lookup_value, **lookup_vector**, [result_vector])							
9/9/24	LOOKUP(lookup_value, **array**) FiscalLookupT							

Approximate Match Lookup Table to convert month number to fiscal quarter number where July 1 is firts month in fiscal quarter.

   ii. After entering Defined Name lookup Table, this is what you will see:

X ✓ fx   ="Q"&LOOKUP(MONTH(U9),FiscalLookupT)

T	U	V	W
	Date	Fiscal Q	Fiscal Q	Fisc
	1/4/24	Q3		
	5/19/24	Q4		

4. To use a Defined Name that defines a LAMBDA function:
   i. Type an equal sign and then the first few letters in LAMBDA defined function name, as shown here:

U	V	W	X	Y	Z	AA	AB	AC	AD
Date	Fiscal Q	Fiscal Q	Fiscal Q						
1/4/24	Q3	Q3	=FiscalQ|						
5/19/24	Q4	Q4	fx FiscalQuarter						

This function allows you to enter a serial number date and the function will deliver a fiscal quarter based on July 1 as first day in Fiscal Year.

   ii. After invoking the LAMBDA defined function and entering the serial number date (as a single cell or as a range):

	T	U	V	W	X	Y	Z	AA
9				**LAMBDA #1**				
10				**Goal:** Create Fiscal Quarter Label from Date				
11								
12		Date	Fiscal Q	Fiscal Q	Fiscal Q			
13		1/4/24	Q3	Q3	=FiscalQuarter(U13:U23)			
14		5/19/24	Q4	Q4	C FiscalQuarter(**Date**)			

Here is what the Name Manager (Ctrl + F3) looks like when you have many Defined Names (Selected Cells Icon) and Excel Tables (Table Icon):

1. **Name column** = name of Defined Name or Excel Table.
2. **Value column** = preview of value if available (formulas and functions show no values).
3. **Refers to column** = formula that defined a cell, range, noncontiguous ranges, worksheet formulas, LAMBDA defined functions, Excel Tables, memorized Print Areas and memorized Criteria and Extract ranges from Advanced Filter feature.
4. **Scope column** = tells you where you can use the Defined Name:
   - **Workbook** => means you can use it anywhere in workbook.
   - Worksheet Name, like "**Topic**" => means you can only use it on that specified worksheet.
   - **Defined Names** => is a label given to the automatic Defined Names that are created when you use the Advanced Filter Feature. The two names created are "Criteria" and "Extract".
5. **Comment** = Comment added to help users understand what the Defined Name does. When you add a comment to a LAMBDA defined custom function, it appears in the function screen tip when you are typing the function name in a formula.
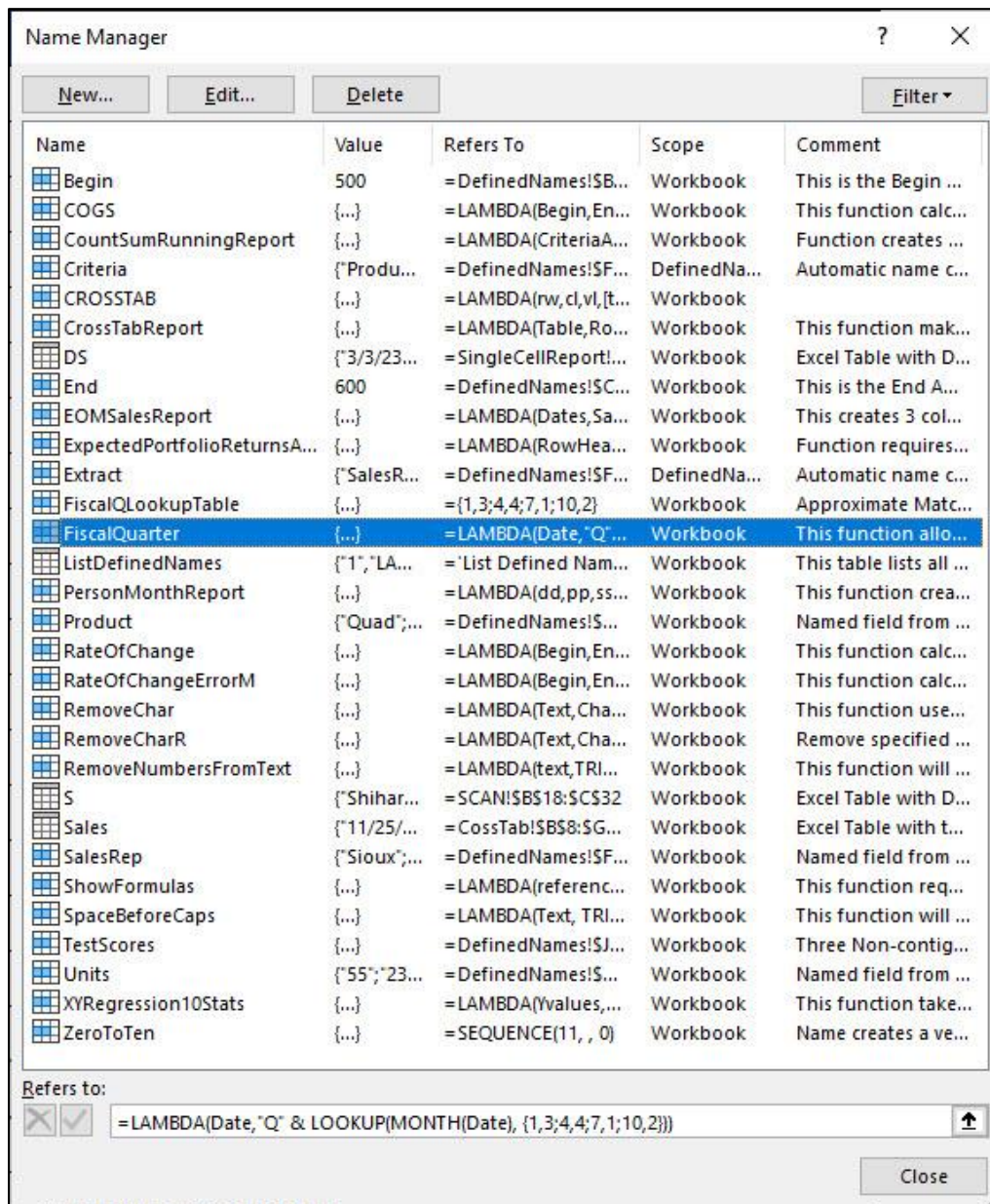
- Defined Name Keyboards:
    1. Open the Name Manager = **Ctrl + F3**
    2. Create Names From Selection = **Ctrl + Shift + F3**
    3. Paste Name = **F3**
    4. List all Defined Names in Worksheet: **F3, then click "Paste Names"**.

- **Defined Name Warning**: Worksheets copied from one workbook to another bring all Defined Names from source workbook into destination workbook.

LAMBDA Function

- Define LAMBDA function
    - The LAMBDA function allows to create a custom function value, which:
        1] Can be stored in a Defined Name to create a reusable function
        or
        2] Can be used in one of six LAMBDA Helper Functions for specific tasks such as spilling an aggregate calculation down a set of rows. When you use LAMBDA in a helper function you can use the formula directly in the worksheet or you can store it in a Defined Name to create a reusable function.
    - The arguments for the LAMBDA function are shown here:

LAMBDA([parameter1,parameter2,… parameter_n], calculation)

parameter = function argument name which shows up in function screen tip. Parameters are **formula inputs** used in calculation argument to define what the function will do.

calculation argument = formula you want to execute and return as the result of the function. This formula uses the parameters defined in the first part of LAMBDA.

