

Excel

For

DATA ANALYSTS

Day-7

(Top 10 Excel Functions)



BY



MOHAMMED UMAER

DATA ANALYST

1. Excel VLOOKUP Function

VLOOKUP Function

The **VLOOKUP** function is a premade function in Excel, which allows searches across columns.

It is typed =VLOOKUP and has the following parts:

=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])

Note: The column which holds the data used to lookup must always be to the left.

Note: The different parts of the function are separated by a symbol, like comma, or semicolon ;

Lookup_value: Select the cell where search values will be entered.

Table_array: The table range, including all cells in the table.

Col_index_num: The data which is being looked up. The input is the number of the column, counted from the left:

	A	B	C	D	E
1	1	2	3	4	5
2	#	Name	Type 1	Type 2	Total
3	1	Bulbasaur	Grass	Poison	318
4	2	Ivysaur	Grass	Poison	405
5	3	Venusaur	Grass	Poison	525
6	4	Charmander	Fire		309
7	5	Charmeleon	Fire		405
8	6	Charizard	Fire	Flying	534
9	7	Squirtle	Water		314
10	8	Wartortle	Water		405
11	9	Blastoise	Water		530
12	10	Caterpie	Bug		195
13	11	Metapod	Bug		205
14	12	Butterfree	Bug	Flying	395
15	13	Weedle	Bug	Poison	195
16	14	Kakuna	Bug	Poison	205
17	15	Beedrill	Bug	Poison	395
18	16	Pidgey	Normal	Flying	251
19	17	Pidgeotto	Normal	Flying	349
20	18	Pidgeot	Normal	Flying	479
21	19	Rattata	Normal		253
22	20	Raticate	Normal		413

Range_lookup: TRUE if numbers (1) or FALSE if text (0).

Note: Both 1 / 0 and True / False can be used in **Range_lookup**.

How to use the **VLOOKUP** function.

1. Select a cell (H4)
2. Type =VLOOKUP
3. Double click the VLOOKUP command
4. Select the cell where search value will be entered (H3)
5. Type (,)
6. Mark table range (A2:E21)
7. Type (,)
8. Type the number of the column, counted from the left (2)
9. Type True (1) or False (0) (1)
10. Hit enter
11. Enter a value in the cell selected for the Lookup_value H3(7)


Let's have a look at an example!

Use the **VLOOKUP** function to find the Pokemon names based on their **ID#**:

	A	B	C	D	E	F	G	H
1	ID#	Name	Type 1	Type 2	Total			
2	1	Bulbasaur	Grass	Poison	318			
3	2	Ivysaur	Grass	Poison	405		Search ID#	
4	3	Venusaur	Grass	Poison	525		Name	=
5	4	Charmander	Fire		309			
6	5	Charmeleon	Fire		405			
7	6	Charizard	Fire	Flying	534			
8	7	Squirtle	Water		314			
9	8	Wartortle	Water		405			
10	9	Blastoise	Water		530			
11	10	Caterpie	Bug		195			
12	11	Metapod	Bug		205			
13	12	Butterfree	Bug	Flying	395			
14	13	Weedle	Bug	Poison	195			
15	14	Kakuna	Bug	Poison	205			
16	15	Beedrill	Bug	Poison	395			
17	16	Pidgey	Normal	Flying	251			
18	17	Pidgeotto	Normal	Flying	349			
19	18	Pidgeot	Normal	Flying	479			
20	19	Rattata	Normal		253			
21	20	Raticate	Normal		413			
22								

Excel for Data Analysts

H4 is where the search result is displayed. In this case, the Pokemons names based on their ID#.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID#	Name	Type 1	Type 2	Total									
2	1	Bulbasaur	Grass	Poison	318									
3	2	Ivysaur	Grass	Poison	405		Search ID#							
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP						
5	4	Charmander	Fire		309			 VLOOKUP	Looks for a value in the leftmost column of a table, and then returns a value in the same row from a column you specify. By default, the table must be sorted in an ascending order					
6	5	Charmeleon	Fire		405									
7	6	Charizard	Fire	Flying	534									
8	7	Squirtle	Water		314									
9	8	Wartortle	Water		405									
10	9	Blastoise	Water		530									
11	10	Caterpie	Bug		195									
12	11	Metapod	Bug		205									
13	12	Butterfree	Bug	Flying	395									
14	13	Weedle	Bug	Poison	195									
15	14	Kakuna	Bug	Poison	205									
16	15	Beedrill	Bug	Poison	395									
17	16	Pidgey	Normal	Flying	251									
18	17	Pidgeotto	Normal	Flying	349									
19	18	Pidgeot	Normal	Flying	479									
20	19	Rattata	Normal		253									
21	20	Raticate	Normal		413									
22														

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP(
5	4	Charmander	Fire		309			VLOOKUP (lookup_value; table_array; col_index_num; [range_lookup])				
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

H3 selected as **lookup_value**. This is the cell where the search query is entered. In this case the Pokemons ID#.

Excel for Data Analysts

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP(H3				
5	4	Charmander	Fire		309			VLOOKUP (lookup_value; table_array; col_index_num; [range_lookup])				
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP(H3;A2:E21				
5	4	Charmander	Fire		309			VLOOKUP (lookup_value; table_array; col_index_num; [range_lookup])				
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

The range of the table is marked at **table_array**, in this example A2:E21.

Excel for Data Analysts

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP(H3;A2:E21;2				
5	4	Charmander	Fire		309			VLOOKUP (lookup_value; table_array; col_index_num; [range_lookup])				
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

The number 2 is entered as **col_index_number**. This is the second column from the left and is the data that is being looked up.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	=VLOOKUP(H3;A2:E21;2;1				
5	4	Charmander	Fire		309			VLOOKUP (lookup_value; table_array; col_index_num; [range_lookup])				
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

An illustration for selecting **col_index_number** 2.

Excel for Data Analysts

	A	B	C	D	E
	1	2	3	4	5
1					
2	#	Name	Type 1	Type 2	Total
3	1	Bulbasaur	Grass	Poison	318
4	2	Ivysaur	Grass	Poison	405
5	3	Venusaur	Grass	Poison	525
6	4	Charmander	Fire		309
7	5	Charmeleon	Fire		405
8	6	Charizard	Fire	Flying	534
9	7	Squirtle	Water		314
10	8	Wartortle	Water		405
11	9	Blastoise	Water		530
12	10	Caterpie	Bug		195
13	11	Metapod	Bug		205
14	12	Butterfree	Bug	Flying	395
15	13	Weedle	Bug	Poison	195
16	14	Kakuna	Bug	Poison	205
17	15	Beedrill	Bug	Poison	395
18	16	Pidgey	Normal	Flying	251
19	17	Pidgeotto	Normal	Flying	349
20	18	Pidgeot	Normal	Flying	479
21	19	Rattata	Normal		253
22	20	Raticate	Normal		413

Ok, so next - 1 (**True**) is entered as **range_lookup**. This is because the most left column has numbers only. If it was text, 0 (**False**) would have been used.

	A	B	C	D	E	F	G	H	I	J	K	L
1	ID#	Name	Type 1	Type 2	Total							
2	1	Bulbasaur	Grass	Poison	318							
3	2	Ivysaur	Grass	Poison	405		Search ID#					
4	3	Venusaur	Grass	Poison	525		Name	#N/A				
5	4	Charmander	Fire		309							
6	5	Charmeleon	Fire		405							
7	6	Charizard	Fire	Flying	534							
8	7	Squirtle	Water		314							
9	8	Wartortle	Water		405							
10	9	Blastoise	Water		530							
11	10	Caterpie	Bug		195							
12	11	Metapod	Bug		205							
13	12	Butterfree	Bug	Flying	395							
14	13	Weedle	Bug	Poison	195							
15	14	Kakuna	Bug	Poison	205							
16	15	Beedrill	Bug	Poison	395							
17	16	Pidgey	Normal	Flying	251							
18	17	Pidgeotto	Normal	Flying	349							
19	18	Pidgeot	Normal	Flying	479							
20	19	Rattata	Normal		253							
21	20	Raticate	Normal		413							
22												

Excel for Data Analysts

The function returns the #N/A value. This is because there have not been entered any value to the Search ID# H3.

Let us feed a value to it, type H3(7):

	A	B	C	D	E	F	G	H
1	ID#	Name	Type 1	Type 2	Total			
2	1	Bulbasaur	Grass	Poison	318			
3	2	Ivysaur	Grass	Poison	405		Search ID#	7
4	3	Venusaur	Grass	Poison	525		Name	Squirtle
5	4	Charmander	Fire		309			
6	5	Charmeleon	Fire		405			
7	6	Charizard	Fire	Flying	534			
8	7	Squirtle	Water		314			
9	8	Wartortle	Water		405			
10	9	Blastoise	Water		530			
11	10	Caterpie	Bug		195			
12	11	Metapod	Bug		205			
13	12	Butterfree	Bug	Flying	395			
14	13	Weedle	Bug	Poison	195			
15	14	Kakuna	Bug	Poison	205			
16	15	Beedrill	Bug	Poison	395			
17	16	Pidgey	Normal	Flying	251			
18	17	Pidgeotto	Normal	Flying	349			
19	18	Pidgeot	Normal	Flying	479			
20	19	Rattata	Normal		253			
21	20	Raticate	Normal		413			
22								

Have a look at that! The **VLOOKUP** function has successfully found the Pokemon Squirtle which has the ID# 7.

One more time, type (H3)4:

Excel for Data Analysts

	A	B	C	D	E	F	G	H
1	ID#	Name	Type 1	Type 2	Total			
2	1	Bulbasaur	Grass	Poison	318			
3	2	Ivysaur	Grass	Poison	405		Search ID#	4
4	3	Venusaur	Grass	Poison	525		Name	Charmander
5	4	Charmander	Fire		309			
6	5	Charmeleon	Fire		405			
7	6	Charizard	Fire	Flying	534			
8	7	Squirtle	Water		314			
9	8	Wartortle	Water		405			
10	9	Blastoise	Water		530			
11	10	Caterpie	Bug		195			
12	11	Metapod	Bug		205			
13	12	Butterfree	Bug	Flying	395			
14	13	Weedle	Bug	Poison	195			
15	14	Kakuna	Bug	Poison	205			
16	15	Beedrill	Bug	Poison	395			
17	16	Pidgey	Normal	Flying	251			
18	17	Pidgeotto	Normal	Flying	349			
19	18	Pidgeot	Normal	Flying	479			
20	19	Rattata	Normal		253			
21	20	Raticate	Normal		413			
22								

It still works! The function returned Charmanders name, which has 4 as its ID#. **That's great.**

2. INDEX and MATCH Function in Excel

INDEX-MATCH has become a more popular tool for Excel as it solves the limitation of the VLOOKUP function, and it is easier to use. INDEX-MATCH function in Excel has a number of advantages over the VLOOKUP function:

1. INDEX and MATCH are more flexible and faster than Vlookup
2. It is possible to execute horizontal lookup, vertical lookup, 2-way lookup, left lookup, case-sensitive lookup, and even lookups based on multiple criteria.
3. In sorted Data, INDEX-MATCH is 30% faster than VLOOKUP. This means that in a larger dataset 30% faster makes more sense.

Let's begin with the detailed concepts of each INDEX and MATCH.

INDEX Function

The INDEX function in Excel is very powerful at the same time a flexible tool that retrieves the value at a given location in a range. In other words, It returns the content of a cell, specified by row and column offset.

Syntax:

=INDEX(reference, [row], [column])

Parameters:

- **reference:** The array of cells to be offset into. It can be a single range or an entire dataset in a table of data.
- **row [optional]:** The number of offset rows. It means if we choose a table reference range as "A1:A5" then the Cell/content that we want to extract is at how much vertical distance. Here, for A1 row will be 1, for A2 row = 2, and so on. If we give row = 4 then it will extract A4. As row is optional so if we don't specify any row number then it extracts entire rows in the reference range. That is A1 to A5 in this case.
- **column [optional]:** The number of offset columns. It means if we choose a table reference range as "A1:B5" then the Cell/content we want to extract is at how much horizontal distance. Here, for A1 row will be 1 and the column will be 1, for B1 row will be 1 but the column will be 2 similarly for A2 row = 2 column = 1, for B2 row = 2 column = 2, and so on. If we give

row = 5 and column 2 then it will extract B5. As the column is optional so if we don't specify any row no. then it will extract the entire column in the reference range. For example, if we give row = 2 and column as empty then it will extract (A2:B2). If we don't specify Row and column both then it will extract the entire reference table that is (A1:B5).

Reference Table: The following table will be used as a reference table for all the examples of the INDEX function. The first Cell is at B3 ("FOOD") and the Last Diagonal Cell is at F10 ("180").

E2 ▾ *fx* |

	A	B	C	D	E	F
1						
2						
3		Food	Types	Cost	Quantity	Total Cost
4		Chowmein	Chinese	50	2	100
5		Pasta	Italian	60	1	60
6		Dhokla	Gujrati	40	1	40
7		Masala Dosa	South Indian	100	2	200
8		Biryani	Hyderabadi	140	1	140
9		Cheese Maggi	Chinese	80	4	320
10		Uttapam	South Indian	60	3	180

Examples: Below are some examples of Index functions.

Excel for Data Analysts

Case 1: No Rows and Columns are mentioned.

Input Command: =INDEX(B3:C10)

G3 fx =INDEX(B3:C10)

	A	B	C	D	E	F	G	H
1								
2							Output	
3		Food	Types	Cost	Quantity	Total Cost	Food	Types
4		Chowmein	Chinease	50	2	100	Chowmein	Chinease
5		Pasta	Italian	60	1	60	Pasta	Italian
6		Dhokla	Gujrati	40	1	40	Dhokla	Gujrati
7		Masala Dosa	South Indian	100	2	200	Masala Dosa	South Indian
8		Biryani	Hyderabadi	140	1	140	Biryani	Hyderabadi
9		Cheese Maggi	Chinease	80	4	320	Cheese Maggi	Chinease
10		Uttapam	South Indian	60	3	180	Uttapam	South Indian

Case 2: Only Rows are Mentioned.

Input Command: =INDEX(B3:C10,2)

G3 fx =INDEX(B3:C10,2)

	A	B	C	D	E	F	G	H
1								
2							Output	
3		Food	Types	Cost	Quantity	Total Cost	Chowmein	Chinease
4		Chowmein	Chinease	50	2	100		
5		Pasta	Italian	60	1	60		
6		Dhokla	Gujrati	40	1	40		
7		Masala Dosa	South Indian	100	2	200		
8		Biryani	Hyderabadi	140	1	140		
9		Cheese Maggi	Chinease	80	4	320		
10		Uttapam	South Indian	60	3	180		
11								

Excel for Data Analysts

Case 3: Both Rows And Columns are mentioned.

Input Command: =INDEX(B3:D10,4,2)

G3 fx =INDEX(B3:D10,4,2)

	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	Gujrati
4		Chowmein	Chinease	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinease	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							
12							

Case 4: Only Columns are mentioned.

Input Command: =INDEX(B3 : D10 , , 2)

G3 fx =INDEX(B3:D10 , , 2)

	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	Types
4		Chowmein	Chinease	50	2	100	Chinease
5		Pasta	Italian	60	1	60	Italian
6		Dhokla	Gujrati	40	1	40	Gujrati
7		Masala Dosa	South Indian	100	2	200	South Indian
8		Biryani	Hyderabadi	140	1	140	Hyderabadi
9		Cheese Maggi	Chinease	80	4	320	Chinease
10		Uttapam	South Indian	60	3	180	South Indian
11							

Problem With INDEX Function: The problem with the INDEX function is that there is a need to specify rows and columns for the data that we are looking for. Let's assume we are dealing with a machine learning dataset of 10000 rows and columns then it will be very difficult to search and extract the data that we are looking for. Here comes the concept of Match Function, which will identify rows and columns based on some condition.

