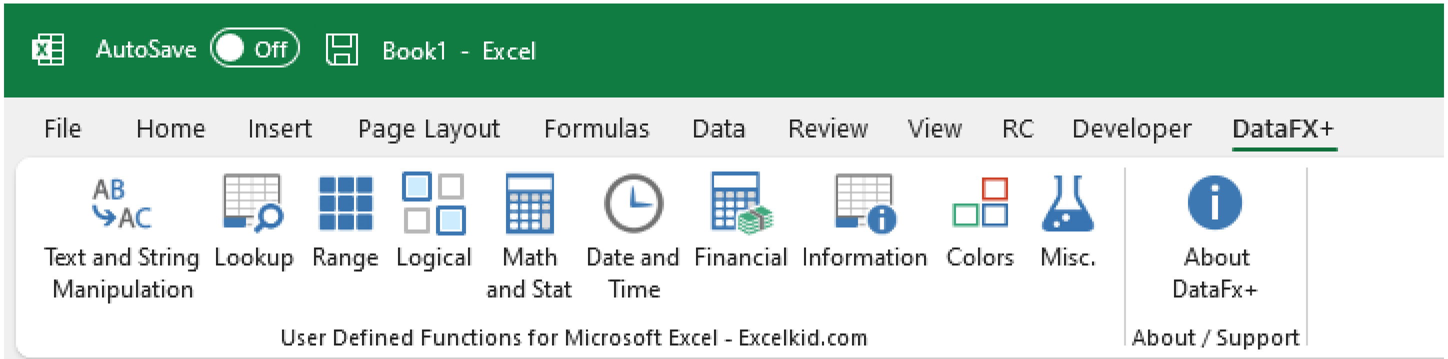
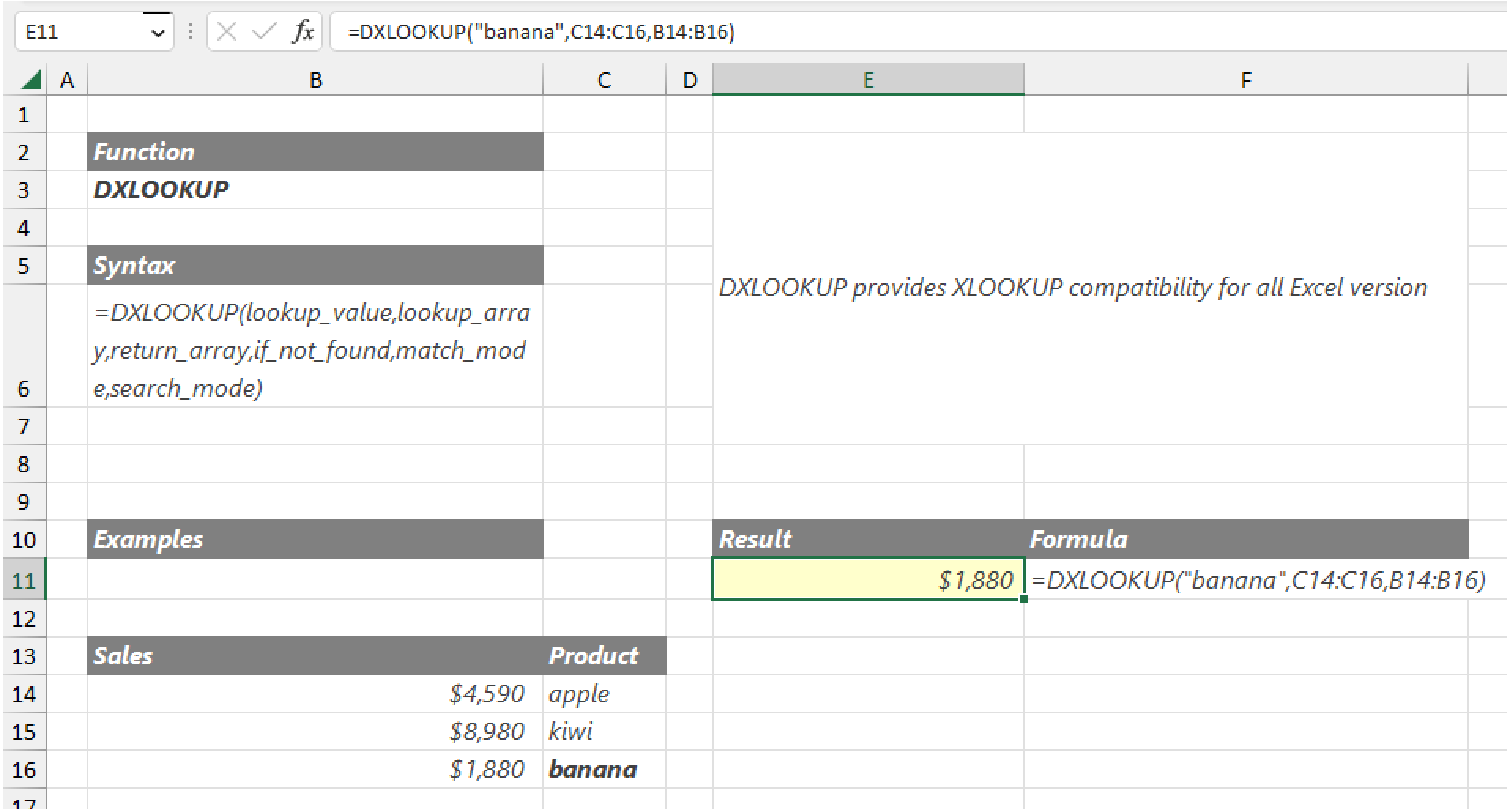
DataFX for Excel