Example of reusable worksheet function:
=LAMBDA(Begin ,End , End/Begin-1)   What it looks like in worksheet:

Parameter / formula input #1
Parameter / formula input #2
Formula that LAMBDA function executes

=RateOfChange(
RateOfChange(**Begin, End**)

Example of LAMBDA inside helper function:

LAMBDA Helper Function: **BYROW**
LAMBDA used inside: **BYROW**
Formula that LAMBDA function executes

	Jan	Feb	Mar	Apr	Spilled Row Total
10	Jan	Feb	Mar	Apr	Spilled Row Total
11	500	600	600	3,900	=BYROW(B11:E15,LAMBDA(r,SUM(r)))
12	1,000	1,800	2,800	2,700	BYROW(array, [function])
13	2,500	6,250	4,000	3,700	16,450
14	250	275	2,100	2,050	4,675
15	100	200	750	1,550	2,600

Parameter / formula input = r = each row in BYROW array

We will also see "Eta-Lambdas" to help shorten this formula to: BYROW(B11:E15, SUM)

- **Steps to create a Defined Name LAMBDA custom function for a Rate of Change calculation**
    1. In the worksheet type your LAMBDA function, as shown below. Note, the parameters that you enter in the first few arguments of LAMBDA, become the argument labels in the function screen tip. This means that you should name the parameters in a way that can help the user to understand what to enter in the function arguments.

	A	B	C	D	E	F	G
1							
2		LAMBDA #2					
3		Goal: Create reusable function to calculate rate of change.					
4							
5		Begin	End	Rate of Change			
6			500	600 =LAMBDA(Begin,End,End/Begin-1)			

If you enter the LAMBDA into the worksheet without testing the formula with formula inputs, you will see this:

5	Begin	End	Rate of Change	
6	500	600	#CALC!	D6: =LAMBDA(Begin,End,End/Begin-1)

   2. To test the LAMBDA function in the worksheet, you type open parentheses, the formulas inputs in the same order as you entered the parameters in first part of the LAMBDA function, then close parentheses, as shown here:

5	Begin	End	Rate of Change	
6	500	600	=LAMBDA(Begin,End,End/Begin-1)(B6,C6)	

5	Begin	End	Rate of Change	
6	500	600	0.20	D6: =LAMBDA(Begin,End,End/Begin-1)(B6,C6)

3. To create a Defined Name for the LAMBDA function (this allows you to re-use function anywhere in workbook), copy the LAMBDA function you created in the worksheet (without the testing parentheses at end) and then:
    i. Use Ctrl + F3 to open Name Manager
    ii. Click New Name button
    iii. In the New Name dialog box, enter:
        1) **Name:** The name of the function (will appear in function dropdown list when you create formulas)
        2) **Scope:** Where you want the function to appear: in Workbook (available everywhere in workbook) or a specified worksheet name.
        3) **Comment**: Create a description of the function (this appears in function screen tip when typing function name in a formula)
        4) **Refers to**: Paste the LAMBDA function you created in the worksheet (without the testing parentheses at end).
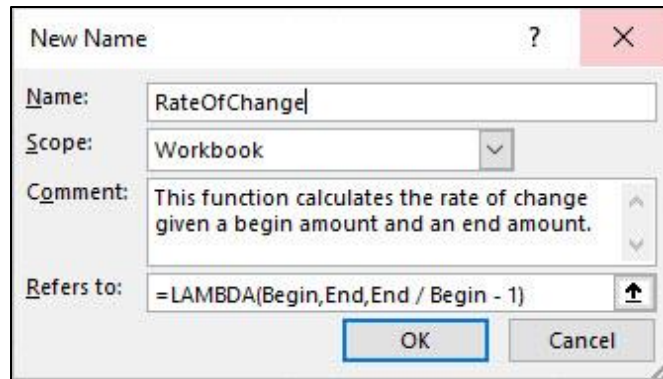
4. Test new function in worksheet by typing an equal sign and then the first few letters in function name (screen tip with description pops up), as shown here:

5. Test new function with arguments, as shown here:

6. Test the output of the new Defined Name LAMBDA function, as shown here:

- You can use the Advanced Formula Environment to create a Defined Name LAMBDA custom functions
  - The Advanced Formula Environment is an Excel Worksheet Add-in that is not installed by default in your Excel version. As shown below, you can install it with the Get-Add-ins button in the Add-ins group in the Home tab in the Excel Ribbon. However, many work entities have the Add-ins feature disabled. You can contact your network administrator to have it installed. Once you have it installed, you can use the Excel Labs button from the Home tab to open the Advanced Formula Environment.
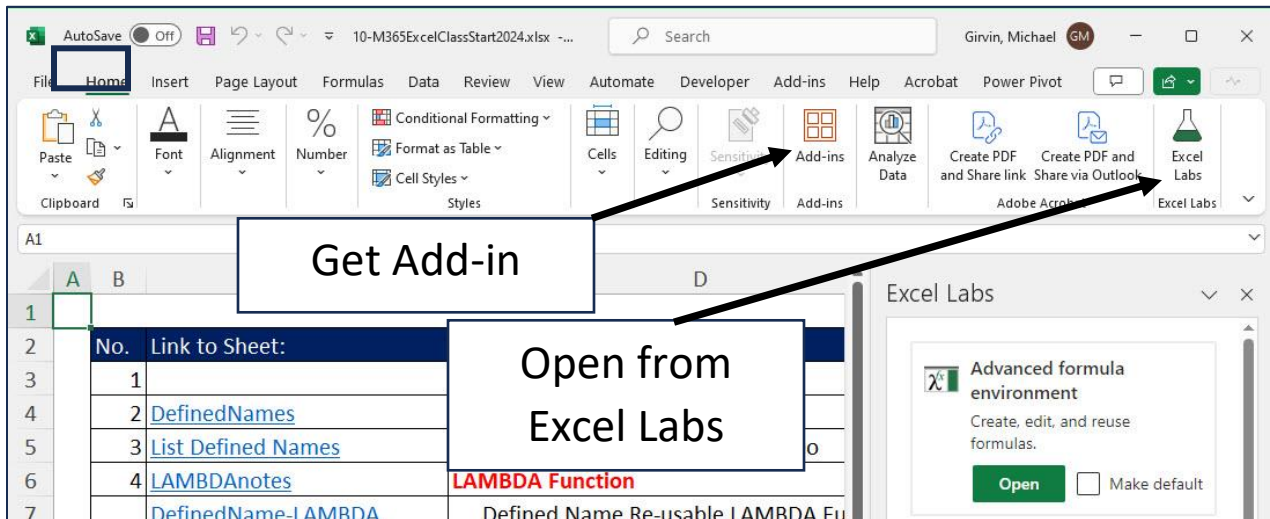
Using Advanced Formula Environment:

1. When you create a LAMBDA defined re-usable custom function, you can use the Advanced Formula Environment Task Pane as a user interface for the Defined Name Dialog Box. When you create a LAMBDA function in the Advanced Formula Environment Task Pane it is saved in the Defined Name Dialog Box. The reverse is also true: When you create a LAMBDA function in the Defined Name Dialog Box, it will appear in the Advanced Formula Environment Task Pane.
2. The advantage of using the Advanced Formula Environment Task Pane is that it is easier to type and create a LAMBDA function in the Advanced Formula Environment Task Pane than it is in the Defined Name Dialog Box.
3. The disadvantage of using the Advanced Formula Environment Task Pane is that you cannot directly access references from the worksheet, and you cannot directly test it in the worksheet.
4. An example of the Advanced Formula Environment Task Pane is shown to the right:
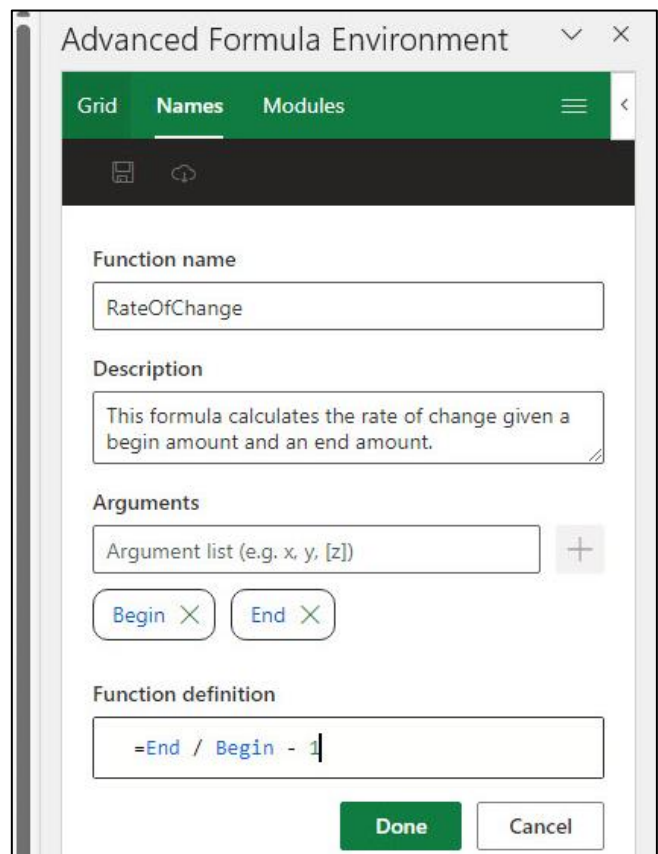
- Steps to create a Defined Name LAMBDA custom function for a COGS calculation
    1. **Create and test in worksheet:**

10	**LAMBDA #3:**				
11	**Goal:** Calculate Cost of Goods Sold (COGS) in Accounting. COGS = (Beg-End)*Value Each				
12					
13	Product	Beginning	End Quantity	Value Each	COGS
14	Quad	239	178	21.37	=LAMBDA(Begin,End,Value,(Begin-End)*Value)(C14,D14,E14)

    2. **Create Defined Name LAMBDA custom function:**

New Name		?	✕
Name:	COGS		
Scope:	Workbook	∨	
Comment:	This function calculates Cost Of Goods Sold (COGS) based on number of units at the beginning of the period (Begin), number of units at the end of the period (End) and the value per unit (Value) using the formula: (Begin-End)*Value.		
Refers to:	=LAMBDA(Begin,End,Value,(Begin - End) * Value)	⬆	
		OK	Cancel

    3. **Use new COGS function in worksheet:**

12					
13	Product	Beginning	End Quantity	Value Each	COGS
14	Quad	239	178	21.37	=COGS(C14,D14,E14)
15					COGS(Begin, End, Value)

- Save LAMBDA functions to Excel file that opens with Ctrl + N:
    1. Create new Excel file with a single worksheet
    2. Add Defined Name LAMBDA functions to workbook
    3. Save file as Excel Template file with name book.xltx to C:\Dingo Folder:

📁 C:\Dingo

Name

📗 book.xltx
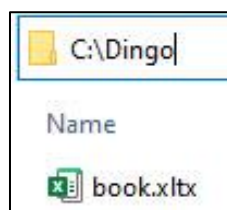
    4. When you save this custom file:
        - Blank Workbook button DOES NOT open custom file.
        - Open Excel App and click Esc (not Blank Workbook button) DOES open custom file.
        - Ctrl + N opens custom file.
    5. Templates, to always open with Excel and also using the CTRL+N command, can be copied in the C:\Program Files\Microsoft Office\Root\Office16\XLSTART\ folder, with administrator permissions.

- Defined Name LAMBDA custom function for a show formula text in vertical column formula

LAMBDA #4

Goal: Create function that will list all formulas from a selected range as text in a vertical array.

Date	Quantity Purchased w_i	Price per Unit x_i	Calcuate % Frequency		Formulas in Model:
1/2/2022	108	$24.30	10.59%		=LAMBDA(Reference,
2/15/2022	72	$26.45	7.06%		IFERROR(
3/28/2022	120	$26.45	11.76%		TOCOL(
4/27/2022	108	$17.42	10.59%		ADDRESS(
5/19/2022	36	$26.02	3.53%		ROW(Reference),
6/19/2022	96	$21.72	9.41%		COLUMN(Reference),
7/22/2022	36	$20.64	3.53%		4
9/4/2022	120	$25.37	11.76%		)&": "&FORMULATEXT(Reference),
9/26/2022	36	$23.44	3.53%		2),
11/3/2022	144	$21.72	14.12%		""))
12/3/2022	144	$20.21	14.12%		(B6:E27)
Total	1,020				
Ave.	93				Formulas in Model:
Standard Dev.	40				E6: =C6:C16/C17
					C17: =SUM(C6:C16)
Total Units		1020			C18: =ROUND(AVERAGE(C6:C16),0)
Total Inventory Value		$23,275.20			C19: =ROUND(STDEV.P(C6:C16),0)
Weighted Average		$22.82			D21: =SUM(C6:C16)
Units Left on Shelf in 2021		155			D22: =SUMPRODUCT(C6:C16,D6:D16)
End Inventory		$3,536.92			D23: =D22/C17
COGS		$19,738.28			D25: =D24*D23
Check Total		$23,275.20			D26: =D22-D25
					D27: =SUM(D25:D26)

G19: =ShowFormulas(B5:E27)

- **Why Do We Need LAMBDA Helper Functions?**
  - ✓ If the new LAMBDA is so amazing at creating custom functions, why do we need helper functions for LAMBDA? The reason is because there are some tasks that Built-in Excel Functions and LAMBDA Designed Custom Functions have a hard time doing. Here are two examples of tasks that functions have a hard time doing:
    1. Without matrix algebra, functions cannot spill an aggregate row total down a column. But by using LAMBDA inside the LAMBDA Helper Function BYROW, you can do exactly that with this formula: =BYROW(B11:E15,LAMBDA(r,SUM(r))), where the range B11:E15 contains rows of numbers. You can also use an Eta-Lambda to replace some aggregate functions, like SUM, to create a simpler formula: =BYROW(B11:E15, SUM)
    2. Functions cannot spill an account balance formula that must always refer to the previous cell's balance. But by using LAMBDA inside the LAMBDA Helper Function SCAN, you can do exactly that with this formula: =SCAN(M17,L18:L22,LAMBDA(i,a,i+a)), where the cell M17 contains the starting balance and the range L18:L22 contains the array of change values.
  - ✓ You can use LAMBDA inside the LAMBDA Helper Functions directly in the worksheet or in Defined Name Re-usable LAMBDA Custom Functions.