MATCH Function

It retrieves the position of an item/value in a range. It is a less refined version of a VLOOKUP or HLOOKUP that only returns the location information and not the actual data. MATCH is not case-sensitive and does not care whether the range is Horizontal or Vertical.

Syntax:

=MATCH(search_key, range, [search_type])

Parameters:

- **search_key:** The value to search for. For example, 42, "Cats", or I24.
- **range:** The one-dimensional array to be searched. It Can Either be a single row or a single column.eg->A1:A10 , A2:D2 etc.
- **search_type [optional]:** The search method. = 1 (default) finds the largest value less than or equal to search_key when the range is sorted in ascending order.
 - = 0 finds the exact value when the range is unsorted.
 - = -1 finds the smallest value greater than or equal to search_key when the range is sorted in descending order.

Row number or Column number can be found using the match function and can use it inside the index function so if there is any detail about an item, then all information can be extracted about the item by finding the row/column of the item using match then nesting it into index function.

Reference Table: The following table will be used as a reference table for all the examples of the MATCH function. First Cell is at B3 ("FOOD") and the Last Diagonal Cell is At F10 ("180")

Excel for Data Analysts

E2 ▾ *fx* |

	A	B	C	D	E	F
1						
2						
3		Food	Types	Cost	Quantity	Total Cost
4		Chowmein	Chinease	50	2	100
5		Pasta	Italian	60	1	60
6		Dhokla	Gujrati	40	1	40
7		Masala Dosa	South Indian	100	2	200
8		Biryani	Hyderabadi	140	1	140
9		Cheese Maggi	Chinease	80	4	320
10		Uttapam	South Indian	60	3	180

Examples: Below are some examples of the MATCH function-

Case 1: Search Type 0, It means Exact Match.

Input Command: =MATCH("South Indian",C3:C10,0)

G3 ▾ *fx* | =MATCH("South Indian",C3:C10,0)

	A	B	C	D	E	F	G
1							
2							
3		Food	Types	Cost	Quantity	Total Cost	Output
4		Chowmein	Chinease	50	2	100	5
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinease	80	4	320	
10		Uttapam	South Indian	60	3	180	

Excel for Data Analysts

Case 2: Search Type 1 (Default).

Input Command: =MATCH("South Indian",C3:C10)

G3		fx	=MATCH("South Indian",C3:C10)				
	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	8
4		Chowmein	Chinease	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinease	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							

Case 3: Search Type -1.

Input Command: =MATCH("South Indian",C3:C10,-1)

G3		fx	=MATCH("South Indian",C3:C10,-1)				
	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	5
4		Chowmein	Chinease	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinease	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							

INDEX-MATCH Together

In the previous examples, the static values of rows and columns were provided in the INDEX function. Let's assume there is no prior knowledge about the rows and column position then rows and columns position can be provided using the MATCH function. This is a dynamic way to search and extract value.

Syntax:

**=INDEX(Reference Table , [Match(SearchKey,Range,Type)/StaticRowPosition],
[Match(SearchKey,Range,Type)/StaticColumnPosition])**

Reference Table: The following reference table will be used. The first Cell is at B3 ("FOOD") and the Last Diagonal Cell is At F10 ("180")

E2						
	A	B	C	D	E	F
1						
2						
3		Food	Types	Cost	Quantity	Total Cost
4		Chowmein	Chinese	50	2	100
5		Pasta	Italian	60	1	60
6		Dhokla	Gujrati	40	1	40
7		Masala Dosa	South Indian	100	2	200
8		Biryani	Hyderabadi	140	1	140
9		Cheese Maggi	Chinese	80	4	320
10		Uttapam	South Indian	60	3	180

Example: Let's say the task is to find the cost of Masala Dosa. It is known that column 3 represents the cost of items, but the row position of Masala Dosa is not known. The problem can be divided into two steps-

Step 1: Find the position of Masala Dosa by using the formula:

=MATCH("Masala Dosa",B3:B10,0)

Here B3:B10 represents Column "Food" and 0 means Exact Match. It will return the row number of Masala Dosa.

Step 2: Find the cost of Masala Dosa. Use the INDEX Function to find the cost of Masala Dosa. By substituting the above MATCH function query inside the INDEX

=INDEX(B3:F10, MATCH("Masala Dosa", B3:B10 , 0) ,3)

G3		=INDEX(B3:F10, MATCH("Masala Dosa", B3:B10 , 0) , 3)					
	A	B	C	D	E	F	G
1							
2							
3		Food	Types	Cost	Quantity	Total Cost	Output
4		Chowmein	Chinese	50	2	100	100
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinese	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							

In the previous example, the column position Of the “Cost” attribute was hardcoded. So, It was not fully dynamic.

=MATCH("Cost",B3:F3,0)

Case 2: When row, as well as column value, are provided via MATCH function (without giving static value) then it is called Two-Way Lookup. It can be achieved using the formula:

G3								
	A	B	C	D	E	F	G	
1								
2								
3		Food	Types	Cost	Quantity	Total Cost	Output	
4		Chowmein	Chinese	50	2	100	100	
5		Pasta	Italian	60	1	60		
6		Dhokla	Gujrati	40	1	40		
7		Masala Dosa	South Indian	100	2	200		
8		Biryani	Hyderabadi	140	1	140		
9		Cheese Maggi	Chinese	80	4	320		
10		Uttapam	South Indian	60	3	180		
11								

Left Lookup

One of the key advantages of INDEX and MATCH over the VLOOKUP function is the ability to perform a “left lookup”. It means it is possible to extract the row position of an item from using any attribute at right and the value of another attribute in left can be extracted.

For Example, Let’s say buy food whose cost should be 140 Rs. Indirectly we are saying buy “Biryani”. In this example, the cost Rs 140/- is known, there is a need to extract the “Food”. Since the Cost column is placed to the right of the Food column. If VLOOKUP is applied it will not be able to search the left side of the Cost column. That is why using VLOOKUP it is not possible to get Food Name.

To overcome this disadvantage INDEX-MATCH function Left lookup can be used.

Step 1: First extract row position of Cost 140 Rs using the formula:

=MATCH(140, D3:D10,0)

Here D3: D10 represents the Cost column where the search for the Cost 140 Rs row number is being done.

Step 2: After getting the row number, the next step is to use the INDEX Function to extract Food Name using the formula:

=INDEX(B3:B10, MATCH(140, D3:D10,0))

Here B3:B10 represents Food Column and 140 is the Cost of the food item.

G3		=INDEX (B3:B10 , MATCH(140, D3:D10 ,0))					
	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	Biryani
4		Chowmein	Chinese	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinese	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							
12							

Case Sensitive Lookup

By itself, the MATCH function is not case-sensitive. This means if there is a Food Name “DHOKLA” and the MATCH function is used with the following search word:

1. "Dhokla"
2. "dhokla"
3. "DhOkLA"

All will return the row position of DHOKLA. However, the EXACT function can be used with INDEX and MATCH to perform a lookup that respects upper and lower case.

Exact Function: The Excel EXACT function compares two text strings, taking into account upper and lower case characters, and returns TRUE if they are the same, and FALSE if not. EXACT is case-sensitive.

Examples:

1. **EXACT("DHOKLA","DHOKLA")**: This will return True.
2. **EXACT("DHOKLA","Dhokla")**: This will return False.
3. **EXACT("DHOKLA","dhokla")**: This will return False.
4. **EXACT("DHOKLA","DhOkLA")**: This will return False.

Example: Let say the task is to search for the Type Of Food “Dhokla” but in Case-Sensitive Way. This can be done using the formula-

=INDEX(C3:C10, MATCH(TRUE , EXACT("Dhokla", B3:B10) ,0))

Here the EXACT function will return True if the value in Column B3:B10 matches with “Dhokla” with the same case, else it will return False. Now MATCH function will apply in Column B3:B10 and search for a row with the Exact value TRUE. After that INDEX Function will retrieve the value of Column C3:C10 (Food Type Column) at the row returned by the MATCH function.

G3		fx	=INDEX(C3:C10, MATCH(TRUE , EXACT("Dhokla", B3:B10) ,0))				
	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	Gujrati
4		Chowmein	Chinease	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinease	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							

Multiple Criteria Lookup

One of the trickiest problems in Excel is a lookup based on multiple criteria. In other words, a lookup that matches more than one column at the same time. In the example below, the INDEX and MATCH functions and Boolean logic are used to match on 3 columns-

1. Food.
2. Cost.
3. Quantity.

To extract total cost.

Example: Let's say the task is to calculate the total cost of Pasta where

1. **Food:** Pasta.
2. **Cost:** 60.
3. **Quantity:** 1.

So in this example, there are three criteria to perform a Match. Below are the steps for the search based on multiple criteria-

Step 1: First match Food Column (B3:B10) with Pasta using the formula:

"PASTA" = B3:B10

This will convert B3:B10 (Food Column) values as Boolean. That Is True where Food is Pasta or else False.

Step 2: After that, match the Cost criteria in the following manner:

60 = D3:D10

This will replace D3:D10 (Cost Column) values as Boolean. That is True where Cost=60 else False.

Step 3: Next step is to match the third criteria that are Quantity = 1 in the following manner:

1 = E3:E10

This will replace E3:E10 Column (Quantity Column) as True where Quantity = 1 else it will be False.

Step 4: Multiply the result of the first, second, and third criteria. This will be the intersection of all conditions and convert Boolean True / False as 1/0.

Step 5: Now the result will be a Column with 0 And 1. Here use the MATCH Function to find the row number of columns that contain 1. Because if a column is having the value 1, then it means it satisfies all three criteria.

Step 6: After getting the row number use the INDEX function to get the total cost of that row.

=INDEX(F3:F10, MATCH(1, ("Pasta"=B3:B10) * (60=D3:D10) * (1=E3:E10) , 0))

Here F3:F10 represents the Total Cost Column.

G3		fx	$\text{=INDEX(F3:F10, MATCH(1, ("Pasta"=B3:B10) * (60=D3:D10) * (1=E3:E10) , 0))}$				
	A	B	C	D	E	F	G
1							
2							Output
3		Food	Types	Cost	Quantity	Total Cost	60
4		Chowmein	Chinese	50	2	100	
5		Pasta	Italian	60	1	60	
6		Dhokla	Gujrati	40	1	40	
7		Masala Dosa	South Indian	100	2	200	
8		Biryani	Hyderabadi	140	1	140	
9		Cheese Maggi	Chinese	80	4	320	
10		Uttapam	South Indian	60	3	180	
11							

3. Excel CONCAT Function

CONCAT Function

To concatenate is to link something together.

CONCAT is a function in Excel and is short for concatenate.

The **CONCAT** function is used to link multiple cells without adding any delimiters between the combined cell values.

It is typed =CONCAT

=CONCAT(**cell1, delimiter, cell2**)

Note: Delimiters are spacing or symbols used to separate content elements apart from each other.

Example of delimiters

comma , semicolon ; quotes " or ' braces {} pipes | slashes / \

Info: CONCAT is a new version of the old CONCATENATE function. Recent versions of Excel do not accept the old CONCATENATE function as it is only compatible with earlier versions of Excel. To use CONCATENATE in the latest versions of Excel, use CONCAT.


How to Use CONCAT Function

To combine values from multiple Excel cells, use **CONCAT**.

Note: The different parts of the function are separated by a symbol, like comma, or semicolon ;

Step 1) Start the CONCAT function

1. Select a cell E2
2. Type =CONCAT
3. Double click the **CONCAT** command

	A	B	C	D	E	F	G
1	Name	Type 1	Total stats				
2	Mankey	Fighting	305		=CONCAT		
3	Poliwrath	Water	510		 CONCAT	Concatenates a list or range of text strings	
4	Victreebel	Grass	490				
5	Tentacool	Water	335				
6	Magnetron	Electric	465				
7	Dewgong	Water	475				
8	Cloyster	Water	525				
9	Onix	Rock	385				
10	Dragonair	Dragon	420				
11	Pidgeotto	Normal	349				
12	Rattata	Normal	253				
13	Beedrill	Bug	395				
14	Doduo	Normal	310				
15	Kingler	Water	475				
16	Nidoqueen	Poison	505				
17	Hitmonchan	Fighting	455				
18	Charmeleon	Fire	405				
19	Arbok	Poison	438				
20	Gastly	Ghost	310				
21	Magikarp	Water	200				

Step 2) Link cells

4. Select a cell (A2)
5. Add a comma (,)
6. Add a space inside of quotation marks to create a delimiter (" ")
7. Select another cell (A3)
8. Hit enter

Note: The **CONCAT** function does not provide any delimiters between cell values used in the function. If you want your cell values to be displayed with delimiters, you have to add the delimiters between the selected cells. You can use any symbol as a delimiter, if you want the delimiter to be a space, enter " ".

	A	B	C	D	E
1	Name	Type 1	Total stats		
2	Mankey	Fighting	305		=CONCAT(A2," ",A3)
3	Poliwrath	Water	510		
4	Victreebel	Grass	490		
5	Tentacool	Water	335		
6	Magnetron	Electric	465		
7	Dewgong	Water	475		
8	Cloyster	Water	525		
9	Onix	Rock	385		
10	Dragonair	Dragon	420		
11	Pidgeotto	Normal	349		
12	Rattata	Normal	253		
13	Beedrill	Bug	395		
14	Doduo	Normal	310		
15	Kingler	Water	475		
16	Nidoqueen	Poison	505		
17	Hitmonchan	Fighting	455		
18	Charmeleon	Fire	405		
19	Arbok	Poison	438		
20	Gastly	Ghost	310		
21	Magikarp	Water	200		
22					

The function returns the content of A2 and A3 separated with a " " dilimiter.

	A	B	C	D	E
1	Name	Type 1	Total stats		
2	Mankey	Fighting	305		Mankey Poliwrath
3	Poliwrath	Water	510		
4	Victreebel	Grass	490		
5	Tentacool	Water	335		
6	Magnetron	Electric	465		
7	Dewgong	Water	475		
8	Cloyster	Water	525		
9	Onix	Rock	385		
10	Dragonair	Dragon	420		
11	Pidgeotto	Normal	349		
12	Rattata	Normal	253		
13	Beedrill	Bug	395		
14	Doduo	Normal	310		
15	Kingler	Water	475		
16	Nidoqueen	Poison	505		
17	Hitmonchan	Fighting	455		
18	Charmeleon	Fire	405		
19	Arbok	Poison	438		
20	Gastly	Ghost	310		
21	Magikarp	Water	200		
22					

You can also use more symbols as a demilter, such as adding the word " and ".

Excel for Data Analysts

	A	B	C	D	E
1	Name	Type 1	Total stats		
2	Mankey	Fighting	305		=CONCAT(A2," and ",A3)
3	Poliwrath	Water	510		
4	Victreebel	Grass	490		
5	Tentacool	Water	335		
6	Magnetron	Electric	465		
7	Dewgong	Water	475		
8	Cloyster	Water	525		
9	Onix	Rock	385		
10	Dragonair	Dragon	420		
11	Pidgeotto	Normal	349		
12	Rattata	Normal	253		
13	Beedrill	Bug	395		
14	Doduo	Normal	310		
15	Kingler	Water	475		
16	Nidoqueen	Poison	505		
17	Hitmonchan	Fighting	455		
18	Charmeleon	Fire	405		
19	Arbok	Poison	438		
20	Gastly	Ghost	310		
21	Magikarp	Water	200		

	A	B	C	D	E
1	Name	Type 1	Total stats		
2	Mankey	Fighting	305		Mankey and Poliwrath
3	Poliwrath	Water	510		
4	Victreebel	Grass	490		
5	Tentacool	Water	335		
6	Magnetron	Electric	465		
7	Dewgong	Water	475		
8	Cloyster	Water	525		
9	Onix	Rock	385		
10	Dragonair	Dragon	420		
11	Pidgeotto	Normal	349		
12	Rattata	Normal	253		
13	Beedrill	Bug	395		
14	Doduo	Normal	310		
15	Kingler	Water	475		
16	Nidoqueen	Poison	505		
17	Hitmonchan	Fighting	455		
18	Charmeleon	Fire	405		
19	Arbok	Poison	438		
20	Gastly	Ghost	310		
21	Magikarp	Water	200		

The **CONCAT** function accepts cell values of both text and numbers, you can combine them in any way you prefer.