Lookup functions



DXLOOKUP



=DXLOOKUP("banana",C14:C16,B14:B16)

Function

DXLOOKUP

Syntax

=DXLOOKUP(lookup_value,lookup_array,return_array,if_not_found,match_mode,search_mode)

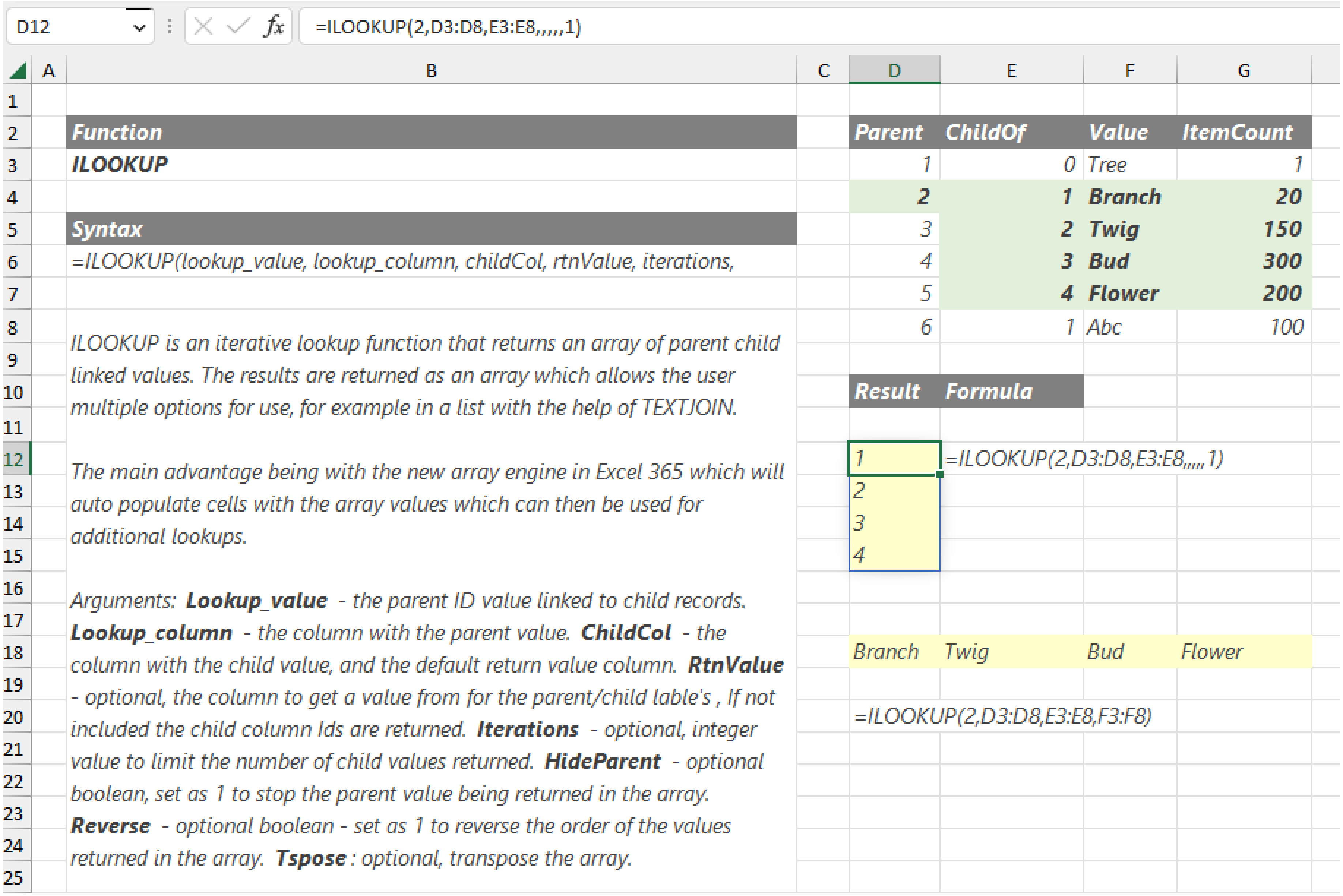
DXLOOKUP provides XLOOKUP compatibility for all Excel version