- **These are the nine LAMBDA Helper Functions**
  1. BYCOL (array, function)
     - The BYCOL function iterates a function defined by the LAMBDA function over a specified array, making a function calculation for each column and returning an array of the results. This function is good for spilling an aggregate calculation for each column in a specified array.
  2. BYROW(array, function)
     - The BYROW function iterates a function defined by the LAMBDA function over a specified array, making a function calculation for each row and returning an array of the results. This function is good for spilling an aggregate calculation for each row in a specified array.
  3. MAKEARRAY(rows, cols, function)
     - The MAKEARRAY function returns a calculated array of a specified row and column size, by applying a function defined by the LAMBDA function.
  4. MAP (array1, [array2], [array_n]… function)
     - The MAP function Iterates a function defined by the LAMBDA function over a specified set of arrays (1 or more), making a function calculation for each cell in the corresponding arrays and then returning an array of the results. If you are iterating over a single row or single column, then it is safer to use the MAP function rather than the BYROW or BYCOL function, respectively. When the LAMBDA function expands as it iterates, BYCOL and BYROW may give you the wrong answer, whereas the MAP function will not. For example, a formula like: =BYROW(SEQUENCE(3),LAMBDA(r,CONCAT(SEQUENCE(,r)))) will yield the text array: {1;1;1}, whereas a formula like: =MAP(SEQUENCE(3),LAMBDA(x,CONCAT(SEQUENCE(,x)))) will yield the text array: {1;12;123}.
  5. SCAN ([starting_value], array, function)
     - The SCAN function scans an array by applying a LAMBDA to each value and returns an array that has each intermediate value. For example, it can create a spilled cumulative total formula. This function is good for spilling a cumulative total or an account balance formula, where the spilled account balance formula must always refer to the previous cell's balance (a relative cell reference that always looks "one cell above").

6. REDUCE([starting _value], array, function)
    - The REDUCE function reduces an array to an accumulated value by applying a LAMBDA to each value and returning the last value in the accumulator array. This is similar to the SCAN function, but only returns the last value of a parallel SCAN spilled array.
7. ISOMITTED(LAMBDA_parameter_name)
    - The ISOMITTED function checks whether the value in a LAMBDA formula input argument is missing and returns TRUE or FALSE.
8. GROUPBY(row_fields, values, function, [field_headers], [total_depth], [sort_order], [filter_array])
    - Using a formula, the GROUPBY function creates PivotTable-like summary reports with conditional calculations based on row area and filter area conditions. This function allows you to group, aggregate, sort, and filter data based on the fields you specify.
9. PIVOTBY(row_fields, col_fields, values, function, [field_headers], [row_total_depth], [row_sort_order], [col_total_depth], [col_sort_order], [filter_array])
    - Using a formula, it creates PivotTable-like summary reports with conditional calculations based on row area, column area and filter area conditions. This function allows you to group, aggregate, sort, and filter data based on the fields you specify.

- Note about function argument and Eta-Lambdas:
    - In the **function** argument of the LAMBDA Helper Functions, you can use the LAMBDA function to define the calculation, such as this formula to calculate a running balance based on a value to the left and a previous balance from above:
      =SCAN(E6,D7:D11,LAMBDA(i,a,a+i))
    - **Eta-Lambdas** are defined as names of aggregate functions that can substitute for a LAMBDA function definition when the LAMBDA is iterating and making one of the following 16 aggregate calculations:
        1) SUM
        2) PERCENT OF
        3) AVERAGE
        4) MEDIAN
        5) COUNT
        6) COUNTA
        7) MAX
        8) MIN
        9) PRODUCT
        10) ARRAYTOTEXT
        11) CONCAT
        12) STDEV.S
        13) STDEV.P
        14) VAR.S
        15) VAR.P
        16) MODE.SNGL
    - For example, when spilling a row total for each row in an array, rather than using:
      =BYROW(B9:E13, LAMBDA(r, SUM(r))), you can simply use**: =BYROW(B9:E13, SUM)**
    - For a second example, when creating a summary report, rather than using:
      =GROUPBY(Region,Sales,LAMBDA(r,SUM(r)),3,,-2), you can simply use:
      =GROUPBY(Region,Sales,SUM,3,,-2)

- Examples of BYROW and BYCOL LAMBDA Helper Functions to spill aggregate calculations

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		LAMBDA #5				LAMBDA Helper Function: BYROW						
3		Goal: Spill an aggregate row total for an array with five rows of numbers										
4												
5		LAMBDA #6				LAMBDA Helper Function: BYCOL						
6		Goal: Spill an aggregate column total for an array with five columns of numbers										
7												
8		Jan	Feb	Mar	Apr	Spilled Row Total	With Eta-Lambda:					
9		500	600	600	3,900	5,600	5,600		F9: =BYROW(B9:E13, LAMBDA(r, SUM(r)))			
10		1,000	1,800	2,800	2,700	8,300	8,300		G9: =BYROW(B9:E13,SUM)			
11		2,500	6,250	4,000	3,700	16,450	16,450					
12		250	275	2,100	2,050	4,675	4,675					
13		100	200	750	1,550	2,600	2,600					
14	Total	4,350	9,125	10,250	13,900	37,625						
15	Eta-Lambda:	4,350	9,125	10,250	13,900	37,625						
16		B14: =BYCOL(B9:F13, LAMBDA(c, SUM(c)))										
17		B15: =BYCOL(B9:F13,SUM)										
18												
19		LAMBDA #7				LAMBDA Helper Function: BYROW						
20		Goal: Spill an aggregate row total for an array with five rows of numbers										
21												
22		Jan	Feb	Mar	Apr	Spilled Row Ave.	With Eta-Lambda:					
23		500	600	600	3,900	1,400	1,400		F23: =BYROW(B23:E27, LAMBDA(r, AVERAGE(r)))			
24		1,000	1,800	2,800	2,700	2,075	2,075		G23: =BYROW(B23:E27,AVERAGE)			
25		2,500	6,250	4,000	3,700	4,113	4,113					
26		250	275	2,100	2,050	1,169	1,169					
27		100	200	750	1,550	650	650					

- Examples of MAP LAMBDA Helper Function

	A	B	C	D	E	F	G	H	I	J	K	L
2	LAMBDA #8		LAMBDA Helper: MAP									
4	Task: Students are eligible for scholarship if:											
5	They have completed 45 or more credits AND have GPA more than 2.5.											
6	Goal 1: Create worksheet logical formula that shows if they are eligible for the scholarship.											
8							Formulas:					
9							G14: =AND(E14>=C10,F14>C11)					
10	Credit Hurdle:		45				H14: =(E14:E23>=C10)*(F14:F23>C11)					
11	GPA Hurdle:		2.5				I14: =MAP(E14:E23,F14:F23,LAMBDA(c,g,AND(c>=C10,g>C11)))					
13	Student		Start Date	Major	Credits	GPA	Eligible?	Eligible?	Eligible?			
14	Carey, Zada		9/29/2020	Business	45	1.7	FALSE	0	FALSE			
15	Emmons, Christi		7/14/2018	Accounting	135	2.3	FALSE	0	FALSE			
16	Lear, Vania		9/3/2020	Chemistry	45	3	TRUE	1	TRUE			
17	Meador, Corazon		11/21/2019	Accounting	90	3.1	TRUE	1	TRUE			
18	Mohamed, Abdi		1/28/2021	Business	23	1.6	FALSE	0	FALSE			
19	Nga, Luong		7/7/2020	Physics	45	2.4	FALSE	0	FALSE			
20	Robinson, Chantel		4/12/2020	History	70	4	TRUE	1	TRUE			
21	Rouse, Sioux		6/30/2020	Chemistry	40	2.4	FALSE	0	FALSE			
22	Simone, Alanna		8/2/2019	Physics	60	3.5	TRUE	1	TRUE			
23	Thornburg, Tyrone		12/27/2019	Sociology	75	3.9	TRUE	1	TRUE			