Combine the cell values with text strings as delimiters to make the output more descriptive!

Excel for Data Analysts

	A	B	C	D	E	F
1	Name	Type 1	Total stats			
2	Mankey	Fighting	305		=CONCAT("The total stats of ",A3," are ",C3,".")	
3	Poliwrath	Water	510			
4	Victreebel	Grass	490			
5	Tentacool	Water	335			
6	Magnetron	Electric	465			
7	Dewgong	Water	475			
8	Cloyster	Water	525			
9	Onix	Rock	385			
10	Dragonair	Dragon	420			
11	Pidgeotto	Normal	349			
12	Rattata	Normal	253			
13	Beedrill	Bug	395			
14	Doduo	Normal	310			
15	Kingler	Water	475			
16	Nidoqueen	Poison	505			
17	Hitmonchan	Fighting	455			
18	Charmeleon	Fire	405			
19	Arbok	Poison	438			
20	Gastly	Ghost	310			
21	Magikarp	Water	200			
22						

	A	B	C	D	E
1	Name	Type 1	Total stats		
2	Mankey	Fighting	305		The total stats of Poliwrath are 510.
3	Poliwrath	Water	510		
4	Victreebel	Grass	490		
5	Tentacool	Water	335		
6	Magnetron	Electric	465		
7	Dewgong	Water	475		
8	Cloyster	Water	525		
9	Onix	Rock	385		
10	Dragonair	Dragon	420		
11	Pidgeotto	Normal	349		
12	Rattata	Normal	253		
13	Beedrill	Bug	395		
14	Doduo	Normal	310		
15	Kingler	Water	475		
16	Nidoqueen	Poison	505		
17	Hitmonchan	Fighting	455		
18	Charmeleon	Fire	405		
19	Arbok	Poison	438		
20	Gastly	Ghost	310		
21	Magikarp	Water	200		
22					

4. Excel IFERROR Function

When you work with data and formulas in Excel, you're bound to encounter errors.

To handle errors, Excel has provided a useful function – the IFERROR function.

Before we get into the mechanics of using the IFERROR function in Excel, let's first go through the different kinds of errors you can encounter when working with formulas.

Types of Errors in Excel

Knowing the errors in Excel will better equip you to identify the possible reason and the best way to handle these.




Below are the types of errors you might find in Excel.

#N/A Error


This is called the 'Value Not Available' error.

You will see this when you use a lookup formula and it can't find the value (hence Not Available).

Below is an example where I use the VLOOKUP formula to find the price of an item, but it returns an error when it can't find that item in the table array.

E3	:	  	=VLOOKUP(D3,\$A\$2:\$B\$6,2,0)
----	---	---	--------------------------------

	A	B	C	D	E
1	Product	Price			
2	Item A	44			
3	Item B	55			
4	Item C	56			
5	Item D	59			
6	Item E	71			




Lookup Value	Result
Item F 	#N/A

#DIV/0! Error

You're likely to see this error when a number is divided by 0.

This is called the division error. In the below example, it gives a #DIV/0! error as the quantity value (the divisor in the formula) is 0.

Excel for Data Analysts




C4	:	  	=A4/B4
	A	B	C
1	Total Cost	Quantity	Cost Per Quantity
2	3320	50	66.4
3	1800	16	112.5
4	2250	0	#DIV/0!
5	4200	25	168.0
6	4800	21	228.6
7	3270	45	72.7
8	2700	54	50.0

#VALUE! Error

The value error occurs when you use an incorrect data type in a formula.

For example, in the below example, when I try to add cells that have numbers and character A, it gives the value error.

This happens as you can only add numeric values, but instead, I tried adding a number with a text character.

A5	:	  	=A1+A2+A3
	A	B	C
1	1		
2	2		
3	A		
4			
5	#VALUE!		

#REF! Error

This is called the reference error and you will see this when the reference in the formula is no longer valid. This could be the case when the formula refers to a cell reference and that cell reference does not exist (happens when you delete a row/column or worksheet that was referred to in the formula).

Excel for Data Analysts

In the below example, while the original formula was =A2/B2, when I deleted Column B, all the references to it became #REF! and it also gave the #REF! error as the result of the formula.

B2	:	X	✓	<i>fx</i>	=A2/#REF!
	A	B			
1	Total Cost	Cost Per Quantity			
2	3520	#REF!			
3	1800	#REF!			
4	2250	#REF!			
5	4200	#REF!			
6	4800	#REF!			
7	3270	#REF!			
8	2700	#REF!			

#NAME ERROR

This error is likely to be a result of a misspelled function.

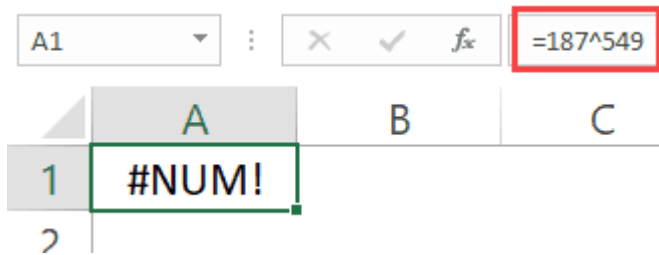
For example, if instead of VLOOKUP, you by mistake use VLOKUP, it will give a name error.

E3	:	X	✓	<i>fx</i>	=VLOKUP(D3,\$A\$2:\$B\$6,2,0)
	A	B	C	D	E
1	Product	Price			
2	Item A	44			
3	Item B	55			
4	Item C	56			
5	Item D	59			
6	Item E	71			

Lookup Value	Result
Item A	#NAME?

#NUM ERROR

Num error can occur if you try and calculate a very large value in Excel. For example, =187^549 will return a number error.



Another situation where you can get the NUM error is when you give a non-valid number argument to a formula. For example, if you're calculating the Square Root of a number and you give a negative number as the argument, it will return a number error.

For example, in the case of the Square Root function, if you give a negative number as the argument, it will return a number error (as shown below).

While I have shown only a couple of examples here, there can be many other reasons that can lead to errors in Excel. When you get errors in Excel, you can't just leave it there. If the data is further used in calculations, you need to make sure the errors are handled the right way.

Excel's IFERROR function is a great way to handle all types of errors in Excel.

Using the IFERROR function, you can specify what you want the formula to return instead of the error. If the formula does not return an error, then its own result is returned.

IFERROR Function Syntax

`=IFERROR(value, value_if_error)`

Input Arguments

- **value** – this is the argument that is checked for the error. In most cases, it is either a formula or a cell reference.
- **value_if_error** – this is the value that is returned if there is an error. The following error types evaluated: #N/A, #REF!, #DIV/0!, #VALUE!, #NUM!, #NAME?, and #NULL!.

Additional Notes:

- If you use `""` as the `value_if_error` argument, the cell displays nothing in case of an error.

Excel for Data Analysts

- If the value or value_if_error argument refers to an empty cell, it is treated as an empty string value by the Excel IFERROR function.
- If the value argument is an array formula, IFERROR will return an array of results for each item in the range specified in value.

Excel IFERROR Function – Examples

Here are three examples of using IFERROR function in Excel.

Example 1 – Return Blank Cell Instead of Error

If you have functions that may return an error, you can wrap it within the IFERROR function and specify blank as the value to return in case of an error.

In the example shown below, the result in D4 is the #DIV/0! error as the divisor is 0.

A4	:		=A1/A2
	A	B	C
1	10		
2	0		
3			
4	#DIV/0!		

In this case, you can use the following formula to return blank instead of the ugly DIV error.

=IFERROR(A1/A2,"")

A4	:		=IFERROR(A1/A2,"")		
	A	B	C	D	
1	10				
2	0				
3					
4					
5					




This IFERROR function would check whether the calculation leads to an error. If it does, it simply returns a blank as specified in the formula.

Excel for Data Analysts

Here, you can also specify any other string or formula to display instead of the blank.

For example, the below formula would return the text “Error”, instead of the blank cell.

`=IFERROR(A1/A2,"Error")`




A4	:				=IFERROR(A1/A2,"Error")
	A	B	C	D	
1	10				
2	0				
3					
4	Error				

Note: If you are using Excel 2003 or a prior version, you will not find the IFERROR function in it. In such cases, you need to use the combination of IF function and ISERROR function.

Example 2 – Return ‘Not Found’ when VLOOKUP Can’t Find a Value

When you use the Excel VLOOKUP Function, and it can’t find the lookup value in the specified range, it would return the #N/A error.

For example, below is a data set of student names and their marks. I have used the VLOOKUP function to fetch the marks of three students (in D2, D3, and D4).

E4	:	  	=VLOOKUP(D4,\$A\$2:\$B\$12,2,0)		
	A	B	C	D	E
1	Names	Marks			
2	Mark	90		Jack	78
3	Sam	89		Jill	68
4	Martha	48		Josh	#N/A
5	Hans	94			
6	Jill	68			
7	Jack	78			
8	Stuart	53			
9	Tom	46			
10	Bob	65			
11	Grace	65			
12	Alice	53			




Excel for Data Analysts

While the VLOOKUP formula in the above example finds the names of first two students, it can't find Josh's name on the list and hence it returns the #N/A error.

Here, we can use the IFERROR function to return a blank or some meaningful text instead of the error.

Below is the formula that will return 'Not Found' instead of the error.

`=IFERROR(VLOOKUP(D2,A2:B12,2,0),"Not Found")`

E4	:	  	<code>=IFERROR(VLOOKUP(D4,\$A\$2:\$B\$12,2,0),"Not Found")</code>			
	A	B	C	D	E	F
1	Names	Marks				
2	Mark	90		Jack	78	
3	Sam	89		Jill	68	
4	Martha	48		Josh	Not Found	
5	Hans	94				
6	Jill	68				
7	Jack	78				
8	Stuart	53				
9	Tom	46				
10	Bob	65				
11	Grace	65				
12	Alice	53				

Note that you can also use IFNA instead of IFERROR with VLOOKUP. While IFERROR would treat all kinds of error values, IFNA would only work on the #N/A errors and wouldn't work with other errors.

Example 3 – Return 0 in case of an Error

If you don't specify the value to return by IFERROR in the case of an error, it would automatically return 0.

For example, if I divide 100 with 0 as shown below, it would return an error.

A4	:	<i>fx</i>	=A1/A2
	A	B	C
1	100		
2	0		
3			
4	#DIV/0!		

However, if I use the below IFERROR function, it would return a 0 instead. Note that you still need to use a comma after the first argument.

A4	:			<i>fx</i>	=IFERROR(A1/A2,)
	A	B	C	D	
1	100				
2	0				
3					
4	0				

Example 4 – Using Nested IFERROR with VLOOKUP

Sometimes when using VLOOKUP, you may have to look through the fragmented table of arrays. For example, suppose you have the sales transaction records in 2 separate worksheets and you want to look-up an item number and see it's value.

Doing this requires using nested IFERROR with VLOOKUP.

Suppose you have a dataset as shown below:

	A	B	C	D	E	F	G	H
1	Name	Score		Name	Score			
2	Tom	45		Martha	50			
3	Brad	73		Sam	90		Grace	
4	Jill	76		Brent	85			
5	Phillip	71		Grace	81			

In this case, to find the score for Grace, you need to use the below nested IFERROR formula:

```
=IFERROR(VLOOKUP(G3,$A$2:$B$5,2,0),IFERROR(VLOOKUP(G3,$D$2:$E$5,2,0), "Not Found"))
```

This kind of formula nesting ensure that you get the value from either of the table and any error returned is handled.

Note that in case the tables are on the same worksheet, however, in a real-life example, it likely to be on different worksheets.

5. Excel Filter Function

Below is the syntax of the FILTER function:

`=FILTER(array,include,[if_empty])`

- **array** – this is the range of cells where you have the data and you want to filter some data from it
- **include** – this is the condition that tells the function what records to filter
- **[if_empty]** – this is an optional argument where you can specify what to return in case no results are found by the FILTER function. By default (when not specified), it returns the #CALC! error

Now let's have a look at some amazing Filter function examples and stuff it can do which used to be quite complex in its absence.

Example 1: Filtering Data Based on One Criteria (Region)

Suppose you have a dataset as shown below and you want to filter all the records for the US only.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

Below is the FILTER formula that will do this:

`=FILTER(A2:C11,B2:B11="US")`

E2				=FILTER(\$A\$2:\$C\$11,\$B\$2:\$B\$11="US")			
	A	B	C	D	E	F	G
1	Name	Region	Sales		Name	Region	Sales
2	Rob	US	14020		Rob	US	14020
3	Alice	Canada	18385		Greg	US	19534
4	Greg	US	19534		Sheen	US	11552
5	Sheen	US	11552		Jenny	US	5306
6	Amy	Asia	11932				
7	Jenny	US	5306				
8	Bob	Asia	8815				
9	Stan	Asia	10396				
10	Lisa	Canada	7763				
11	Barb	Brazil	11958				

The above formula uses the dataset as the array and the condition is `B2:B11="US"`

This condition would make the FILTER function check every cell in column B (one that has the region) and only those records that match this criterion would be filtered.

Also, in this example, I have the original data and the filtered data on the same sheet, but you can also have these in separate sheets or even workbooks.

Filter Function returns a result that is a dynamic array (which means that instead of returning one value, it returns an array that spills to other cells).

For this to work, you need to have an area where the result would come to be empty. In any of the cells in this area (E2:G5 in this example) already has something in it, the function will give you the #SPILL error.

Also, since this is a dynamic array, you can not change a part of the result. You can either delete the entire range that has the result or cell E2 (where the formula was entered). Both of these would delete the entire resulting array. But you can not change any individual cell (or delete it).

In the above formula, I have hard-coded the region value, but you can also have it in a cell and then reference that cell that has the region value.

For example, in the below example, I have the region value in cell I2 and this is then referenced in the formula:

Excel for Data Analysts

=FILTER(\$A\$2:\$C\$11,\$B\$2:\$B\$11=I1)

This makes the formula even more useful and now you can simply change the region value in cell I2 and the filter would automatically change.

You can also have a drop-down in cell I2 where you can simply make the selection and it would instantly update the filtered data.

Example 2: Filtering Data Based on One Criteria (More Than or Less Than)

You can also use comparative operators within the filter function and extract all the records that are more or less than a specific value.

For example, suppose you have the dataset as shown below and you want to filter all the records where the sales value is more than 10000.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

The below formula can do this:

=FILTER(\$A\$2:\$C\$11,(\$C\$2:\$C\$11>10000))

E2

✖

✓

fx

=FILTER(\$A\$2:\$C\$11,(\$C\$2:\$C\$11>=10000))

	A	B	C	D	E	F	G	H
1	Name	Region	Sales		Name	Region	Sales	
2	Rob	US	14020		Rob	US	14020	
3	Alice	Canada	18385		Alice	Canada	18385	
4	Greg	US	19534		Greg	US	19534	
5	Sheen	US	11552		Sheen	US	11552	
6	Amy	Asia	11932		Amy	Asia	11932	
7	Jenny	US	5306		Stan	Asia	10396	
8	Bob	Asia	8815		Barb	Brazil	11958	
9	Stan	Asia	10396					
10	Lisa	Canada	7763					
11	Barb	Brazil	11958					

The array argument refers to the entire dataset and the condition, in this case, is ($\$C\$2:\$C\$11 \geq 10000$).