Examples

Result	Formula
$1,880	*=DXLOOKUP("banana",C14:C16,B14:B16)*

Sales	Product
$4,590	*apple*
$8,980	*kiwi*
$1,880	***banana***

ILOOKUP



D12

=ILOOKUP(2,D3:D8,E3:E8,,,,,1)

Function

ILOOKUP

Syntax

=ILOOKUP(lookup_value, lookup_column, childCol, rtnValue, iterations,

ILOOKUP is an iterative lookup function that returns an array of parent child linked values. The results are returned as an array which allows the user multiple options for use, for example in a list with the help of TEXTJOIN.

The main advantage being with the new array engine in Excel 365 which will auto populate cells with the array values which can then be used for additional lookups.

*Arguments: **Lookup_value** - the parent ID value linked to child records. **Lookup_column** - the column with the parent value. **ChildCol** - the column with the child value, and the default return value column. **RtnValue** - optional, the column to get a value from for the parent/child lable's , If not included the child column Ids are returned. **Iterations** - optional, integer value to limit the number of child values returned. **HideParent** - optional boolean, set as 1 to stop the parent value being returned in the array. **Reverse** - optional boolean - set as 1 to reverse the order of the values returned in the array. **Tspose** : optional, transpose the array.*

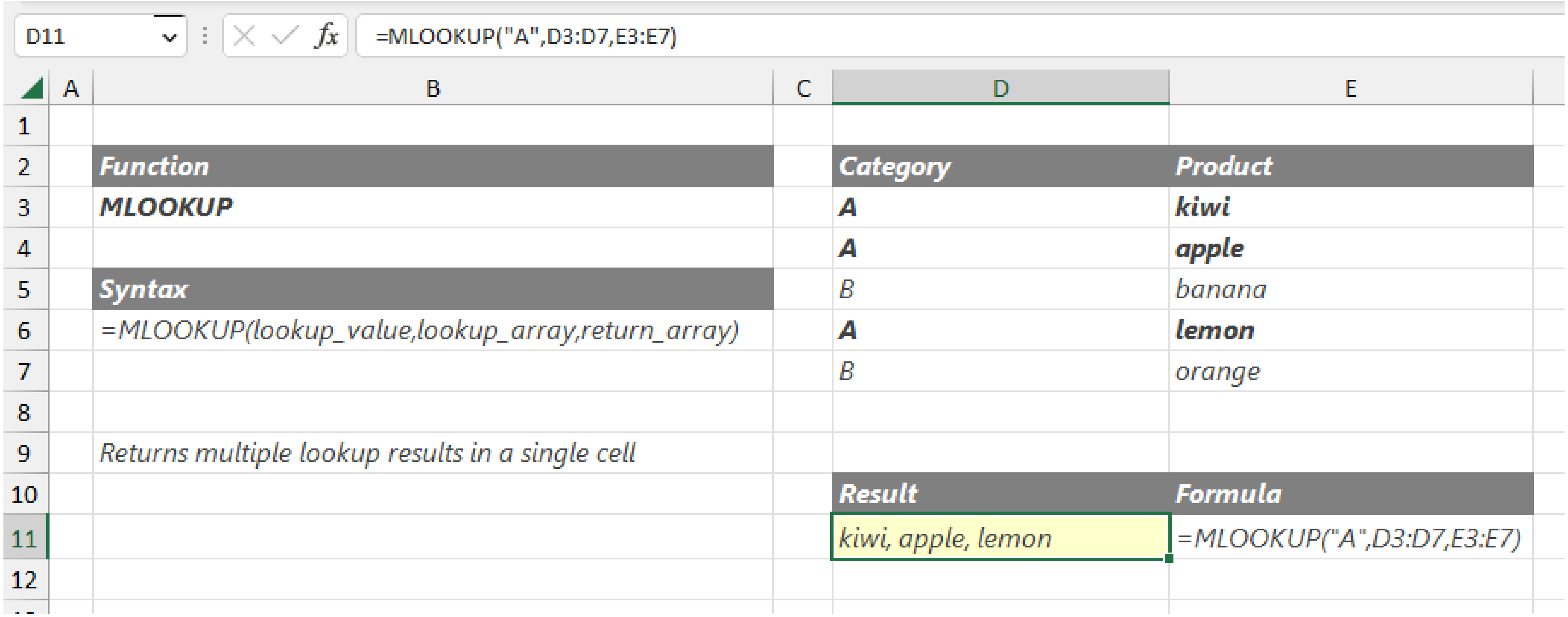
Parent	ChildOf	Value	ItemCount
1	0	Tree	1
2	**1**	**Branch**	**20**
3	**2**	**Twig**	**150**
4	**3**	**Bud**	**300**
5	**4**	**Flower**	**200**
6	1	Abc	100