- Use MAP on Single Column or Single Row rather than BYROW or BYCOL

LAMBDA #8.5 — LAMBDA Helper: **BYROW or MAP**

Goal: Spill Total Commission Formula

SalesRep	Quad	Carlota	Aspen	Yanaki
Chin	0.075	0.033	0.012	0.036
Joe	0.074	0.06	0.074	0.07
Sioux	0.077	0.041	0.045	0.056
Tina	0.027	0.087	0.066	0.037

SalesRep	Product	Sales		SR	Total Commission
Sioux	Quad	55		Tina	20.829
Tina	Quad	237		Sioux	94.227
Tina	Aspen	0		Joe	18.848
Chin	Yanaki	0		Chin	72.126
Sioux	Carlota	59			
Chin	Carlota	2		SR	Total Commission
Chin	Aspen	143		Tina	20.829
Sioux	Yanaki	0		Sioux	94.227
Sioux	Yanaki	164		Joe	18.848
Chin	Carlota	83		Chin	72.126
Sioux	Aspen	291			
Chin	Carlota	71			
Sioux	Quad	343			
Sioux	Quad	0		No	BYROW:		MAP:
Tina	Carlota	26		1	1		1
Sioux	Carlota	24		2	1		12
Chin	Quad	27		3	1		123
Joe	Quad	20					
Chin	Quad	11		BYCOL:	1	1	1
Chin	Quad	95		MAP:	1	12	123
Chin	Carlota	72					
Chin	Aspen	0					
Chin	Aspen	82					

G12: =BYROW(F12:F15,LAMBDA(r,SUM(SUMIFS(D12:D67,B12:B67,r,C12:C67,C5:F5)*XLOOKUP(r,B6:B9,C6:F9))))

G18: =MAP(F18:F21,LAMBDA(r,SUM(SUMIFS(D12:D67,B12:B67,r,C12:C67,C5:F5)*XLOOKUP(r,B6:B9,C6:F9))))

F26: =SEQUENCE(3)

G26: =BYROW(SEQUENCE(3),LAMBDA(r,CONCAT(SEQUENCE(,r))))

H26: =MAP(SEQUENCE(3),LAMBDA(x,CONCAT(SEQUENCE(,x))))

G30: =BYCOL(SEQUENCE(,3),LAMBDA(r,CONCAT(SEQUENCE(,r))))

G31: =MAP(SEQUENCE(,3),LAMBDA(x,CONCAT(SEQUENCE(,x))))

ExcelLambda:
BYROW has a problem with single cell rows when the lambda helper function expands, MAP has not. (for 1D arrays) 2. MAP feeds itself cell by cell, BYROW row by row, so for 2D arrays are totally different. Only for 1D arrays MAP is more versatile than BYROW, because it is.

- Examples of SCAN LAMBDA Helper Function

This example shows how to use SACN to iterate across an array and deliver each intermediate amount, like in a running total. SCAN also is the function method to simulate a Relative Cell Reference in a spilled array.

LAMBDA #9 — LAMBDA Helper Function: **SCAN**

Goal: Create a dynamic spilled array formula for a running checkbook balance

Date	Number	Change	Balance	LAMBDA	Eta-LAMBDA	
Bal. For.			$3,502.25			
12/24/2021	2156	34.98	$3,537.23	$3,537.23	$3,537.23	E7: =D7+E6
12/25/2021	2157	-2504.3	$1,032.91	$1,032.91	$1,032.91	F7: =SCAN(E6,D7:D11,LAMBDA(i,a,a+i))
12/26/2021	2158	2500.3	$3,533.21	$3,533.21	$3,533.21	G7: =SCAN(E6,D7:D11,SUM)
12/27/2021	2159	-1225.9	$2,307.32	$2,307.32	$2,307.32	
12/28/2021	2160	7.75	$2,315.07	$2,315.07	$2,315.07	

LAMBDA #10 — LAMBDA Helper Function: **SCAN**

Goal: Create a dynamic spilled array formula that can calculate a running total

SalesRep	Sales ($)		SalesRep	Sales ($)	Running Total	Eta-Lambda	
Shihara	102.38		Chantel	984.38	984.38	984.38	G18: =SCAN(0,F18#,LAMBDA(i,a,a+i))
Chantel	194.74		Joe	1,099.39	2,083.77	2,083.77	H18: =SCAN(0,F18#,SUM)
Joe	328.07		Shihara	641.13	2,724.90	2,724.90	
Sioux	243.67		Sioux	494.07	3,218.97	3,218.97	

- Examples of REDUCE LAMBDA Helper Function

This example shows how to use REDUCE to iterate across an array and deliver the final result after all iterations.

	Alphanumeric	REDUCE & LAMBDA	2nd LAMBDA	New Function
6	Quad63544398106	=LAMBDA(text,		Quad
7	8803456464 Sioux Radcool	REDUCE(		Sioux Radcool
8	014369Carlota	text,		Carlota
9	c1o43ol	SEQUENCE(10,,0),		cool
10	76441ART/Lin01dam56WGT	LAMBDA(i,a, SUBSTITUTE(i, a, "")))))(B6:B15)		ART/LindamWGT
11	*a1s2f{aa}3	LAMBDA(parameter_or_calculation, [parameter_or_calculation], …)		*asf{aa}
12	ex1ce3l4is2fu4n	excelisfun	excelisfun	excelisfun
13	10098ApplesWA98106	ApplesWA	ApplesWA	ApplesWA
14	9hhyst56	hhyst	hhyst	hhyst
15	Data1670ΩImport	DataΩImport	DataΩImport	DataΩImport

LAMBDA # 11

Goal: Remove numbers from a text string   LAMBDA Helper Function: REDUCE

Two more examples of REDUCE on next page:

LAMBDA #12

Goal: Add space before Capital Letters

	Add Space Before Cap	Add Space Before Cap		New Function
	HighlineCommunityCollege	=LAMBDA(Text,		Highline Community College
	OurWebSiteName	TRIM(REDUCE(		Our Web Site Name
	CarlotaBoomerang	Text,		Carlota Boomerang
	ExcelIsFun	CHAR(SEQUENCE(26, , 65)),		Excel Is Fun
	ReduceFunctionIsFun	LAMBDA(i,a, SUBSTITUTE(i, a, " "&a)))))(B21:B25)		Reduce Function Is Fun
		LAMBDA(parameter_or_calculation, [parameter_or_calculation], ...)		

LAMBDA # 13

Goal: remove specified characters   **LAMBDA Helper Function: REDUCE**

Specify:

ArtQuadDataΩ

Alphanumeric	REDUCE & LAMBDA			REDUCE & LAMBDA
Quad63544398106	=LAMBDA(Text,CharactersToRemove,			63544398106
8803456464 Sioux Radcool	TRIM(REDUCE(Text,			8803456464 Siox Rcool
014369Carlota	MID(CharactersToRemove, SEQUENCE(LEN(CharactersToRemove)), 1),			014369Clo
c1o43ol	LAMBDA(i,a, SUBSTITUTE(i, a, "")))))(B33:B42,C30)			c1o43ol
76441ART/Lin01dam56WGT	76441RT/Lin01m56WGT			76441RT/Lin01m56WGT
*a1s2f{aa}3	*1s2f{}3			*1s2f{}3
ex1ce3l4is2fu4n	ex1ce3l4is2f4n			ex1ce3l4is2f4n
10098ApplesWA98106	10098pplesW98106			10098pplesW98106
9hhyst56	9hhys56			9hhys56
Data1670ΩImport	1670Impo			1670Impo

- Example of Recursion in the LAMBDA function

	A	B	C	D	E	F
1						
2		**Recursion**				
3		Recursion means that a function can call itself and is allowed to iterate over a value until the task is achieved.				
4		In the Excel worksheet, if know the number of iteration, you can use the RECDUCE function rather than recursion				
5		to call the function for each iteration.				