The formula checks each record for the value in Column C. If the value is more than 10000, it is filtered, else it's ignored.

In case you want to get all the records less than 10000, you can use the below formula:

```
=FILTER($A$2:$C$11,($C$2:$C$11<10000))
```

You can also get more creative with the FILTER formula. For example, if you want to filter the top three record based on the sales value, you can use the below formula:

```
=FILTER($A$2:$C$11,($C$2:$C$11>=LARGE(C2:C11,3)))
```

E2				fx		=FILTER(\$A\$2:\$C\$11,(\$C\$2:\$C\$11>=LARGE(C2:C11,3)))			
	A	B	C	D	E	F	G	H	I
1	Name	Region	Sales		Name	Region	Sales		
2	Rob	US	14020		Rob	US	14020		
3	Alice	Canada	18385		Alice	Canada	18385		
4	Greg	US	19534		Greg	US	19534		
5	Sheen	US	11552						
6	Amy	Asia	11932						
7	Jenny	US	5306						
8	Bob	Asia	8815						
9	Stan	Asia	10396						
10	Lisa	Canada	7763						
11	Barb	Brazil	11958						

The above formula uses the LARGE function to get the third-largest value in the dataset. This value is then used in the FILTER function criteria to get all the records where the sales value is more than or equal to the third-largest value.

Example 3: Filtering Data with Multiple Criteria (AND)

Suppose you have the below dataset and you want to filter all the records for the US where the sale value is more than 10000.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

This is an AND condition where you need to check for two things – the region needs to be the US and the sales need to be more than 10000. If only one condition is met, the results should not be filtered.

Excel for Data Analysts

Below is the FILTER formula that will filter records with the US as the region and sales of more than 10000:

```
=FILTER($A$2:$C$11,($B$2:$B$11="US")*($C$2:$C$11>10000))
```

E2										
	A	B	C	D	E	F	G	H	I	
1	Name	Region	Sales		Name	Region	Sales			
2	Rob	US	14020		Rob	US	14020			
3	Alice	Canada	18385		Greg	US	19534			
4	Greg	US	19534		Sheen	US	11552			
5	Sheen	US	11552							
6	Amy	Asia	11932							
7	Jenny	US	5306							
8	Bob	Asia	8815							
9	Stan	Asia	10396							
10	Lisa	Canada	7763							
11	Barb	Brazil	11958							

Note that the criterion (called the include argument) is (\$B\$2:\$B\$11="US")*(\$C\$2:\$C\$11>10000)

Since I am using two conditions and I need both to be true, I have used the multiplication operator to combine these two criteria. This returns an array of 0's and 1's, where a 1 is returned only when both conditions are met.

In case there are no records that meet the criteria, the function would return the #CALC! error.

And in case you want to return something meaningful (instead of the error), you can use a formula as shown below:

```
=FILTER($A$2:$C$11,($B$2:$B$11="USA")*($C$2:$C$11>10000),"Nothing Found")
```

Here, I have used "Not Found" as the third argument, which is used when no records are found that match the criteria.

Example 4: Filtering Data with Multiple Criteria (OR)

You can also modify the 'include' argument in the FILTER function to check for an OR criteria (where any one of the given conditions can be true).

For example, suppose you have the dataset as shown below and you want to filter the records where the country is either the US or Canada.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

Below is the formula that will do this:

=FILTER(\$A\$2:\$C\$11,(\$B\$2:\$B\$11="US")+(\$B\$2:\$B\$11="Canada"))

Formula bar: `=FILTER(A2:C11,(B2:B11="US")+(B$2:B$11="Canada"))`

	A	B	C	D	E	F	G	H	I
1	Name	Region	Sales		Name	Region	Sales		
2	Rob	US	14020		Rob	US	14020		
3	Alice	Canada	18385		Alice	Canada	18385		
4	Greg	US	19534		Greg	US	19534		
5	Sheen	US	11552		Sheen	US	11552		
6	Amy	Asia	11932		Jenny	US	5306		
7	Jenny	US	5306		Lisa	Canada	7763		
8	Bob	Asia	8815						
9	Stan	Asia	10396						
10	Lisa	Canada	7763						
11	Barb	Brazil	11958						

Note that in the above formula, I have simply added the two conditions by using the addition operator. Since each of these conditions returns an array of TRUEs and FALSEs, I can add to get a combined array where it's TRUE if any one of the conditions is met.

Another example could be when you want to filter all the records where either the country is the US or the sale value is more than 10000.

The below formula will do this:

```
=FILTER($A$2:$C$11,($B$2:$B$11="US")+(C2:C11>10000))
```

Note: When using AND criteria in a FILTER function, use the multiplication operator (*) and when using the OR criteria, use the addition operator (+).

Example 5: Filtering Data To Get Above/Below Average Records




You can use formulas within the FILTER function to filter and extract records where the value is above or below the average.

For example, suppose you have the dataset as shown below and you want to filter all the records where the sale value is above average.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

You can do that using the following formula:

```
=FILTER($A$2:$C$11,C2:C11>AVERAGE(C2:C11))
```

E2		:	  		=FILTER(\$A\$2:\$C\$11,C2:C11>AVERAGE(C2:C11))			
	A	B	C	D	E	F	G	H
1	Name	Region	Sales		Name	Region	Sales	
2	Rob	US	14020		Rob	US	14020	
3	Alice	Canada	18385		Alice	Canada	18385	
4	Greg	US	19534		Greg	US	19534	
5	Sheen	US	11552					
6	Amy	Asia	11932					
7	Jenny	US	5306					
8	Bob	Asia	8815					
9	Stan	Asia	10396					
10	Lisa	Canada	7763					
11	Barb	Brazil	11958					

Similarly, for below-average, you can use the below formula:

=FILTER(\$A\$2:\$C\$11,C2:C11<AVERAGE(C2:C11))

Example 6: Filtering Only the EVEN Number Records (Or ODD Number Records)

In case you need to quickly filter and extract all the records from even-number rows or odd-number rows, you can do that with the FILTER function.

To do this, you need to check the row number within the FILTER function, and only filter row numbers that meet the row number criteria.

Suppose you have the dataset as shown below and I only want to extract even-numbered records from this dataset.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

Excel for Data Analysts

Below is the formula that will do this:

`=FILTER(A2:C11,MOD(ROW(A2:A11)-1,2)=0)`

E2								<code>=FILTER(\$A\$2:\$C\$11,MOD(ROW(A2:A11)-1,2)=0)</code>
	A	B	C	D	E	F	G	H
1	Name	Region	Sales		Name	Region	Sales	
2	Rob	US	14020		Alice	Canada	18385	
3	Alice	Canada	18385		Sheen	US	11552	
4	Greg	US	19534		Jenny	US	5306	
5	Sheen	US	11552		Stan	Asia	10396	
6	Amy	Asia	11932		Barb	Brazil	11958	
7	Jenny	US	5306					
8	Bob	Asia	8815					
9	Stan	Asia	10396					
10	Lisa	Canada	7763					
11	Barb	Brazil	11958					

The above formula uses the MOD function to check the row number of each record (which is given by the ROW function).

The formula `MOD(ROW(A2:A11)-1,2)=0` returns TRUE when the row number is even and FALSE when it's odd. Note that I have subtracted 1 from the `ROW(A2:A11)` part as the first record is in the second row, and this adjusts the row number to consider the second row as the first record.

Similarly, you can filter all the odd-numbered records using the below formula:

`=FILTER(A2:C11,MOD(ROW(A2:A11)-1,2)=1)`

Example 7: Sort the Filtered the Data With Formula

Using FILTER function with other functions allows us to get a lot more done.

For example, if you filter a dataset using the FILTER function, you can use the SORT function with it to get the result that is already sorted.

Suppose you have a dataset as shown below and you want to filter all the records where the sales value is more than 10000. You can use the SORT function with the function to make sure the resulting data is sorted based on the sales value.

	A	B	C
1	Name	Region	Sales
2	Rob	US	14020
3	Alice	Canada	18385
4	Greg	US	19534
5	Sheen	US	11552
6	Amy	Asia	11932
7	Jenny	US	5306
8	Bob	Asia	8815
9	Stan	Asia	10396
10	Lisa	Canada	7763
11	Barb	Brazil	11958

The below formula will do this:

=SORT(FILTER(\$A\$2:\$C\$11,(\$C\$2:\$C\$11>10000)),3,-1)

E2

⋮

✕

✓

fx

=SORT(FILTER(\$A\$2:\$C\$11,(\$C\$2:\$C\$11>10000)),3,-1)

	A	B	C	D	E	F	G	H	I
1	Name	Region	Sales		Name	Region	Sales		
2	Rob	US	11020		Greg	US	19534		
3	Alice	Canada	18385		Alice	Canada	18385		
4	Greg	US	19534		Sheen	US	14552		
5	Sheen	US	14552		Barb	Brazil	11958		
6	Amy	Asia	11932		Amy	Asia	11932		
7	Jenny	US	5306		Rob	US	11020		
8	Bob	Asia	8815		Stan	Asia	10396		
9	Stan	Asia	10396						
10	Lisa	Canada	7763						
11	Barb	Brazil	11958						

The above function uses the FILTER function to get the data where the sale value in column C is more than 10000. This array returned by the FILTER function is then used within the SORT function to sort this data based on the sales value.

The second argument in the SORT function is 3, which is to sort based on the third column. And the fourth argument is -1 which is to sort this data in descending order.

So these are 7 examples to use the FILTER function in Excel.

6. Sequence Function in Excel

Sequencing your data in an Excel worksheet is a tedious task. But fortunately, those days are long gone! Though many users may think of using the AutoFill feature to list down the numbers series in a sequence. But this modern autofill feature doesn't work if you have a more specified sequential task to perform. Therefore, in such cases, you can use the inbuilt SEQUENCE function, which is specially created for this pursuit.

In this tutorial, you will learn what is Sequence function, its syntax, parameter, points to remember, how this formula works and various examples by using the Sequence function to return a new dynamic array by auto-generating a series of Roman numbers and random integers

What is Sequence Function?

"The SEQUENCE function is an inbuilt Excel function that generates an array of sequential numbers whether number, roman number, alphabets. For example: 1, 2, 3, etc."

The SEQUENCE function is a new dynamic array function that generates a list of sequential numbers in an array. Though the resultant array can be one-dimensional or two-dimensional, determined by the specified parameters i.e., rows and columns. This function always returns a dynamic array that arranging the rows and columns in the specified sequence.

The SEQUENCE function is introduced in the new Microsoft Excel 365. Therefore, if you are using the previous Excel versions, this function won't be available. To use this function, make sure to update your Excel version to Excel 365 or above.

NOTE: If you are trying to access the Sequence function in previous Excel versions, unlike Excel 2019, Excel 2016, and lower, you won't find it.

Syntax

SEQUENCE(rows, [columns], [start], [step])

Parameters

Rows (required) - This parameter represents the number of rows to sequence your data.

Columns (optional) - This parameter represents the number of columns to fill. If the user doesn't provide this argument, by default it takes 1.

Start (optional) - This parameter represents the starting number in the sequence. If the user doesn't provide this argument, by default it takes 1.

Step (optional) - This parameter represents the increment or decrement for each subsequent value in the sequence. It can take the following values:

- If you provide a positive number, the start number gets added in the subsequent value, producing an ascending sequential order.
- If you provide a negative number, the start number gets subtracted from the subsequent values decrease, creating a sequential order.
- If you don't provide any value to this parameter, by default it takes 1.

Return Type

The Excel Sequence Function returns a dynamic array by automatically arranging the rows and columns in the specified sequence.

Formula to create a number sequence in Excel

If you want to create a column of rows with sequential numbers starting at 1, you can use the Excel SEQUENCE function in its simplest form:

To place values in the first **row**:

=SEQUENCE(1)

To put value in a **column**:

SEQUENCE(1, n)

Where n represents the number of values available in the sequence.

For instance, to create a column with 100 numbers, enter the following formula in the first cell (we have taken B2 in our case) and press the Enter button:

=SEQUENCE(100)

The output will be automatically be spilled in the rows starting from the selected row.

Things to Remember about Sequence Function

Before working on Microsoft Excel Sequence Function, make sure to go through with the below facts:

- ***The SEQUENCE function is only available in the Microsoft Excel 365*** or the newer versions. Therefore, if you are using the previous Excel versions (2019, 2018, etc), this function won't be available and it won't create dynamic arrays.
- If the array of sequential numbers is the final output, Excel aligns all the numbers automatically in a spill range. So, ***always make sure you provide sufficient empty cells*** (using the rows and columns parameter) to the down and right of the cell where you enter the formula; otherwise, Excel will throw a #SPILL error.
- ***The returned array can be 1-dimensional or 2-dimensional*** depending upon how you pass the rows and parameters in the Sequence function.
- If any ***optional parameter is omitted, by default, its value is 1.***

Examples:

#SEQUENCE Example1: How to create a number sequence in descending order in Excel

In the below sheet, as you see we have a list of values organised in a random order.

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

Excel for Data Analysts

As asked in the question, to arrange the values in their descending order using the Sequence function, we just need to do a little trick, i.e., we will supply negative one (-1) in the step parameter.


Follow the below-given steps to fetch the descending sequential series of the above-given numbers using the Excel Sequence() function:

Step 1: Select a cell to position your Sequenced data

Place your mouse cursor to a cell from where you want to start the sequencing of your data. In our case, we have selected cell C2 of our Excel worksheet.

Refer to the given below image:

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



Select the cell



STEP 2: Type the SEQUENCE function

In the selected cell, we will start your function with the equal to (=) sign followed by the Sequence function. The formula will be as follows: **=SEQUENCE(**

C2	✖	✔	<i>fx</i>	=SEQUENCE(
	A	B	C	D
1	SEQUENCE		Descending Order	
2		10	=SEQUENCE(
3		4	SEQUENCE (rows, [columns], [start], [step])	
4		6		
5		8		
6		1		
7		3		
8		2		
9		5		
10		7		
11		9		
12				

STEP 3: Insert all the parameters

- At first, this function will ask you to specify the ROWS parameter. We will specify the number to fill and sequence the values. Since we have 10 values so here we will mention 10. The formula will be **=SEQUENCE(10,**
- The next argument is Column. This parameter represents the Column to start the sequence here we will specify 1. **=SEQUENCE(10,1,**
- Next, specify the start parameter. In this, we will specify the value we want to start our sequencing with. Here we will mention 10. **=SEQUENCE(10,1,10,**
- Since we want the sequencing to be ordered in descending order. Therefore in the step parameter, we will specify -1. **=SEQUENCE(10,1,10,-1)**

C2		  <i>fx</i>	=SEQUENCE(10,1,10,-1)
	A	B	C
1	SEQUENCE		Descending Order
2		10	=SEQUENCE(10,1,10,-1)
3		4	
4		6	
5		8	
6		1	
7		3	
8		2	
9		5	
10		7	
11		9	
12			

Step 4: The Sequence function will return the output

As a result, the SEQUENCE function will return the 10 numbers in descending sequential order.