Result	Formula
1	=ILOOKUP(2,D3:D8,E3:E8,,,,,1)
2	
3	
4	

Branch	Twig	Bud	Flower

=ILOOKUP(2,D3:D8,E3:E8,F3:F8)

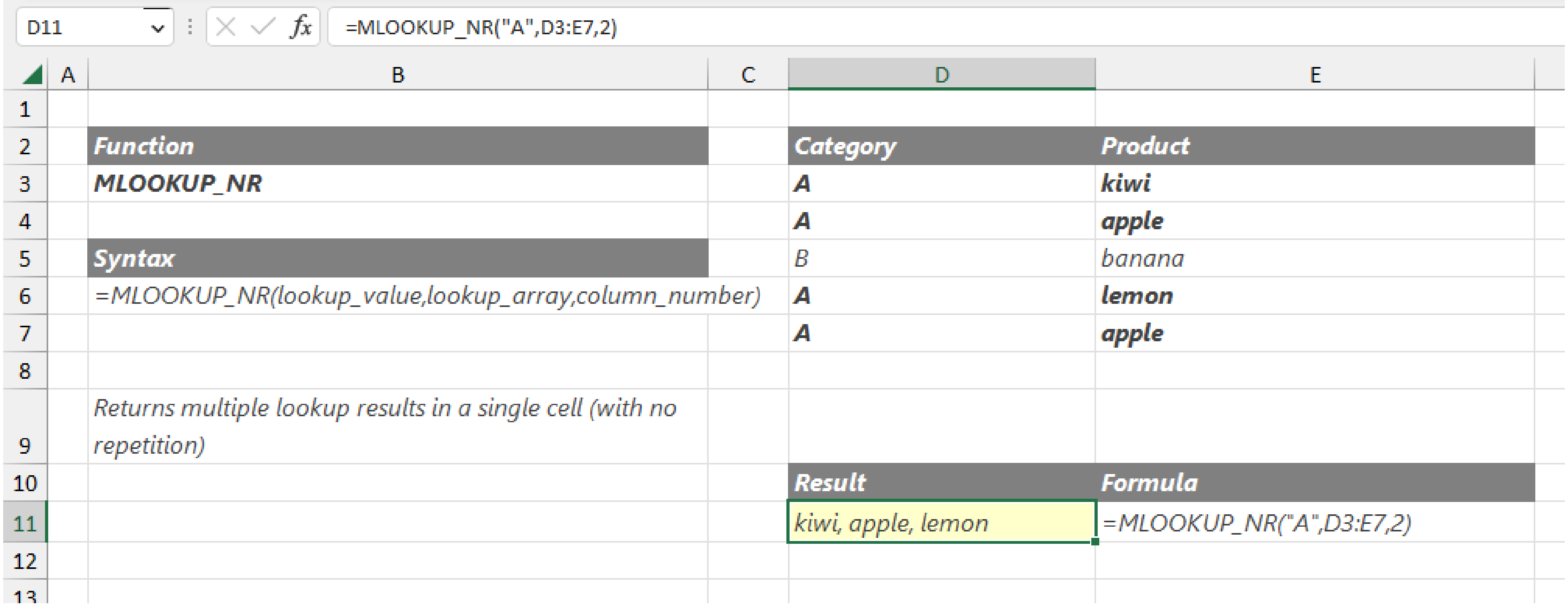
MLOOKUP



D11 =MLOOKUP("A",D3:D7,E3:E7)

	A	B	C	D	E
1					
2		***Function***		***Category***	***Product***
3		***MLOOKUP***		***A***	***kiwi***
4				***A***	***apple***
5		***Syntax***		*B*	*banana*
6		*=MLOOKUP(lookup_value,lookup_array,return_array)*		***A***	***lemon***
7				*B*	*orange*
8					
9		*Returns multiple lookup results in a single cell*			
10				***Result***	***Formula***
11				*kiwi, apple, lemon*	*=MLOOKUP("A",D3:D7,E3:E7)*
12					