=LAMBDA(x, y, ..., <= set function argument parameters
      IF(
         logical_test = create logical test that will stop function iteration when task is complete,
         value_if_true = final result after all iterations,
         value_if_false = LAMBDA to iterate))

The Recursive LAMBDA we will create:

=LAMBDA(Text, CharToRemove,
IF(CharToRemove="",
TRIM(Text),
RemoveChar(
SUBSTITUTE(Text, LEFT(CharToRemove), ""),
RIGHT(CharToRemove, LEN(CharToRemove)-1))))

LAMBDA # 14

Goal: Using recursion, create function that let's you specify what characters to remove

Alphanumeric	LAMBDA and Recursion	REDUCE & LAMBDA		New Function
Quad63544398106	=LAMBDA(Text,CharToRemove,			Quad
8803456464 Sioux Radcool	IF(CharToRemove="",			Sioux Radcool
014369Carlota	TRIM(Text),			Carlota
c1o43ol	RemoveChar(			cool
	SUBSTITUTE(Text, LEFT(CharToRemove), ""),			
	RIGHT(CharToRemove, LEN(CharToRemove)-1)))(B25:B28,"0123456789")			

Example of MAKRARRAY LAMBDA Helper Function

LAMBDA # 15

Goal: Look at MAKEARRAY   LAMBDA Helper Function: **MAKEARRAY**

1	2	3	4	5	6		2	3	4	5	6	7
2	4	6	8	10	12		3	4	5	6	7	8
3	6	9	12	15	18		4	5	6	7	8	9
4	8	12	16	20	24		5	6	7	8	9	10
5	10	15	20	25	30		6	7	8	9	10	11

Carlota	Aspen	Aspen	Aspen	Carlota	Aspen		x	x	x	x	x	x
Carlota	Bellen	Carlota	Carlota	Bellen	Carlota		x	x	x	x	x	x
Carlota	Carlota	Carlota	Bellen	Bellen	Carlota		x	x	x	x	x	x
Aspen	Carlota	Aspen	Bellen	Bellen	Carlota		x	x	x	x	x	x
Carlota	Bellen	Carlota	Aspen	Carlota	Carlota		x	x	x	x	x	x

Bellen	Bellen	Aspen	Carlota	Carlota	Carlota		x	x	x	x	x	x
Aspen	Carlota	Aspen	Bellen	Bellen	Aspen		x	x	x	x	x	x
Bellen	Aspen	Aspen	Bellen	Aspen	Carlota		x	x	x	x	x	x
Carlota	Carlota	Carlota	Aspen	Bellen	Carlota		x	x	x	x	x	x
Aspen	Aspen	Aspen	Carlota	Carlota	Carlota		x	x	x	x	x	x

Formulas:

B5: =MAKEARRAY(5,6,LAMBDA(r,c,r*c))

I5: =MAKEARRAY(5,6,LAMBDA(r,c,r+c))

B11: =MAKEARRAY(5,6,LAMBDA(r,c,INDEX({"Bellen","Aspen","Carlota"},RANDBETWEEN(1,3))))

I11: =MAKEARRAY(5,6,LAMBDA(r,c,"x"))

B17: =INDEX({"Carlota","Bellen","Aspen"},RANDARRAY(5,6,1,3,1))

I17: =INDEX("x",RANDARRAY(5,6,1,1,1))

LET Function

- Define LET function: The LET function allows you to define variables within the function and use those variables to make a calculation. Use LET when you have repeating formula elements or when you want to make complex formulas easier to read. Let arguments are here:

LET(
name1, name_value1,
name2, name_value2, ...
name_n, name_value_n,
calculation)

name argument = variable name.
name_value argument = variable formula element.
Up to 126 variables

calculation argument = final result that is delivered by LET
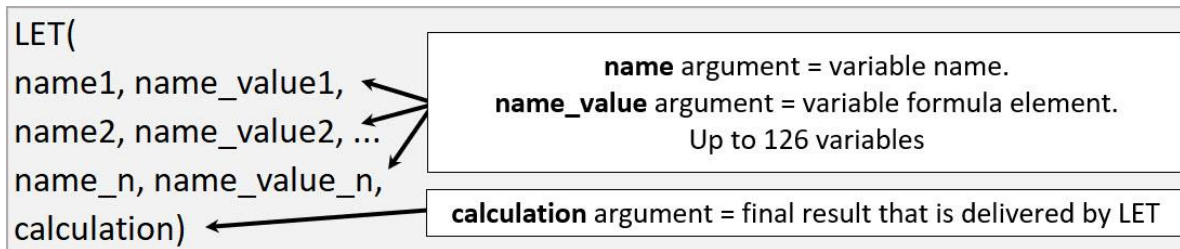
- Advantages of LET:
  - A variable is evaluated a single time, and the result is stored in memory so that it can be used throughout a formula. For formulas with repeating formula elements, this can reduce overall calculation time by preventing duplication of evaluation procedures.
  - Formulas with repeating elements are easy to edit because you have to edit in only one location.
  - Complex formulas can be visually easy to read because each element has a name and can be placed on a different line with the line feed keyboard shortcut, Alt+Enter.
  - You can build single cell report formulas more efficiently with the LET function.