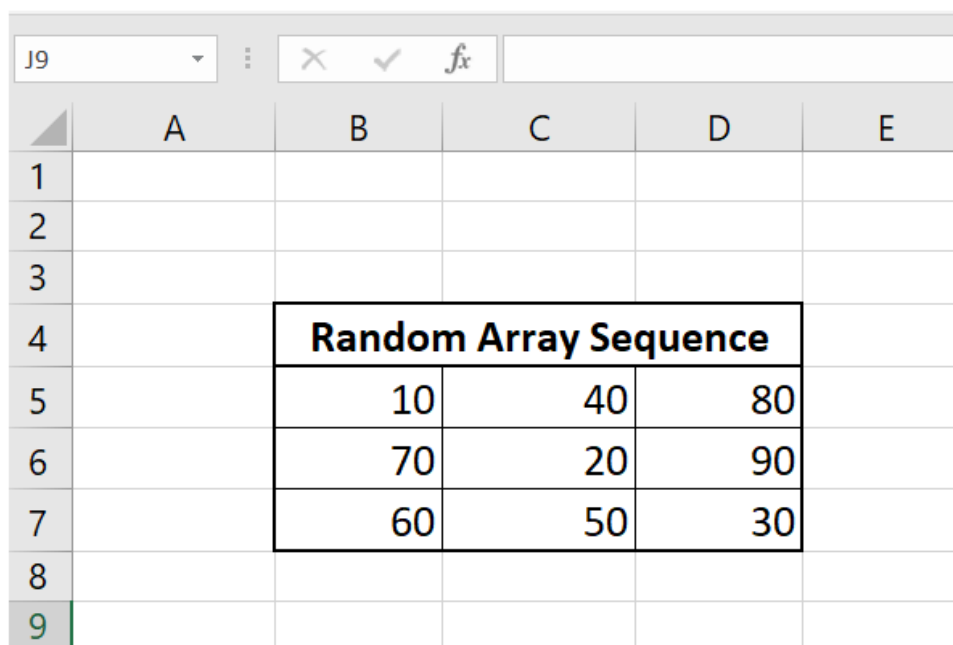
E9	✕	✓	fx	
	A	B	C	D
1	SEQUENCE		Descending Order	
2	10		10	
3	4		9	
4	6		8	
5	8		7	
6	1		6	
7	3		5	
8	2		4	
9	5		3	
10	7		2	
11	9		1	
12				

OUTPUT

This function will select 10 values and start the counting from 10. It will subtract 1 from 10, and return 9. Again, it will subtract 1 and return 8, and in the pattern, it will return 10 values in the descending sequential order.

Example2: How to create a 2-D array where sequential order moves vertically top to bottom.

When working with an array of cells, the standard sequencing rule goes horizontally across the first row till its last element, and next, it moves down to the next row, in a similar method to reading text from left to right. But in various scenarios, the range of cells is not listed in sequential order as per the given Excel data.



	A	B	C	D	E
1					
2					
3					
4		Random Array Sequence			
5		10	40	80	
6		70	20	90	
7		60	50	30	
8					
9					

To propagate the above data in the top to bottom order across the first column and then right to the next column, we will use the combination of the SEQUENCE function and the TRANSPOSE function. Follow the below-given steps to fetch the results using Excel SEQUENCE() function:

Step 1: Select a cell to position your Sequenced data

Place your mouse cursor to a cell from where you want to start the sequencing of your data. In our case, we have selected cell F2 of our Excel worksheet.

Refer to the given below image:

	A	B	C	D	E	F	G	H	I
2									
3									
4		Random Array Sequence				Sequenced Array			
5		10	40	80					
6		70	20	90					
7		60	50	30					
8									

STEP 2: Type the SEQUENCE function

In the selected cell, we will start your function with the equal to (=) sign followed by the Sequence function. The formula will be as follows: **=SEQUENCE(**

	A	B	C	D	E	F	G	H	I
2									
3									
4		Random Array Sequence				Sequenced Array			
5		10	40	80		=SEQUENCE(
6		70	20	90					
7		60	50	30					
8									

STEP 3: Insert all the parameters

- At first, this function will ask you to specify the ROWS parameter. Since in our case we have our data in three rows. Therefore, we will specify 3 in this parameter. The formula will be **=SEQUENCE(3,**
- The next argument is Column. This parameter represents the Column to start the sequence. Since we have 3 columns, therefore will specify 3. The formula will be **=SEQUENCE(3,3,**
- Next, specify the start parameter. In this, we will specify the value we want to start our sequencing with. Here we will mention 10. The formula will be **=SEQUENCE(3,3,10**

- Since we want the sequencing to be ordered with a gap of 10 order. Therefore, in the step parameter, we will specify 10. =
SEQUENCE(3,3,10,10)

	A	B	C	D	E	F	G	H
2								
3								
4		Random Array Sequence				Sequenced Array		
5		10	40	80		=SEQUENCE(3,3,10,10)		
6		70	20	90				
7		60	50	30				
8								

Step 4: The Sequence function will return the output

As a result, the regular SEQUENCE function will moves horizontally left to right (column-wise) and return the following maintaining a gap of 10 between the values.

	A	B	C	D	E	F	G	H	I
2									
3									
4		Random Array Sequence				Sequenced Array			
5		10	40	80		10	20	30	
6		70	20	90		40	50	60	
7		60	50	30		70	80	90	
8									

Step 5: Nest Sequence function inside the Transpose Function

Though the above output is also correct, that's not what we were looking for. In the question, we are asked to create a sequence that moves vertically from top to bottom (row-wise), maintaining order of 10. Therefore, we will use the Transpose function and nest the above Sequence function inside it.

The formula will be as follows:

=TRANSPOSE(SEQUENCE(3, 3, 10, 10))

Excel for Data Analysts

F5									
	A	B	C	D	E	F	G	H	I
2									
3									
4		Random Array Sequence				Sequenced Array			
5		10	40	80		=TRANPOSE(SEQUENCE(3,3,10,10))			
6		70	20	90					
7		60	50	30					
8									
9									

Step 6: Excel will fetch you the result

Once done, press the enter button and Excel will arrange all the data in the specified sequence where the is arranged in the array sequencing vertically from top to bottom (row-wise).

Refer to the below image for your reference.

F4								
	A	B	C	D	E	F	G	H
2								
3								
4		Random Array Sequence				Sequenced Array		
5		10	40	80		10	40	70
6		70	20	90		20	50	80
7		60	50	30		30	60	90
8								
9								



Output Sequencing Vertically from Top to Bottom

7. Excel String Functions – LEFT, RIGHT, and MID

Excel is mostly about numerical, however, we often have data of text data type. Here are a few functions we should know to handle text data.

- LEFT
- RIGHT
- MID

The LEFT function

The LEFT function returns a given text from the left of our text string based on the number of characters specified.

Syntax:

LEFT(text, [num_chars])

Parameters:

- **Text:** The text we want to extract from.
- **Num_chars** (Optional): The number of characters you want to extract. Default num_chars is 1 and the number must be a positive number that is greater than zero.
- **Example:**

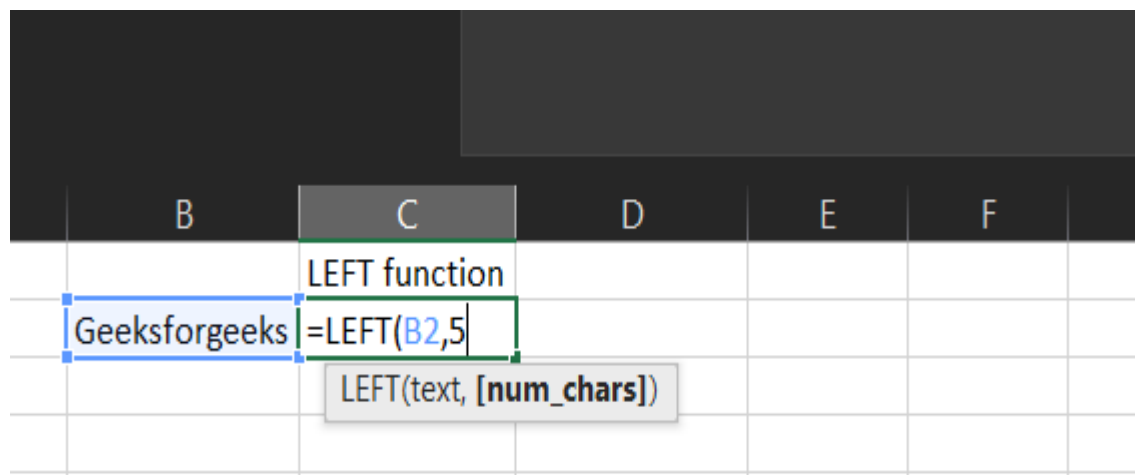
Step 1: Format your data.

Now if you want to get the first “Geeks” from “Geeksforgeeks” in B2. Let us follow the next step.

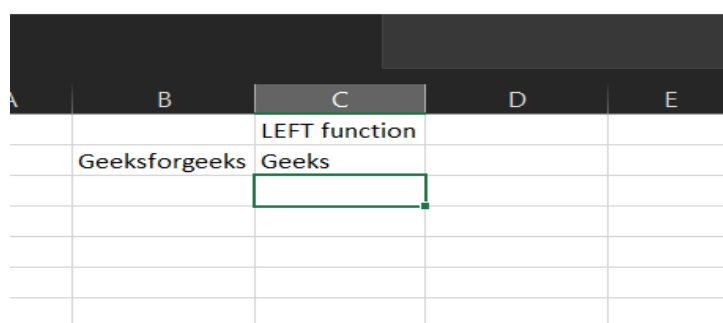
B	C	D	E	F	
	LEFT function				
Geeksforgeeks					

Excel for Data Analysts

Step 2: We will enter =LEFT(B2,5) in the B3 cell. Here we want Excel to extract the first 5 characters from the left side of our text in B2.



This will return "Geeks".



The RIGHT Function

The RIGHT function returns a given text from the right of our text string based on the number of characters specified.

Syntax:

RIGHT(text, [num_chars])

Parameters:

- **Text:** The text we want to extract from.
- **Num_chars** (Optional): The number of characters you want to extract. Default num_chars is 1 and the number must be a positive number that is greater than zero.

Excel for Data Analysts

Example:

Step 1: Format your data.

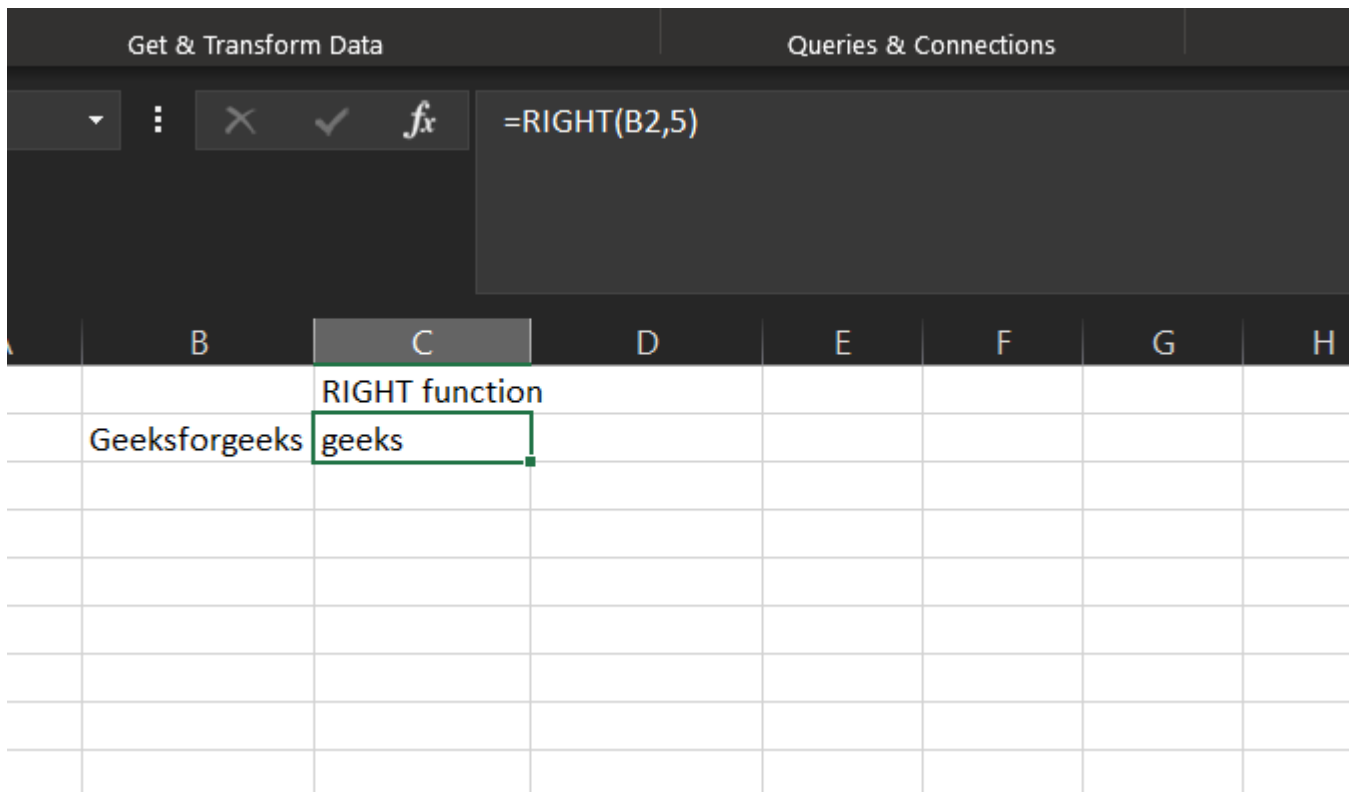
Now if you want to get the last geeks from “Geeksforgeeks” in B2. Let us follow the next step.

A	B	C	D	E
		RIGHT function		
	Geeksforgeeks			

Step 2: We will enter =RIGHT(B2,5) in the B3 cell. Here we want Excel to extract the last 5 characters from the right side of our text in B2.

A	B	C	D	E	F
		RIGHT function			
	Geeksforgeeks	=RIGHT(B2,5)			
		RIGHT(text, [num_chars])			

This will return “geeks”.



The MID function

The MID function returns the text from any middle part of our text string based on the starting position and the number of characters specified.

Syntax:

`MID(text, start_num, num_chars)`

Parameters:

- **Text:** The text we want to extract from.
- **start_num:** The starting number of the first character from the text we want to extract.
- **Num_chars:** The number of characters you want to extract.

Excel for Data Analysts

Example:

Step 1: Format your data.

A	B	C	D	E	F	G	
		MID function					
	Geeksforgeeks						

Now if you want to get the character “for” which is located in the middle of our text “Geeksforgeeks” in B2. Let us follow the next step.

Step 2: We will enter =MID(B2,5,3) in the B3 cell. Here we want Excel to extract the characters located in the middle of our text in B2.

A	B	C	D	E	F	G	H
		MID function					
	Geeksforgeeks	=MID(B2,6,3)					
		MID(text, start_num, num_chars)					

This will return “for”.

[illegible]

8. Excel SMALL and LARGE functions

Want to pick out the smallest or the largest numbers from your dataset? But don't want to reposition it?

The SMALL and LARGE functions of Excel will help you do that without reshaping your data.

I will walk you through both these functions using multiple examples. So don't wait any further – and dive right in.

How to use the SMALL function

The SMALL function (just like you guessed it) will extract the **k-th** smallest value from any given dataset

It has two required arguments:

- **Array** – the array of range values from where the smallest value is sought
- **K** – the k-th smallest value that is sought

I know it's always boring to keep reading about functions. So how about an example

The holiday season is yet not over, and we have a list of airlines with their ticket prices mentioned next to them.