MLOOKUP_NR

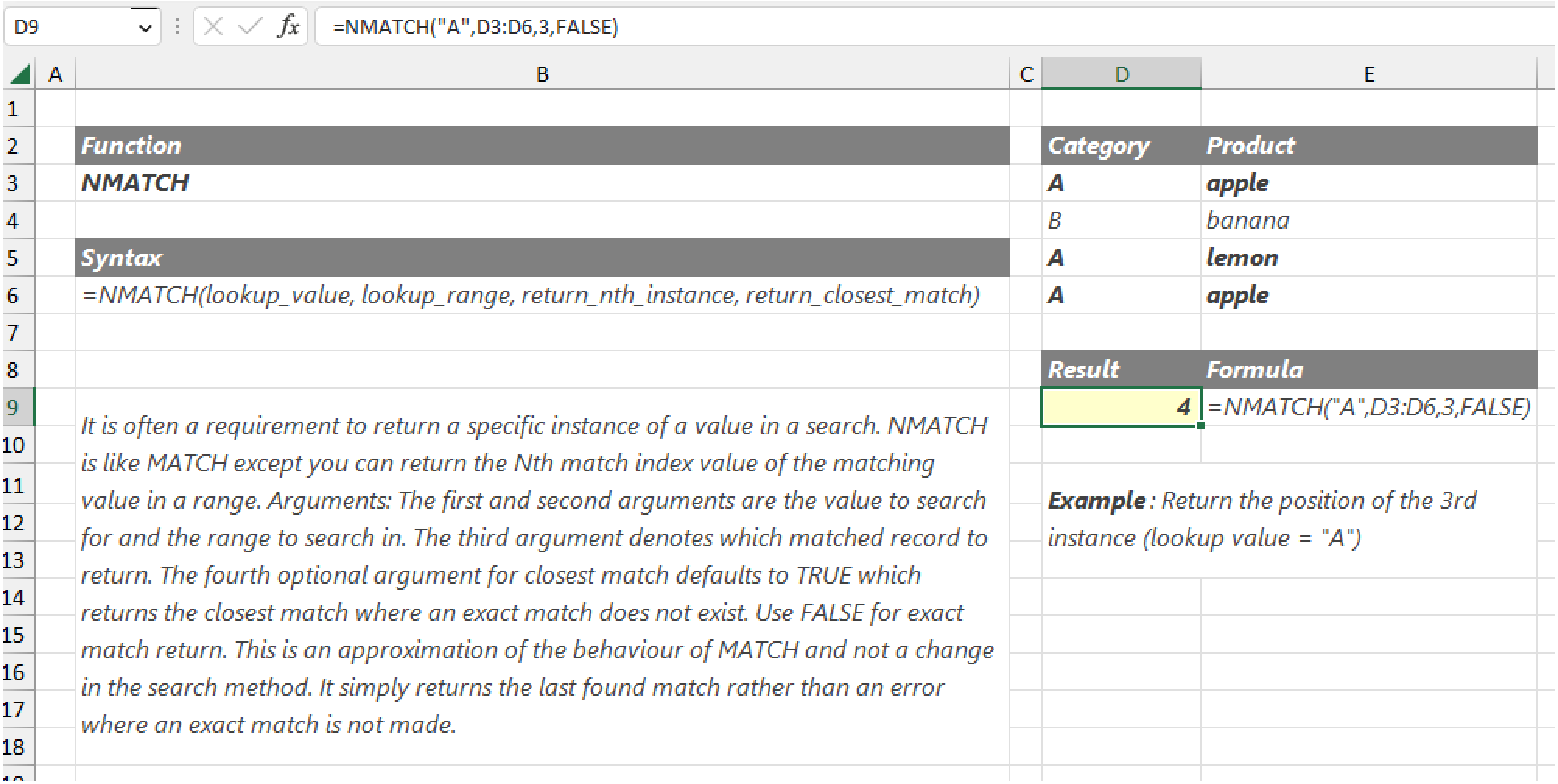


D11

=MLOOKUP_NR("A",D3:E7,2)

	A	B	C	D	E
1					
2		***Function***		***Category***	***Product***
3		***MLOOKUP_NR***		***A***	***kiwi***
4				***A***	***apple***
5		***Syntax***		*B*	*banana*
6		*=MLOOKUP_NR(lookup_value,lookup_array,column_number)*		***A***	***lemon***
7				***A***	***apple***
8					
9		*Returns multiple lookup results in a single cell (with no repetition)*			
10				***Result***	***Formula***
11				*kiwi, apple, lemon*	*=MLOOKUP_NR("A",D3:E7,2)*
12					
13					

NMATCH



D9

=NMATCH("A",D3:D6,3,FALSE)

Function

NMATCH

Syntax

=NMATCH(lookup_value, lookup_range, return_nth_instance, return_closest_match)

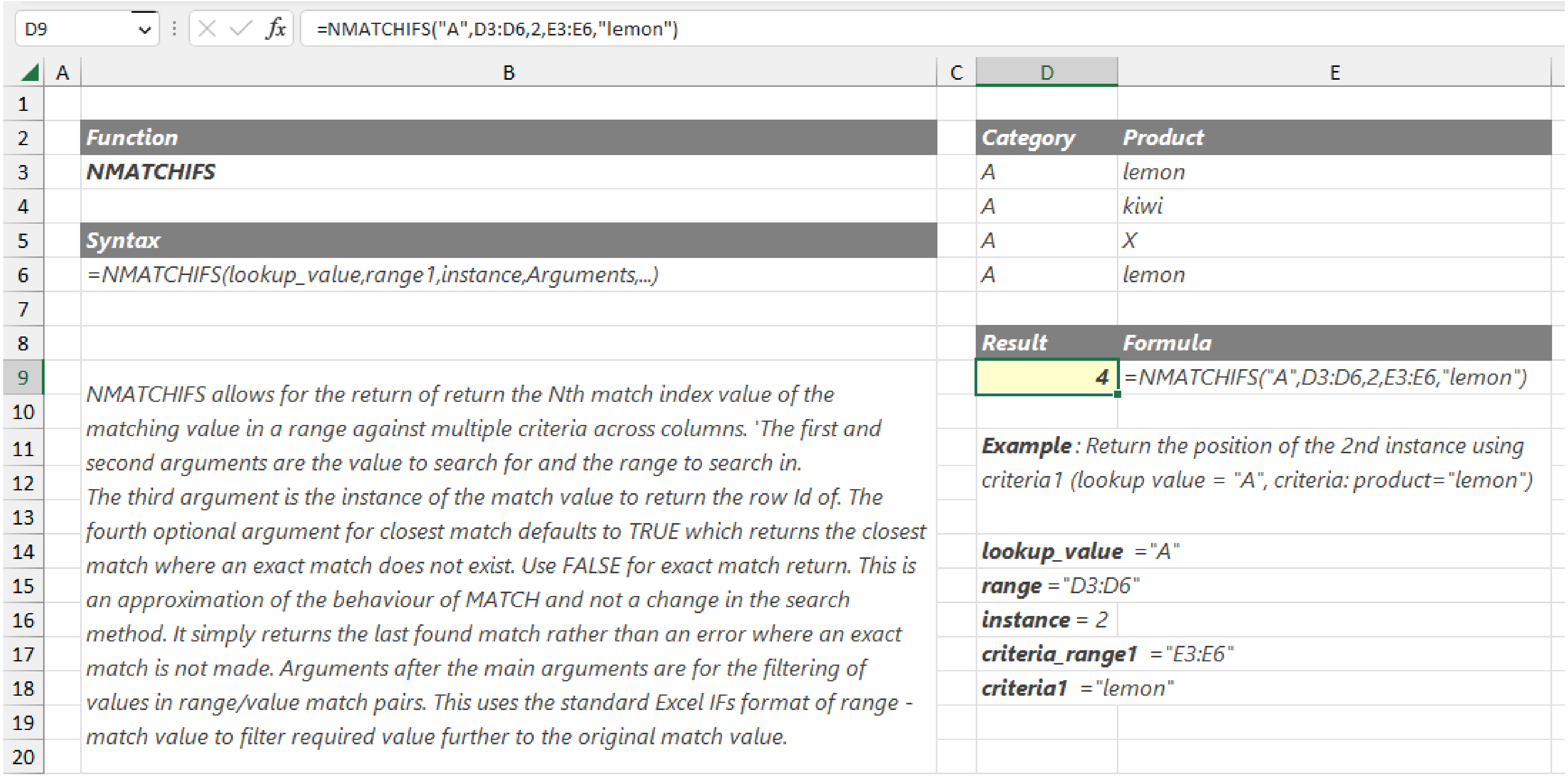
It is often a requirement to return a specific instance of a value in a search. NMATCH is like MATCH except you can return the Nth match index value of the matching value in a range. Arguments: The first and second arguments are the value to search for and the range to search in. The third argument denotes which matched record to return. The fourth optional argument for closest match defaults to TRUE which returns the closest match where an exact match does not exist. Use FALSE for exact match return. This is an approximation of the behaviour of MATCH and not a change in the search method. It simply returns the last found match rather than an error where an exact match is not made.