GROUPBY function to make Single Cell Reports

	Date	Region	SalesRep	Product	Sales	COGS		Region						
								Regional Sales Totals with GROUPBY		**LAMBDA version**			**Regional % of Sales Totals**	
3	6/27/2020	West	Shayla	Carlota	352.003	186.56159								
4	8/17/2020	MidWest	Shayla	Yanaki	302.593	139.19278								
5	2/4/2021	East	Shayla	FlyFast	161.851	89.01805		Region		Region			Region	
6	10/18/2021	MidWest	Ahmed	Quad	228.229	136.9374		East	2,827.55	East	2,827.55		East	23.97%
7	4/12/2021	SouthWest	Ahmed	Yanaki	390.618	160.15338		MidWest	1,545.76	MidWest	1,545.76		MidWest	13.10%
8	7/10/2021	MidWest	Chantel	Quad	154.549	66.45607		NorthWest	834.62	NorthWest	834.62		NorthWest	7.08%
9	5/18/2020	SouthWest	Chantel	FlyFast	464.432	218.28304		South	1,311.13	South	1,311.13		South	11.12%
10	9/24/2020	SouthWest	Chantel	Quad	407.913	191.71911		SouthWest	1,262.96	SouthWest	1,262.96		SouthWest	10.71%
11	11/25/2021	MidWest	Shayla	SpitFire	321.895	180.2612		West	4,013.60	West	4,013.60		West	34.03%
12	9/15/2021	MidWest	Chantel	FlyFast	159.275	73.2665		Total	11,795.62	Total	11,795.62		Total	100.00%
13	11/7/2021	West	Chantel	Sunshine	1413.342	706.671								
14	9/12/2021	MidWest	Ahmed	Yanaki	379.218	189.609		**Regional Sales Totals with PIVOTBY**		**Regional Ave. Sales Report with Calculation Labels**				
15	1/15/2021	South	Shayla	Yanaki	473.771	279.52489								
16	2/1/2021	South	Shayla	Carlota	345.916	200.63128		Region		Region	Total Sales		Region	Ave. Sales
17	8/3/2021	East	Chantel	Quad	497.488	248.744		East	2,827.55	East	353.44		East	353.44
18	11/20/2020	NorthWest	Ahmed	Yanaki	467.306	242.99912		MidWest	1,545.76	MidWest	257.63		MidWest	257.63
19	12/10/2020	West	Shayla	SpitFire	761.005	319.6221		NorthWest	834.62	NorthWest	417.31		NorthWest	417.31
20	8/22/2020	East	Shayla	Carlota	393.039	216.17145		South	1,311.13	South	437.04		South	437.04
21	11/25/2021	West	Shayla	Quad	127.241	69.98255		SouthWest	1,262.96	SouthWest	420.99		SouthWest	420.99
22	7/14/2021	West	Gigi	FlyFast	465.011	195.30462		West	4,013.60	West	573.37		West	573.37
23	4/22/2021	East	Sioux	Quad	145.929	84.63882		Total	11,795.62	Total	406.75		Total	406.75
24	3/22/2021	NorthWest	Ahmed	Quad	367.31	202.0205								
25	1/15/2021	East	Sioux	Sunshine	316.73	183.7034								
26	2/8/2021	South	Sioux	Quad	491.44	280.1208		**Formulas:**						
27	10/20/2020	West	Chantel	Quad	424.324	229.13496		I5: =GROUPBY(C2:C31,F2:F31,SUM,3)						
28	4/29/2021	West	Shayla	Quad	470.674	221.21678		L5: =GROUPBY(C2:C31,F2:F31,LAMBDA(r,SUM(r)),3)						
29	6/30/2020	East	Ahmed	FlyFast	453.922	222.42178		O5: =GROUPBY(C2:C31,F2:F31,PERCENTOF,3)						
30	2/22/2021	East	Sioux	Carlota	412.46	185.607		I16: =PIVOTBY(C2:C31,,F2:F31,SUM,3)						
31	10/16/2021	East	Sioux	Quad	446.135	178.454		O16: =VSTACK({"Region","Ave. Sales"},GROUPBY(C3:C31,F3:F31,AVERAGE,0))						
32								L17: =GROUPBY(C2:C31,F2:F31,AVERAGE,1)						

GROUPBY function to make Single Cell Reports

Date	Region	SalesRep	Product	Sales	COGS
6/27/2020	West	Shayla	Carlota	352.003	186.56159
8/17/2020	MidWest	Shayla	Yanaki	302.593	139.19278
2/4/2021	East	Shayla	FlyFast	161.851	89.01805
10/18/2021	MidWest	Ahmed	Quad	228.229	136.9374
4/12/2021	SouthWest	Ahmed	Yanaki	390.618	160.15338
7/10/2021	MidWest	Chantel	Quad	154.549	66.45607
5/18/2020	SouthWest	Chantel	FlyFast	464.432	218.28304
9/24/2020	SouthWest	Chantel	Quad	407.913	191.71911
11/25/2021	MidWest	Shayla	SpitFire	321.895	180.2612
9/15/2021	MidWest	Chantel	FlyFast	159.275	73.2665
11/7/2021	West	Chantel	Sunshine	1413.342	706.671
9/12/2021	MidWest	Ahmed	Yanaki	379.218	189.609
1/15/2021	South	Shayla	Yanaki	473.771	279.52489
2/1/2021	South	Shayla	Carlota	345.916	200.63128
8/3/2021	East	Chantel	Quad	497.488	248.744
11/20/2020	NorthWest	Ahmed	Yanaki	467.306	242.99912
12/10/2020	West	Shayla	SpitFire	761.005	319.6221
8/22/2020	East	Shayla	Carlota	393.039	216.17145
11/25/2021	West	Shayla	Quad	127.241	69.98255
7/14/2021	West	Gigi	FlyFast	465.011	195.30462
4/22/2021	East	Sioux	Quad	145.929	84.63882
3/22/2021	NorthWest	Ahmed	Quad	367.31	202.0205
1/15/2021	East	Sioux	Sunshine	316.73	183.7034
2/8/2021	South	Sioux	Quad	491.44	280.1208
10/20/2020	West	Chantel	Quad	424.324	229.13496
4/29/2021	West	Shayla	Quad	470.674	221.21678
6/30/2020	East	Ahmed	FlyFast	453.922	222.42178
2/22/2021	East	Sioux	Carlota	412.46	185.607
10/16/2021	East	Sioux	Quad	446.135	178.454

2 Fields, 2 Calculations, Add Headers, Subtotals, Conditional Formatting

2 Different Calculations

Region	SalesRep	Ave. Sales	Ave. Cost
East	Ahmed	453.92	222.42
East	Chantel	497.49	248.74
East	Shayla	277.45	152.59
East	Sioux	330.31	158.10
East		353.44	176.09
MidWest	**Ahmed**	**303.72**	**163.27**
MidWest	Chantel	156.91	69.86
MidWest	Shayla	312.24	159.73
MidWest		257.63	130.95
NorthWest	**Ahmed**	**417.31**	**222.51**
NorthWest		417.31	222.51
South	**Shayla**	**409.84**	**240.08**
South	Sioux	491.44	280.12
South		437.04	253.43
SouthWest	**Ahmed**	**390.62**	**160.15**
SouthWest	Chantel	436.17	205.00
SouthWest		420.99	190.05
West	**Chantel**	**918.83**	**467.90**
West	Gigi	465.01	195.30
West	Shayla	427.73	199.35
West		573.37	275.50
Grand Total		406.75	203.39

Region	Sales	% Sales
East	2,827.55	23.97%
MidWest	1,545.76	13.10%
NorthWest	834.62	7.08%
South	1,311.13	11.12%
SouthWest	1,262.96	10.71%
West	4,013.60	34.03%
Total	11,795.62	100.00%

Formulas:

R5: =VSTACK({"Region","SalesRep","Ave. Sales","Ave. Cost"},GROUPBY(C2:D31,F2:G31,AVERAGE,1,2))

W6: =LET(g,GROUPBY(C3:C31,F3:F31,SUM,0),s,DROP(g,,1),t,TAKE(s,-1),HSTACK(g,s/t))