	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			

Let's find the 4th cheapest ticket from the list above

To help you double-check the accuracy of the function, we have organized the data in ascending order. This is not needed to run the SMALL otherwise

1. Write the SMALL function as below:

= SMALL (

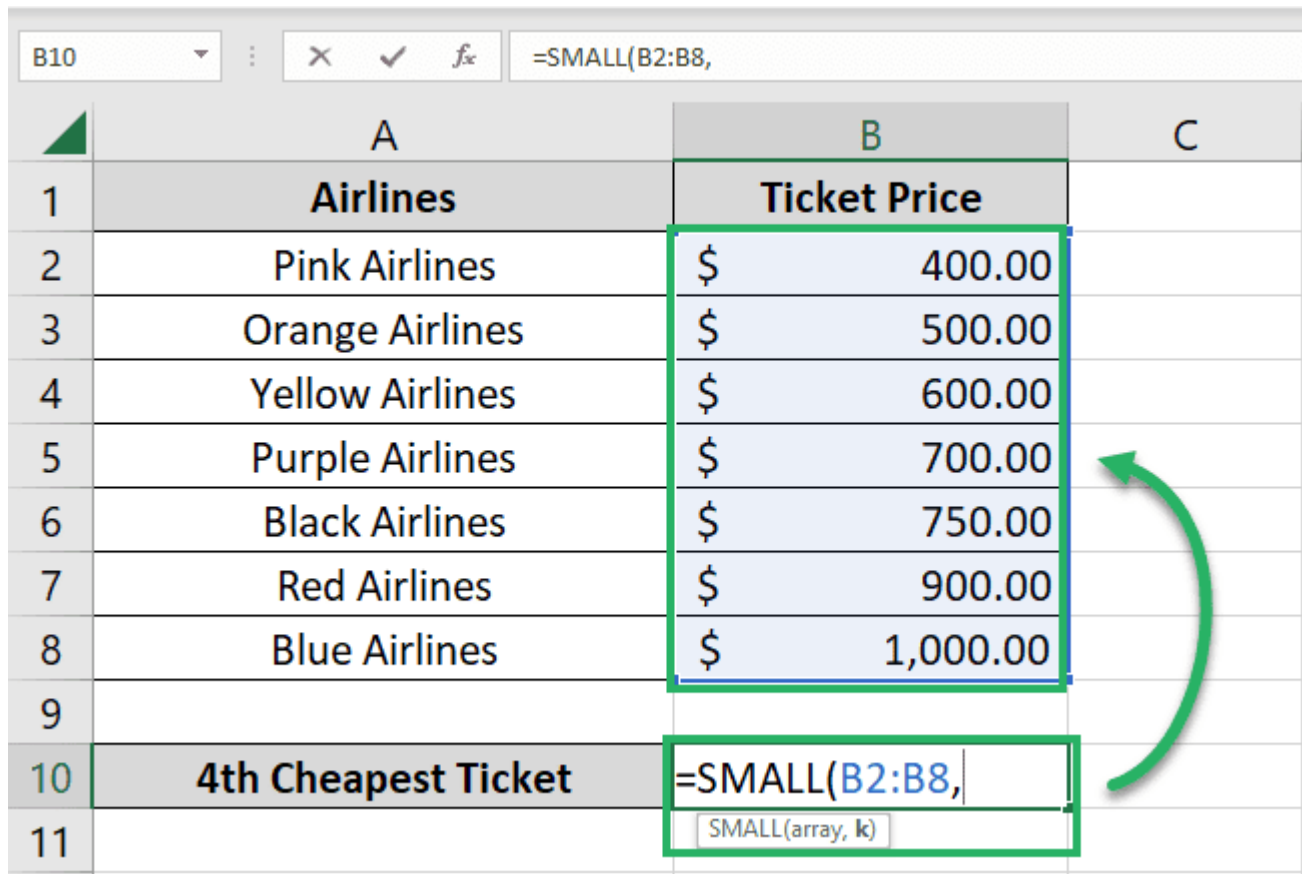
	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	4th Cheapest Ticket	=SMALL(
11		SMALL(array, k)	

Excel for Data Analysts

- For the first argument (array), create a reference to the cell range containing the numeric values (from where the smallest value is sought).

We are creating a reference to the cell range B2:B8.

= SMALL (B2:B8



	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	4th Cheapest Ticket	=SMALL(B2:B8,	
11		SMALL(array, k)	

- For argument K, write the position of the lowest value.

We want to find the **4th cheapest ticket**. In other words, we seek the 4th smallest value from the dataset above.

So our function becomes:

= SMALL (B2:B8, 4)

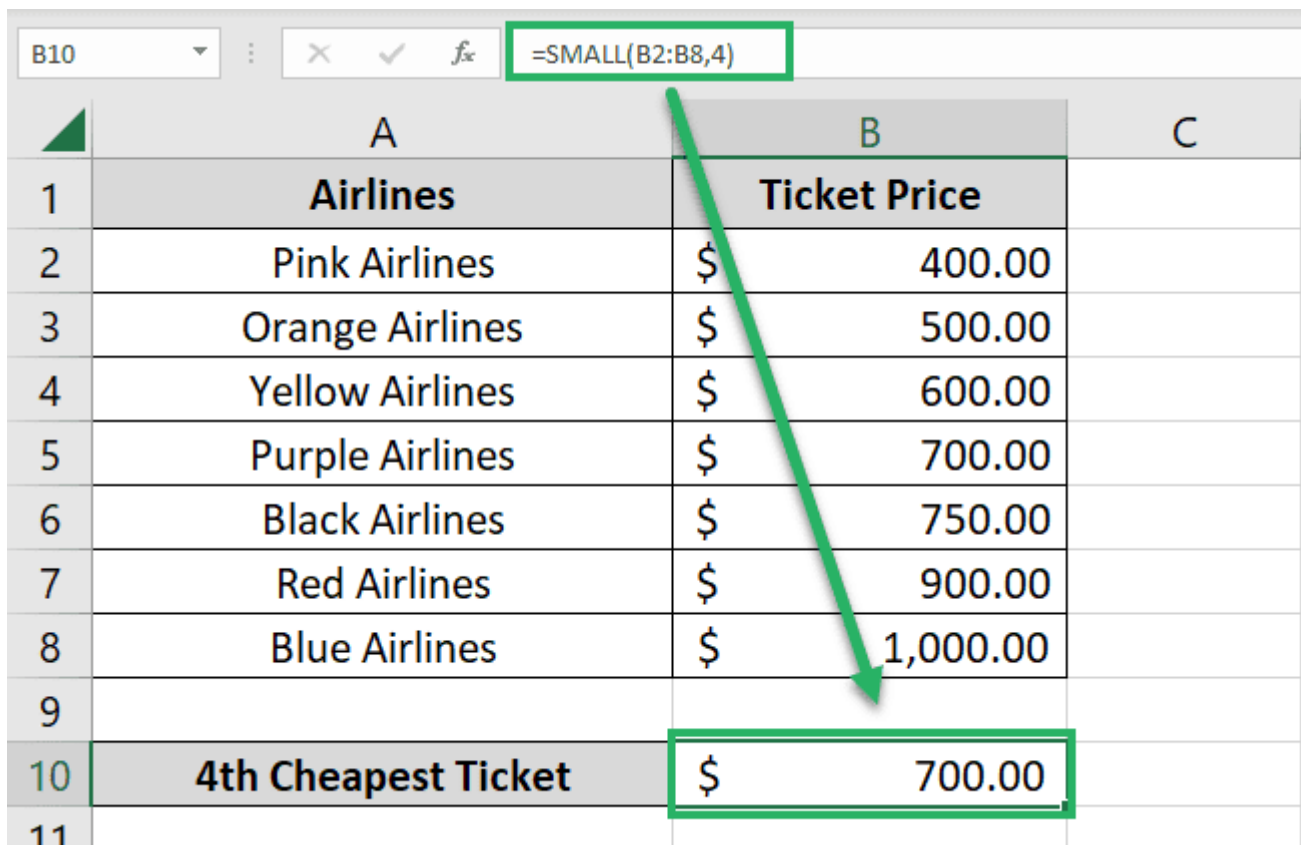
B10 X ✓ fx =SMALL(B2:B8,4)			
	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	4th Cheapest Ticket	=SMALL(B2:B8,4)	
11			

Pro Tip!

Be careful while you write the k argument. If the value for k is a number smaller than 1 or a number greater than the number of values in your dataset, the SMALL function will return the #NUM error or the wrong value.

For example, the data above has a total of 7 numerical values. If you write the k argument as 8 (or any number greater than 7), you'll end up getting the #NUM error

4. You're only an 'Enter' away from the desired results. So go on and hit it.



	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	4th Cheapest Ticket	\$ 700.00	
11			

That's \$700 by Purple Airlines. Impressive to see how the SMALL function works, isn't it

The SMALL function is available in all versions of Microsoft Excel starting from Excel 2010 to Excel 365.

How to use the LARGE function

If you thought the LARGE function would be a dupe of the SMALL function – you made a bang-on target

The Excel LARGE function will extract the **k-th** largest value from any given dataset.

It also has the same two arguments:

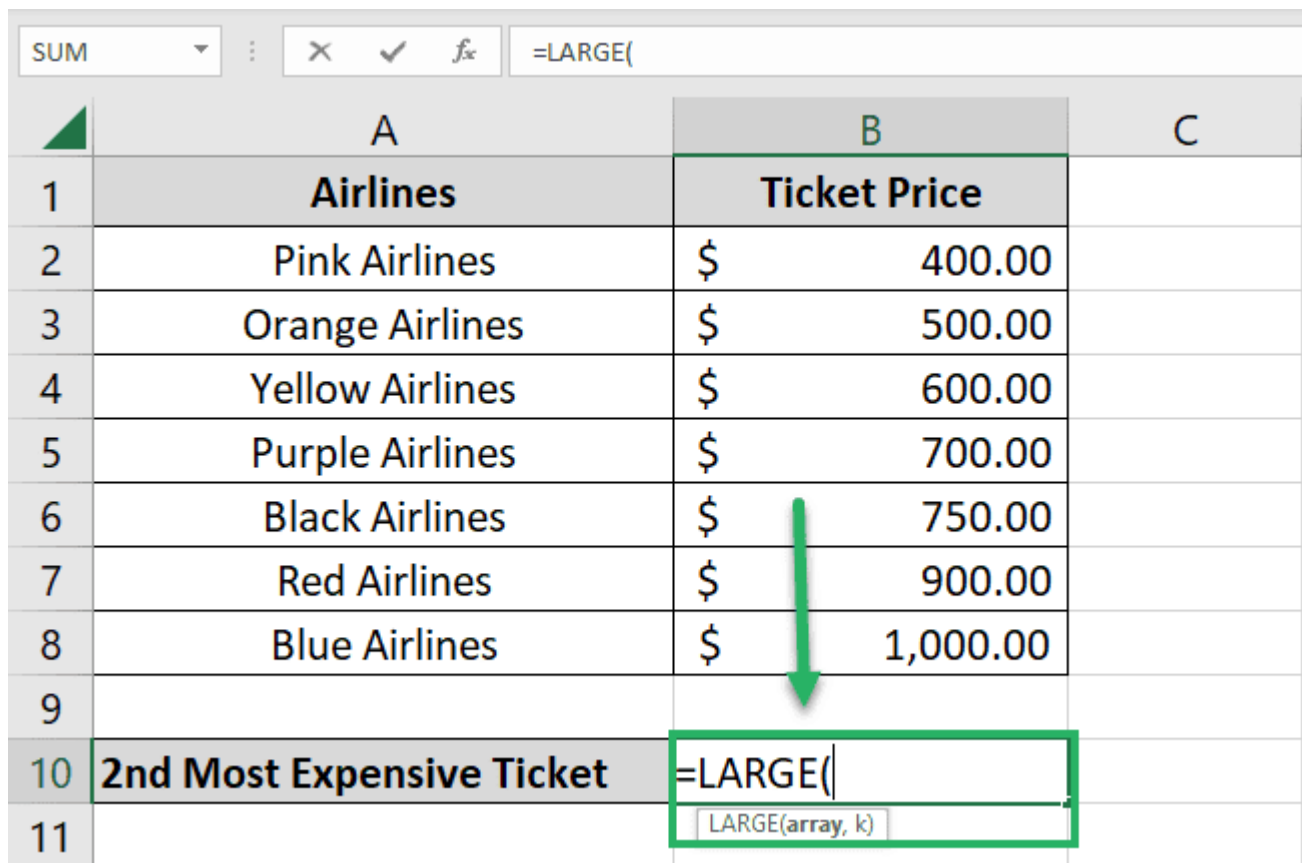
- **Array** – the array of range values from where the largest value is sought
- **K** – the k-th largest value that is sought

Excel for Data Analysts

Let's not go any far and try using the LARGE function on the same example, as above. However, this time, we are on the hunt for the **second-highest ticket price**

1. Write the LARGE function as below:

= LARGE (




	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	2nd Most Expensive Ticket	=LARGE(
11			

2. For the first argument (array), create a reference to the cell range containing the values (from where the largest value is sought).

We are again creating a reference to the cell range B2:B8.

= LARGE (B2:B8

B10 X ✓ fx =LARGE(B2:B8,			
	A	B	C
1	Airlines	Ticket Price	
2	Pink Airlines	\$ 400.00	
3	Orange Airlines	\$ 500.00	
4	Yellow Airlines	\$ 600.00	
5	Purple Airlines	\$ 700.00	
6	Black Airlines	\$ 750.00	
7	Red Airlines	\$ 900.00	
8	Blue Airlines	\$ 1,000.00	
9			
10	2nd Most Expensive Ticket	=LARGE(B2:B8,	
11		LARGE(array, k)	



3. For argument K, write the position of the largest value.

We want to find the 2nd most expensive ticket (the 2nd largest value) from the dataset above.

So our function becomes:

= LARGE (B2:B8, 2)

B10				=LARGE(B2:B8,2)
	A	B	C	
1	Airlines	Ticket Price		
2	Pink Airlines	\$ 400.00		
3	Orange Airlines	\$ 500.00		
4	Yellow Airlines	\$ 600.00		
5	Purple Airlines	\$ 700.00		
6	Black Airlines	\$ 750.00		
7	Red Airlines	\$ 900.00		
8	Blue Airlines	\$ 1,000.00		
9				
10	2nd Most Expensive Ticket	=LARGE(B2:B8,2)		
11				

4. Hit Enter to get the results as follows:

B10				<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
-----	--	--	--	---

That's \$900 by Red Airlines

Both the SMALL and the LARGE functions are super simple to use. And you vouch for that too now, don't you?

Pro Tip!

Why were the LARGE and SMALL functions needed when Excel had the MIN and the MAX functions

The MIN and the MAX functions will only return the minimum and the maximum values from a given dataset. However, you don't want the lowest but the second, third (or any k-th) lowest or highest numeric value from the same dataset, these functions would no more help.

That's where you'd need the LARGE and the SMALL function

SMALL and LARGE functions examples

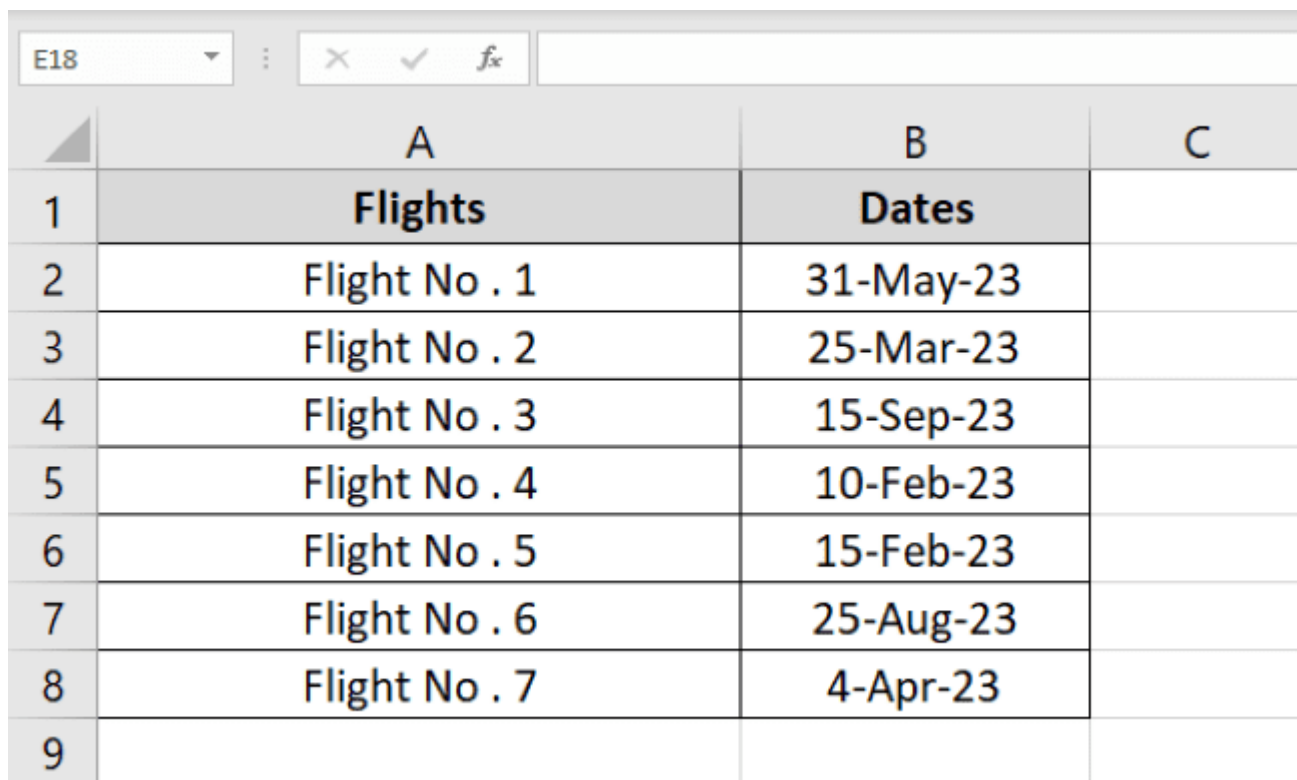
The basic application of the SMALL and the LARGE function is just that simple.