Category	***Product***
A	***apple***
B	*banana*
A	***lemon***
A	***apple***

Result	***Formula***
4	*=NMATCH("A",D3:D6,3,FALSE)*

Example *: Return the position of the 3rd instance (lookup value = "A")*

NMATCHIFS



D9

=NMATCHIFS("A",D3:D6,2,E3:E6,"lemon")

Function

NMATCHIFS

Syntax

=NMATCHIFS(lookup_value,range1,instance,Arguments,...)

NMATCHIFS allows for the return of return the Nth match index value of the matching value in a range against multiple criteria across columns. 'The first and second arguments are the value to search for and the range to search in. The third argument is the instance of the match value to return the row Id of. The fourth optional argument for closest match defaults to TRUE which returns the closest match where an exact match does not exist. Use FALSE for exact match return. This is an approximation of the behaviour of MATCH and not a change in the search method. It simply returns the last found match rather than an error where an exact match is not made. Arguments after the main arguments are for the filtering of values in range/value match pairs. This uses the standard Excel IFs format of range - match value to filter required value further to the original match value.

Category	***Product***
A	*lemon*
A	*kiwi*
A	*X*
A	*lemon*

Result	***Formula***
4	*=NMATCHIFS("A",D3:D6,2,E3:E6,"lemon")*