PIVOTBY function to make Single Cell Reports

	Date	Region	SalesRep	Product	Sales	COGS
3	6/27/2020	West	Gigi	Yanaki	285.28	134.0816
4	8/17/2020	MidWest	Chantel	Yanaki	505.06	277.783
5	2/4/2021	SouthWe:	Chantel	Quad	147.34	66.303
6	10/18/2021	West	Chantel	Yanaki	275.44	126.7024
7	4/12/2021	MidWest	Gigi	Yanaki	372.53	149.012
8	7/10/2021	SouthWe:	Chantel	Quad	403.11	201.555
9	5/18/2020	West	Chantel	Carlota	468.68	215.5928
10	9/24/2020	MidWest	Tim	Carlota	222.15	95.5245
11	11/25/2021	SouthWe:	Tim	Carlota	584.87	280.7376
12	9/15/2021	West	Gigi	Yanaki	539.54	323.724
13	11/7/2021	MidWest	Sioux	Yanaki	161.25	69.3375
14	/12/2021	SouthWe:	Chantel	Yanaki	427.64	256.584
15	/15/2021	West	Gigi	Carlota	464.92	251.0568
16	2/1/2021	MidWest	Tim	Yanaki	688.32	412.992
17	8/3/2021	SouthWe:	Tim	Quad	266.76	109.3716
18	11/20/2020	West	Sioux	Yanaki	343.58	151.1752
19	12/10/2020	MidWest	Chantel	Quad	455.52	259.6464
20	8/22/2020	SouthWe:	Chantel	Quad	660.68	264.272
21	11/25/2021	West	Chantel	Yanaki	123.27	72.7293
22	7/14/2021	MidWest	Tim	Yanaki	713.7	385.398
23	4/22/2021	SouthWe:	Gigi	Yanaki	788.22	370.4634
24	3/22/2021	West	Gigi	Carlota	186.77	100.8558
25	1/15/2021	MidWest	Tim	Yanaki	266.91	112.1022
26	2/8/2021	SouthWe:	Tim	Quad	484.77	276.3189
27	10/20/2020	West	Gigi	Carlota	479.97	254.3841
28	4/29/2021	MidWest	Chantel	Carlota	680.97	360.9141
29	6/30/2020	SouthWe:	Chantel	Yanaki	678.22	271.288
30	2/22/2021	West	Gigi	Quad	378.16	215.5512
31	10/16/2021	MidWest	Gigi	Yanaki	788.93	449.6901
32	4/22/2021	SouthWe:	Tim	Quad	382.45	172.1025
33	3/22/2021	West	Tim	Yanaki	662.52	351.1356
34	1/15/2021	MidWest	Tim	Quad	324.41	171.9373
35	2/8/2021	SouthWe:	Sioux	Quad	155.97	90.4626
36	2/1/2021	West	Chantel	Carlota	630.78	365.8524
37	8/3/2021	MidWest	Gigi	Quad	777.56	404.3312
38	11/20/2020	SouthWe:	Sioux	Yanaki	697.78	390.7568
39	12/10/2020	West	Chantel	Yanaki	579.79	243.5118
40	9/15/2021	MidWest	Gigi	Carlota	797.67	342.9981
41	11/7/2021	SouthWe:	Chantel	Yanaki	327.91	141.0013
42	10/16/2021	West	Gigi	Carlota	267.79	131.2171
43	1/15/2021	MidWest	Tim	Carlota	267.6	160.56
44	2/8/2021	SouthWe:	Chantel	Carlota	799.65	335.853
45	2/1/2021	West	Sioux	Yanaki	258.83	139.7682
46	8/3/2021	MidWest	Tim	Quad	124.5	58.515
47	11/20/2020	SouthWe:	Chantel	Quad	348.83	156.9735
48	12/10/2020	West	Gigi	Quad	537.08	252.4276

Sum of Sales Report Where Sales >250:

	Product			
Region	Yanaki	Quad	Carlota	Total
MidWest	3,335.45	4,121.58	2,434.56	9,891.59
SouthWest	5,840.96	3,516.14	4,082.40	13,439.50
West	2,944.98	5,323.38	5,408.88	13,677.24
Total	12,121.39	12,961.10	11,925.84	37,008.33

		Increase	10.00%

Sales 10.00% bigger?

	Product			
Region	Carlota	Quad	Yanaki	Total
MidWest	3,333.28	4,670.69	3,846.37	11,850.33
SouthWest	4,490.64	4,201.40	6,913.79	15,605.82
West	6,656.74	5,855.72	3,375.08	15,887.53
Total	14,480.65	14,727.80	14,135.23	43,343.69

	Product			
Region	Carlota	Quad	Yanaki	Total
MidWest	3333.275	4670.688	3846.37	11850.333
SouthWest	4490.64	4201.395	6913.786	15605.821
West	6656.738	5855.718	3375.075	15887.531
Total	14480.653	14727.801	14135.231	43343.685

			Filter Items
Row	Region		Chantel
Column	Product		Gigi
Calculation Column	Sales		Sioux
Filter Column	SalesRep		Tim
Filter Condition	Gigi		
Filter Condition	SUM		

SUM of Sales	Product			
Region	Carlota	Quad	Yanaki	Total
West	1,399.45	915.24	824.82	3,139.51
MidWest	797.67	777.56	1,161.46	2,736.69
SouthWest	800.58	969.54	788.22	2,558.34
Total	2,997.70	2,662.34	2,774.50	8,434.54

I4: =PIVOTBY(C2:C91,E2:E91,F2:F91,SUM,3,,,,-1,F3:F91>250)

I13: ="Sales "&TEXT(L12,"0.00%")&" bigger?"

I14: =PIVOTBY(C2:C91,E2:E91,F2:F91*(1+L12),SUM,3)

I21: =PIVOTBY(C2:C91,E2:E91,F2:F91,LAMBDA(c,SUM(c)*(1+L12)),3)

L30: =SORT(UNIQUE(XLOOKUP(J32,C2:E2,C3:E91)))

Custom LAMBDA & LET Function to Calculate Return and Standard Deviation for Portfolio of Stocks

LAMBDA #19

Goal 1: Create function to calculate the estimated returns and standard deviation for a portfolio of stocks.

	Weight of Stock in Portfolio:			60.00%	40.00%
	Probability of Economic State		Stock A Estimated Return	Stock B Estimated Return	
Bad	35.00%		2.50%	-7.50%	
OK	25.00%		8.50%	12.50%	
Good	40.00%		14.50%	22.00%	
			Expected Portfolio Return		9.00%
			Standard Deviation of Portfolio Returns		8.23%
			Expected Portfolio Return		9.00%
			Standard Deviation of Portfolio Returns		8.23%

```
=LAMBDA(RowHeaderEconStateProb,ColHeaderStockWeights,TableEstStockReturns,
LET(
rh,RowHeaderEconStateProb,
ch,ColHeaderStockWeights,
ts,TableEstStockReturns,
erp,SUM(rh*ch*ts),
sdp,SQRT(SUM(BYROW(ts*ch,LAMBDA(r,(SUM(r)-erp)^2))*rh)),

VSTACK(
HSTACK("Expected Portfolio Return",erp),
HSTACK("Standard Deviation of Portfolio Returns",sdp))))(C7:C9,D5:E5,D7:E9)
```

Custom LAMBDA & LET Function to Enhance the built-in Statistics Function LINEST

	A	B	C	D	E	F	G	H	I	J	K
1											
2		**LAMBDA #20**									
3		**Goal 1:** Add labels to statistical output from LINEST Array Function.									
4											
5		Hours Studied (X)	Test Score (Y)		=LAMBDA(Yvalues,Xvalues,						
6		41	82		LET(						
7		21	21		labels,{"Slope m";"Standard Error m";"R^2";"F";"SS Regression";"Intercept b";						
8		8	24		"Standard Error b";"Standard Error y";"df";"SS Residual"},						
9		49	100		stats,TOCOL(LINEST(Yvalues,Xvalues,1,1),,1),HSTACK(labels,stats)))(C6:C37,B6:B37)						
10		17	51		Intercept b	11.80466					
11		59	59		Standard Error b	7.4719					
12		17	17		Standard Error y	19.93489					
13		40	80		df	30					
14		24	24		SS Residual	11921.99					
15		32	64								
16		57	100		Slope m	1.333921		E16: =XYRegression10Stats(C6:C37,B6:B37)			
17		57	57		Standard Error m	0.211897					
18		35	70		R^2	0.569144					
19		52	52		F	39.6288					
20		12	36		SS Regression	15748.48					
21		33	66		Intercept b	11.80466					
22		12	12		Standard Error b	7.4719					
23		43	100		Standard Error y	19.93489					
24		30	30		df	30					
25		9	18		SS Residual	11921.99					
26		10	10								