And now, we are diving into more examples for both these functions.

Example #1: SMALL and LARGE function with Dates

Did you know? You can use the SMALL and LARGE functions with dates too

We will show you that here. The images below have a list of flights (all scheduled for different dates).



The screenshot shows an Excel spreadsheet with a table of flights. The table has three columns: A (Flights), B (Dates), and C (empty). The rows are numbered 1 to 9. The data is as follows:

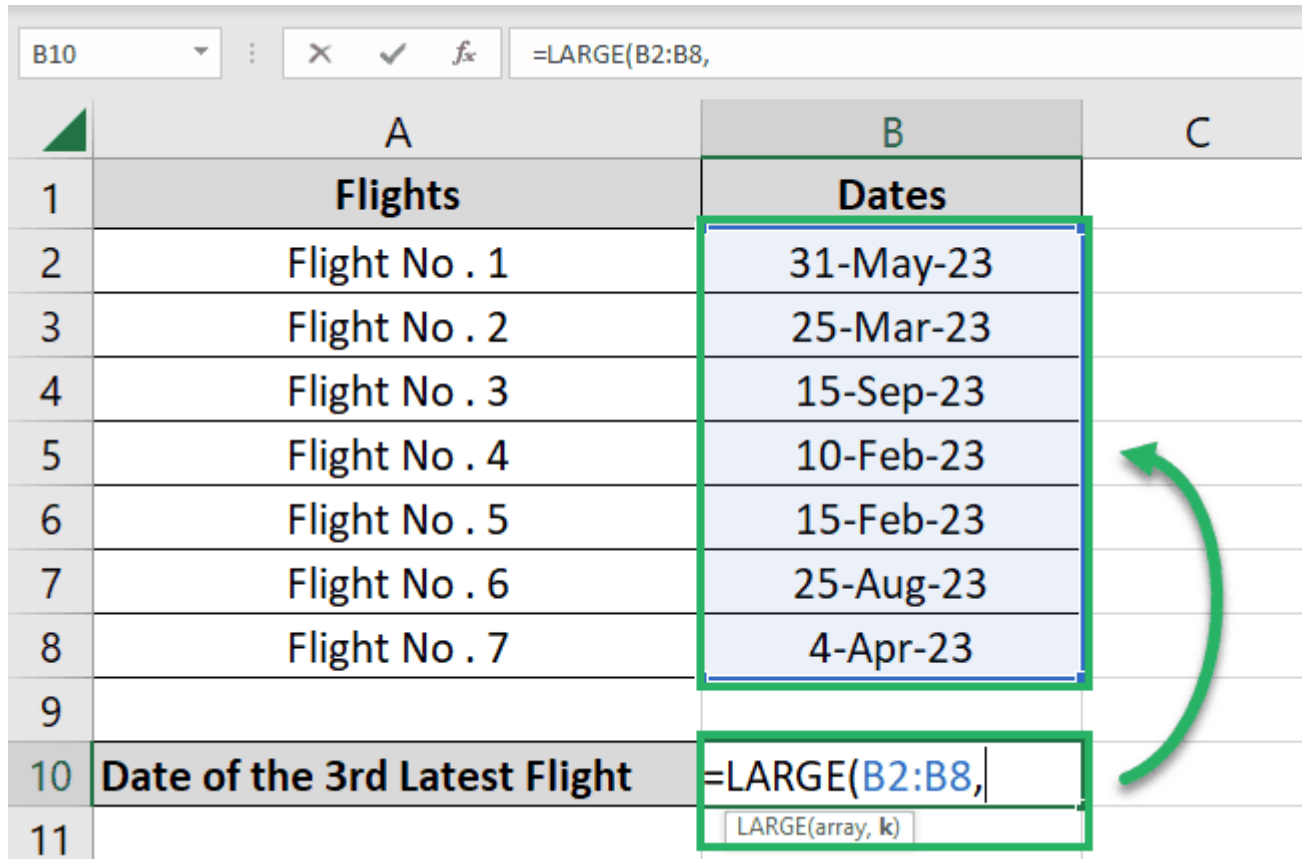
	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			

Excel for Data Analysts

To see **when the third latest flight is scheduled**, let's apply the LARGE function:

1. Write the LARGE function as below:

= LARGE (B2:B8




	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	Date of the 3rd Latest Flight	=LARGE(B2:B8,	
11		LARGE(array, k)	

For the first argument (array), we have referred to the cell range that contains the dates.

2. For argument K, write 3. That's because we want to find the third latest flight.

= LARGE (B2:B8, 3)



3. Hit Enter to get the results as follows

	3
	2
	1
	1
	1
	2

That's on 31st May 2023.

Pro Tip!

Instead of the date (31st May 2023), are you getting a serial number as result? If that's the case, relax

There is no problem with your function. You only need to reformat the subject cell. To do that:

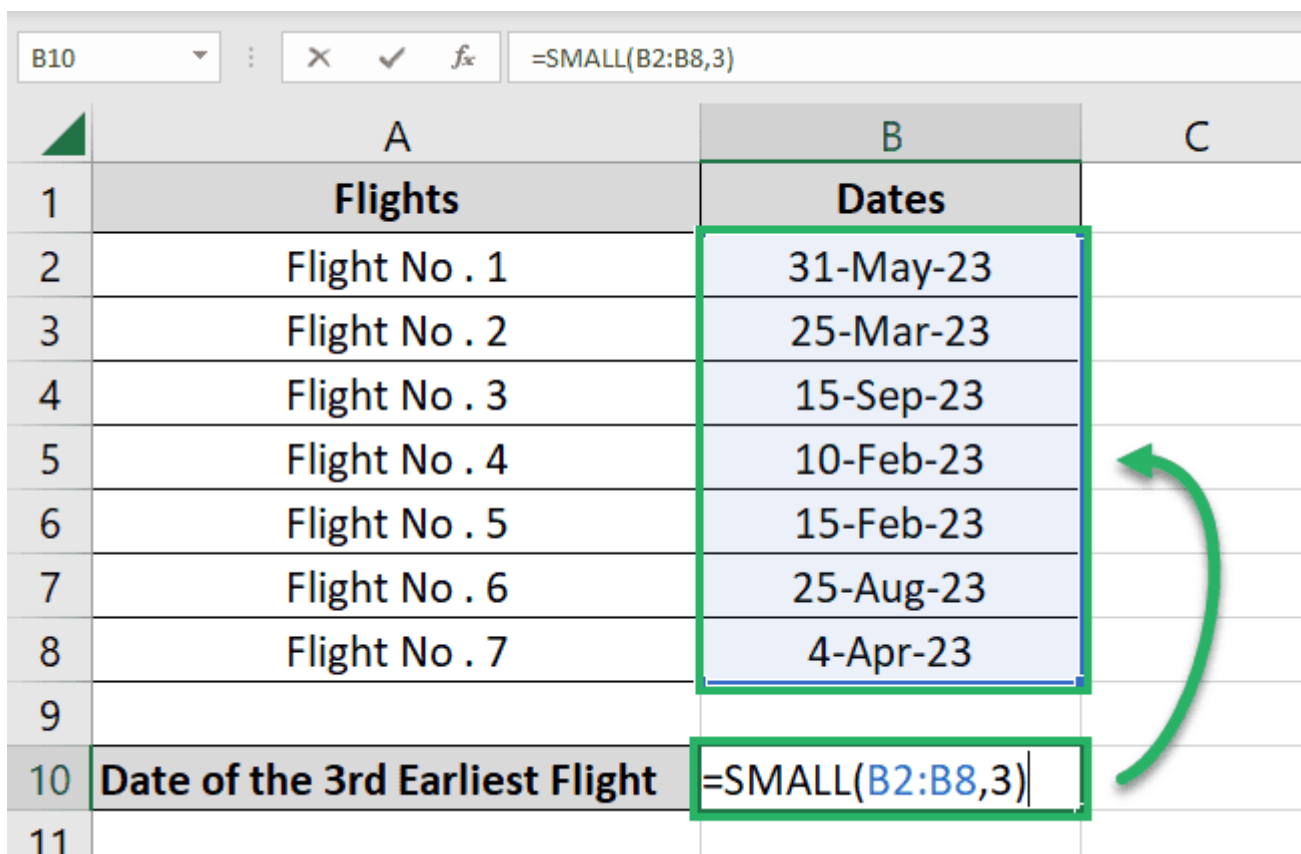
- Select the cell.
- Go to the **Home Tab > Format > Short Date Format**.

And everything will be sorted.

Similarly, if you want to find the **third earliest flight**:

4. Write the SMALL function as follows:

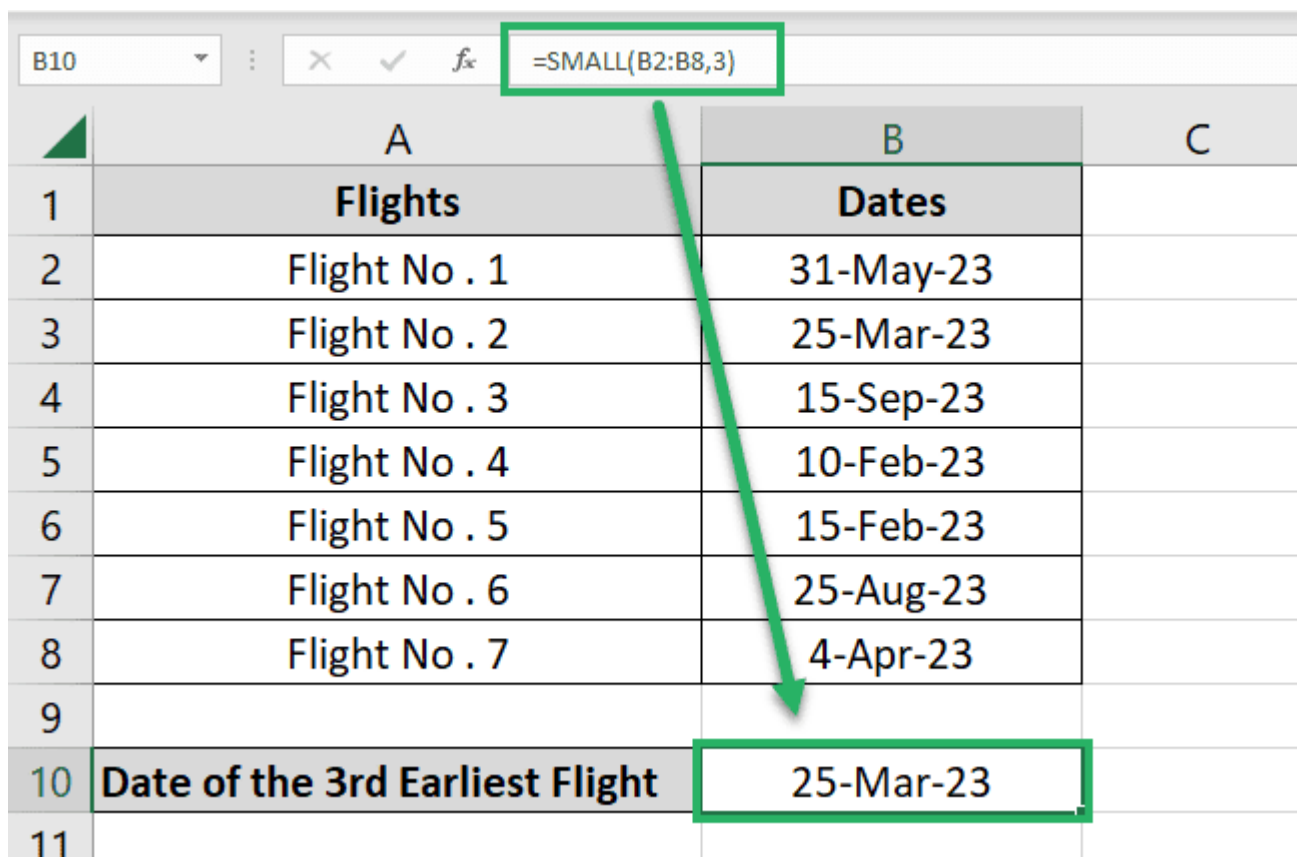
= SMALL (B2:B8, 3)



	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	Date of the 3rd Earliest Flight	=SMALL(B2:B8,3)	
11			

The function above translates as “Find the 3rd smallest date from the cell range B2 to B8”.

5. Hit Enter to get the results.



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	Date of the 3rd Earliest Flight	25-Mar-23	
11			

The formula bar at the top shows the formula `=SMALL(B2:B8,3)` in cell B10. A green arrow points from the formula bar to the result '25-Mar-23' in cell B10.

And there you have it! On 25th March 2023

Example #2: With the XLOOKUP function

In the above example, we have found the third earliest and the third latest dates when the flights are scheduled.

But what if we want the answer to be different? What if you want to fetch Flight No., and not the date when that flight is scheduled

We need to combine the XLOOKUP function with the SMALL and LARGE functions to get that. So let's do it.

1. Write the XLOOKUP function as follows:

= XLOOKUP (

	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	3rd Latest Flight No.	=XLOOKUP(
11		XLOOKUP(lookup_value, lookup_array, return_array, [if_n	

2. As the lookup_value, write in the LARGE function as written above.

= XLOOKUP (LARGE (B2:B8, 3)

	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	3rd Latest Flight No.	=XLOOKUP(LARGE(B2:B8,3)	
11		XLOOKUP(lookup_value, lookup_array, return_array, [if_n	

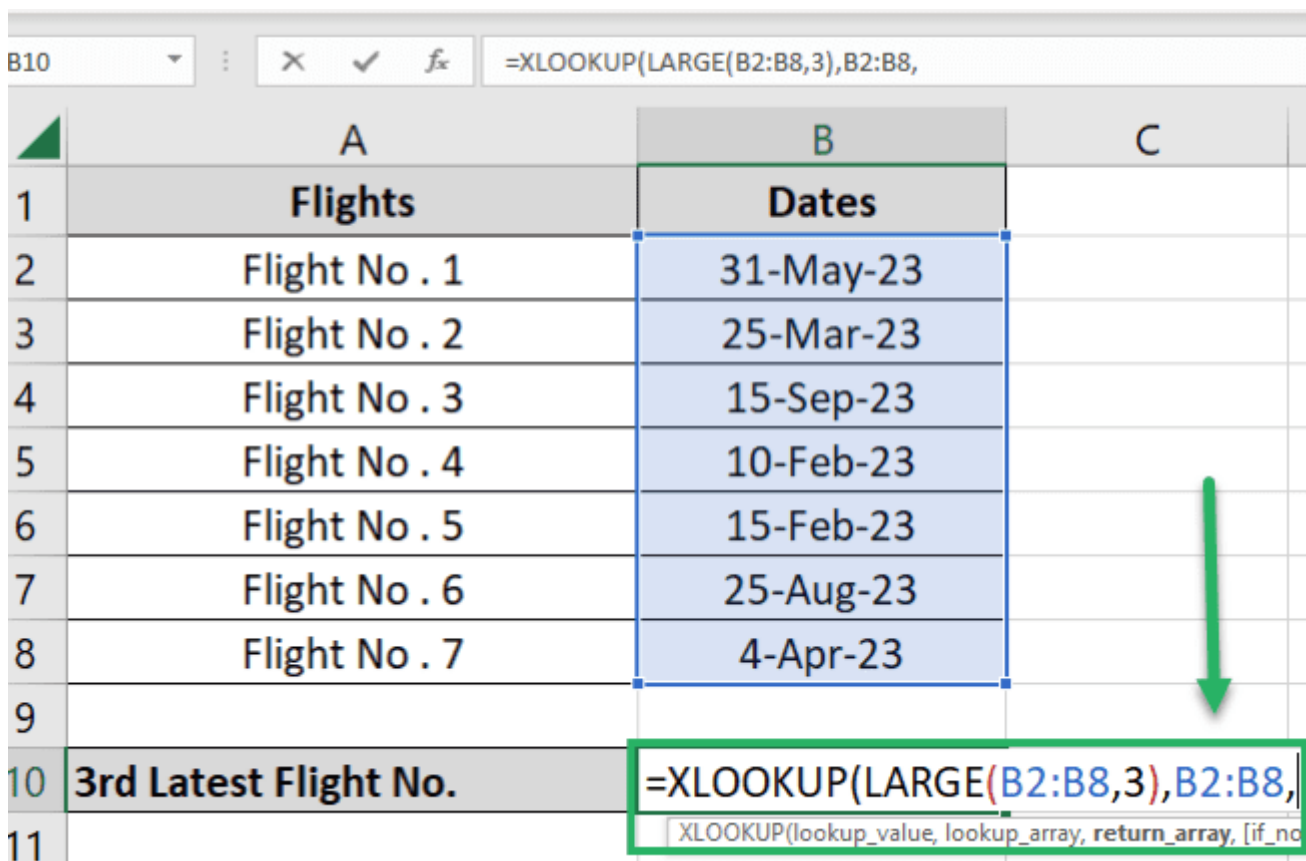
Excel for Data Analysts

We have just told Excel to look for the third largest value from the cell range **B2:B8**.

3. Create a reference to the lookup range (where the lookup value is to be looked for).

The lookup value will be the third latest date, so the lookup range will be the cell range where the dates sit

=XLOOKUP (LARGE (B2:B8, 3), B2:B8



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	3rd Latest Flight No.	=XLOOKUP(LARGE(B2:B8,3),B2:B8,	
11		XLOOKUP(lookup_value, lookup_array, return_array, [if_no	

A green arrow points down from the date range B2:B8 to the formula in cell B10.

4. Create a reference to the return array.

We want the flight number to be returned so, the return array will be the column for flight numbers.

=XLOOKUP (LARGE (B2:B8, 3), B2:B8, A2:A8)

B10				=XLOOKUP(LARGE(B2:B8,3),B2:B8,A2:A8)
	A	B	C	
1	Flights	Dates		
2	Flight No . 1	31-May-23		
3	Flight No . 2	25-Mar-23		
4	Flight No . 3	15-Sep-23		
5	Flight No . 4	10-Feb-23		
6	Flight No . 5	15-Feb-23		
7	Flight No . 6	25-Aug-23		
8	Flight No . 7	4-Apr-23		
9				
10	3rd Latest Flight No.	Flight No . 1		
11				

SUM				=XLOOKUP(LARGE(B2:B8,3),B2:B8,A2:A8)
	A	B	C	D
1	Flights	Dates		
2	Flight No . 1	31-May-23		
3	Flight No . 2	25-Mar-23		
4	Flight No . 3	15-Sep-23		
5	Flight No . 4	10-Feb-23		
6	Flight No . 5	15-Feb-23		
7	Flight No . 6	25-Aug-23		
8	Flight No . 7	4-Apr-23		
9				
10	3rd Latest Flight No.	=XLOOKUP(LARGE(B2:B8,3),B2:B8,A2:A8)		
11				

Excel for Data Analysts

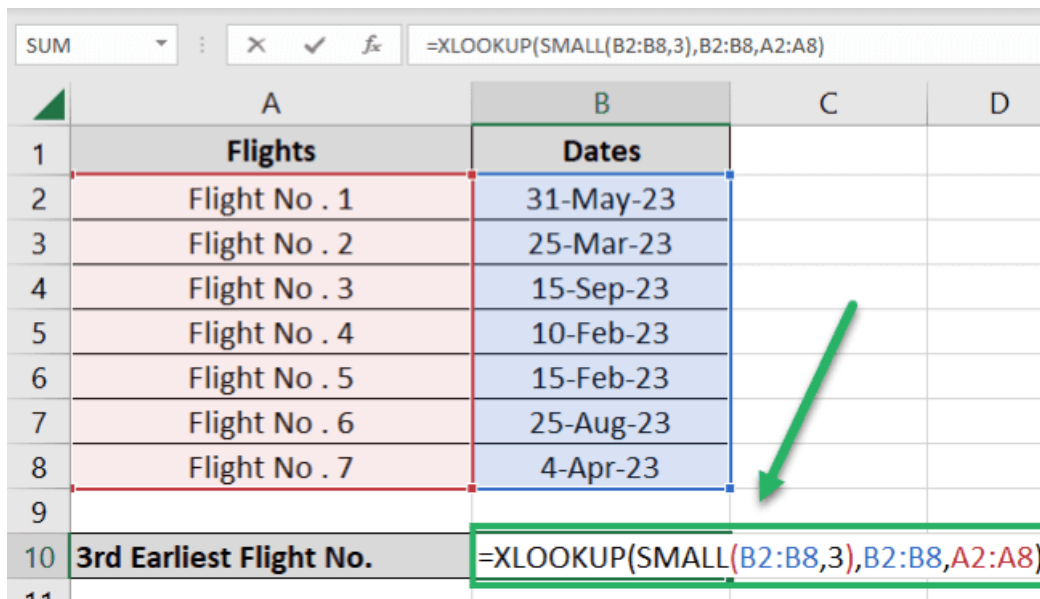
5. Hit Enter.

We get **Flight No. 1**. Check out the date against it. It's 31st May 2023 – the same that we got above

Let's do the same for the third earliest flight. Here you go!

6. Write the XLOOKUP function as follows:

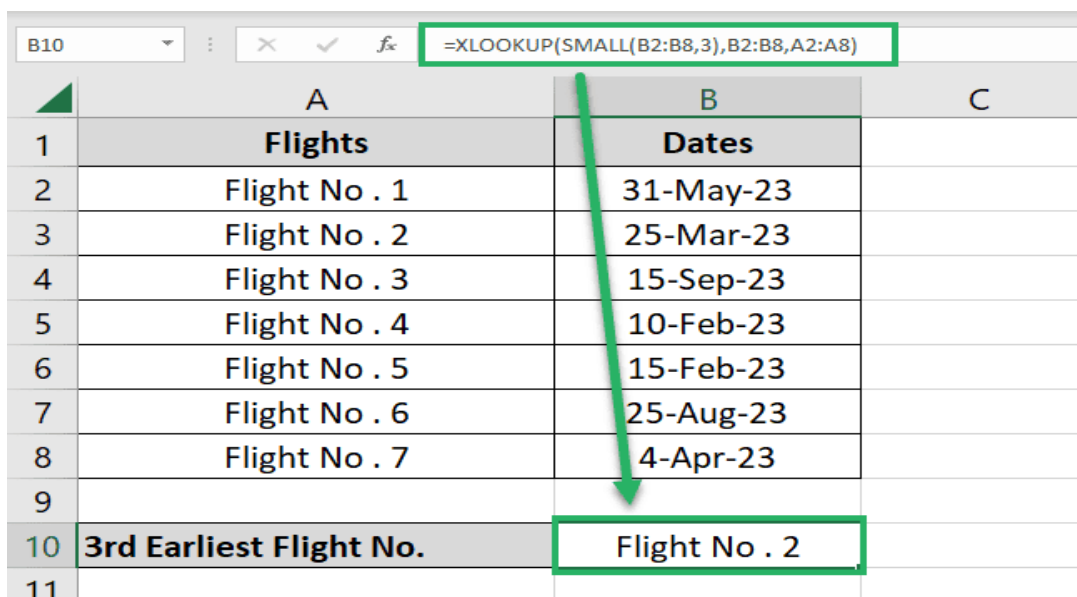
= XLOOKUP (SMALL (B2:B8, 3), B2:B8, A2:A8)



	A	B	C	D
1	Flights	Dates		
2	Flight No . 1	31-May-23		
3	Flight No . 2	25-Mar-23		
4	Flight No . 3	15-Sep-23		
5	Flight No . 4	10-Feb-23		
6	Flight No . 5	15-Feb-23		
7	Flight No . 6	25-Aug-23		
8	Flight No . 7	4-Apr-23		
9				
10	3rd Earliest Flight No.	=XLOOKUP(SMALL(B2:B8,3),B2:B8,A2:A8)		
11				

Take a closer look at the function above. We have only replaced the LARGE function with the SMALL function.

And the results are here.



	A	B	C
1	Flights	Dates	
2	Flight No . 1	31-May-23	
3	Flight No . 2	25-Mar-23	
4	Flight No . 3	15-Sep-23	
5	Flight No . 4	10-Feb-23	
6	Flight No . 5	15-Feb-23	
7	Flight No . 6	25-Aug-23	
8	Flight No . 7	4-Apr-23	
9			
10	3rd Earliest Flight No.	Flight No . 2	
11			

This time it's Flight No. 2 (dated 25th March 2023)

Excel for Data Analysts

The Excel UNIQUE function can extract a list of distinct values, or a list of values that only occur once, i.e. unique values. It can return a unique or distinct list from one column or from multiple columns.

Syntax: =UNIQUE(array, [by_col], [occurs_once])

array is the range or array you want the unique values returned from.

by_col is an optional logical value (TRUE/FALSE) and allows you to compare values by row (FALSE), or by column (TRUE).

occurs_once is also an optional logical value (TRUE/FALSE) and allows you to find the truly unique values, i.e. the values that only occur once (TRUE), or all distinct values (FALSE). If you omit this argument, it will default to FALSE and return a distinct list.

Note: The UNIQUE function is part of the new Excel Dynamic Arrays family and at the time of writing, Dynamic Arrays are only available in Office 365 and are currently in beta on the Insiders channel. Excel 2019 will not have the Dynamic Array functions.

Excel UNIQUE Function Examples

Let's say we want to find a list of distinct values from the Item column (C) in the table below:

	B	C	D	E	F
6	Department	Item	Quantity	Price	Total \$
7	Support	Stamps	50	0.60	30.00
8	Support	Paper	10	4.99	49.90
9	Support	Pens	15	0.95	14.25
10	Support	Staples	1	1.10	1.10
11	Support	Scissors	1	4.70	4.70
12	Front Office	Stamps	250	0.60	150.00
13	Front Office	Paper	30	4.99	149.70
14	Front Office	Pens	50	0.95	47.50
15	Front Office	Staples	10	1.10	11.00
16	Front Office	Scissors	0	4.70	-
17	Sales	Stamps	200	0.60	120.00
18	Sales	Paper	20	4.99	99.80
19	Sales	Pens	25	0.95	23.75
20	Sales	Staples	5	1.10	5.50
21	Sales	Scissors	2	4.70	9.40

Excel for Data Analysts

We can write a simple UNIQUE formula like this:

`=UNIQUE(C7:C21)`

You can see the results in cells B26:B30 in the image below:

The screenshot shows an Excel spreadsheet with a table of office supplies. The formula bar at the top shows `=UNIQUE(C7:C21)` in cell B26. The table has columns: Department, Item, Quantity, Price, and Total \$. The data rows are 7 to 21. The results of the UNIQUE function are shown in cells B26:B30, displaying a list of unique items: Stamps, Paper, Pens, Staples, and Scissors. A dashed orange arrow points from the formula bar to the first result, and another dashed orange arrow points from the formula bar to the list of results.

	B	C	D	E	F
6	Department	Item	Quantity	Price	Total \$
7	Support	Stamps	50	0.60	30.00
8	Support	Paper	10	4.99	49.90
9	Support	Pens	15	0.95	14.25
10	Support	Staples	1	1.10	1.10
11	Support	Scissors	1	4.70	4.70
12	Front Office	Stamps	250	0.60	150.00
13	Front Office	Paper	30	4.99	149.70
14	Front Office	Pens	50	0.95	47.50
15	Front Office	Staples	10	1.10	11.00
16	Front Office	Scissors	0	4.70	-
17	Sales	Stamps	200	0.60	120.00
18	Sales	Paper	20	4.99	99.80
19	Sales	Pens	25	0.95	23.75
20	Sales	Staples	5	1.10	5.50
21	Sales	Scissors	2	4.70	9.40
22	Total				716.60
23					
24					
25	UNIQUE				
26	Stamps				
27	Paper				
28	Pens				
29	Staples				
30	Scissors				

The Excel UNIQUE function ‘spills’ the results into the rows below. This is the new dynamic array behaviour that occurs when the final result of the formula returns multiple values.

Note: In the example above, I’ve omitted the `by_col` argument, which means it will default to FALSE and compare values by row. I’ve also omitted the `occurs_once` argument so it has defaulted to FALSE and returned a distinct list.

Extract a list of Unique Values

	A	B	C	D	E	F
1	Department	Quantity	Price	Total \$		
2	Support	1	4.70	4.70		
3	Front Office	250	0.60	150.00		
4	Front Office	30	4.99	149.70		
5	Front Office	50	0.95	47.50		
6	Front Office	10	1.10	11.00		
7	Front Office	0	4.70	-		
8	Sales	200	0.60	120.00		
9	Sales	20	4.99	99.80		
10						
11	Support					
12						

In the example below, you can see there are whole rows containing duplicates. Setting the `by_col` and `occurs_once` arguments to `FALSE` we get a list of unique rows, as shown below:

	A	B	C	D	E	F
1	Department	Item	Quantity	Price	Total \$	
2	Support	Stamps	50	0.60	30.00	
3	Support	Stamps	50	0.60	30.00	
4	Support	Stamps	50	0.60	30.00	
5	Support	Staples	50	1.10	55.00	
6	Support	Staples	50	1.10	55.00	
7	Front Office	Paper	250	0.60	150.00	
8	Front Office	Paper	250	0.60	150.00	
9	Front Office	Paper	250	0.60	150.00	
10	Front Office	Paper	250	0.60	150.00	
11	Front Office	Paper	250	0.60	150.00	
12	Sales	Pens	200	0.60	120.00	
13	Sales	Pens	200	0.60	120.00	
14	Sales	Pens	200	0.60	120.00	
15	Sales	Pens	200	0.60	120.00	
16	Sales	Pens	200	0.60	120.00	
17						
18	Support	Stamps	50	0.6	30	
19	Support	Staples	50	1.1	55	
20	Front Office	Paper	250	0.6	150	
21	Sales	Pens	200	0.6	120	
22						

10. SUMIFS function in Excel.

Already discussed in Day 6 of #excelwithumaer series.