***Example** : Return the position of the 2nd instance using criteria1 (lookup value = "A", criteria: product="lemon")*

***lookup_value** ="A"*

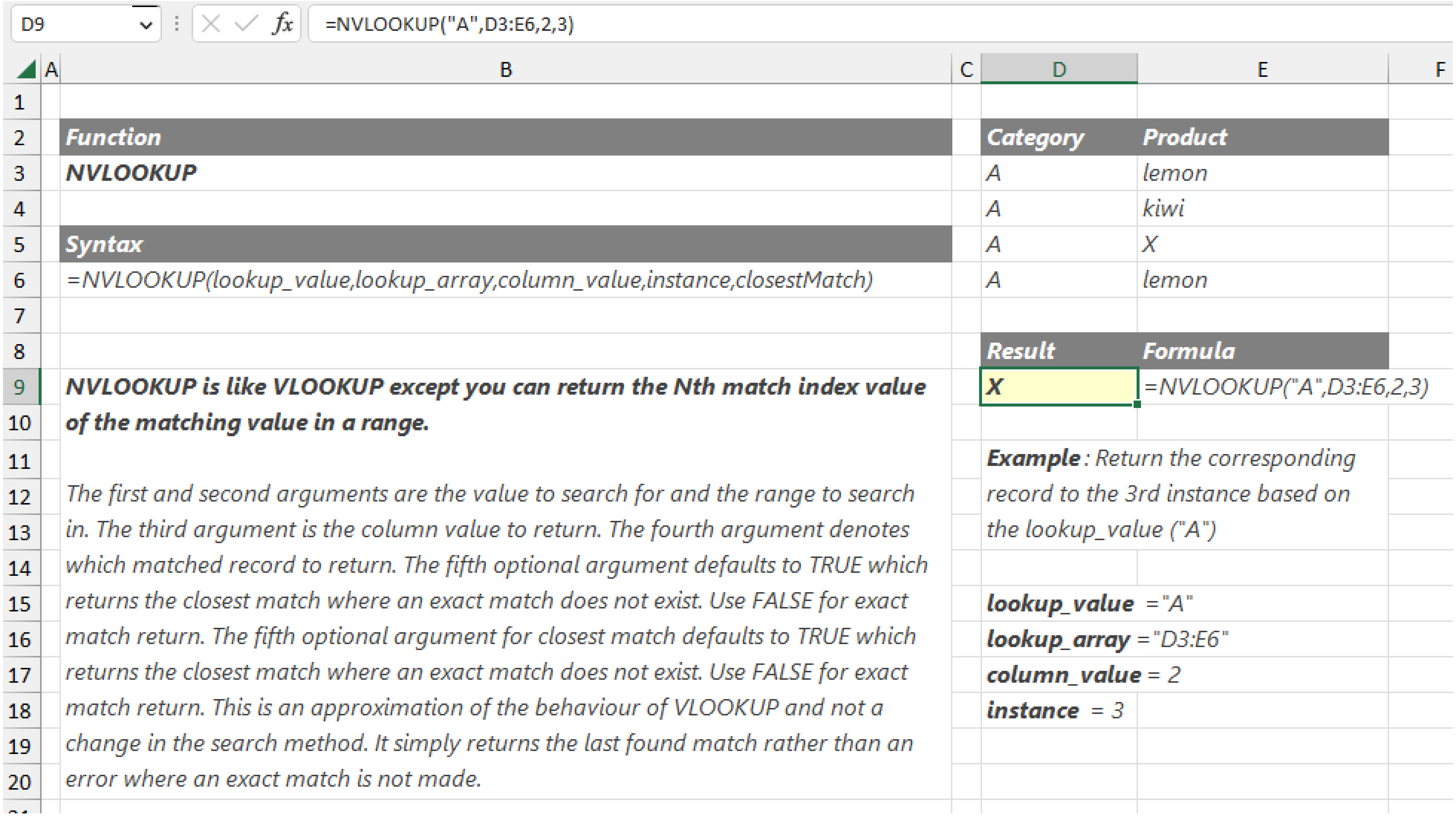
***range** ="D3:D6"*

***instance** = 2*

***criteria_range1** ="E3:E6"*

***criteria1** ="lemon"*

NVLOOKUP



D9

=NVLOOKUP("A",D3:E6,2,3)

Function

NVLOOKUP

Syntax

=NVLOOKUP(lookup_value,lookup_array,column_value,instance,closestMatch)

NVLOOKUP is like VLOOKUP except you can return the Nth match index value of the matching value in a range.

The first and second arguments are the value to search for and the range to search in. The third argument is the column value to return. The fourth argument denotes which matched record to return. The fifth optional argument defaults to TRUE which returns the closest match where an exact match does not exist. Use FALSE for exact match return. The fifth optional argument for closest match defaults to TRUE which returns the closest match where an exact match does not exist. Use FALSE for exact match return. This is an approximation of the behaviour of VLOOKUP and not a change in the search method. It simply returns the last found match rather than an error where an exact match is not made.

Category	***Product***
A	*lemon*
A	*kiwi*
A	*X*
A	*lemon*

Result	***Formula***
X	*=NVLOOKUP("A",D3:E6,2,3)*

***Example** : Return the corresponding record to the 3rd instance based on the lookup_value ("A")*

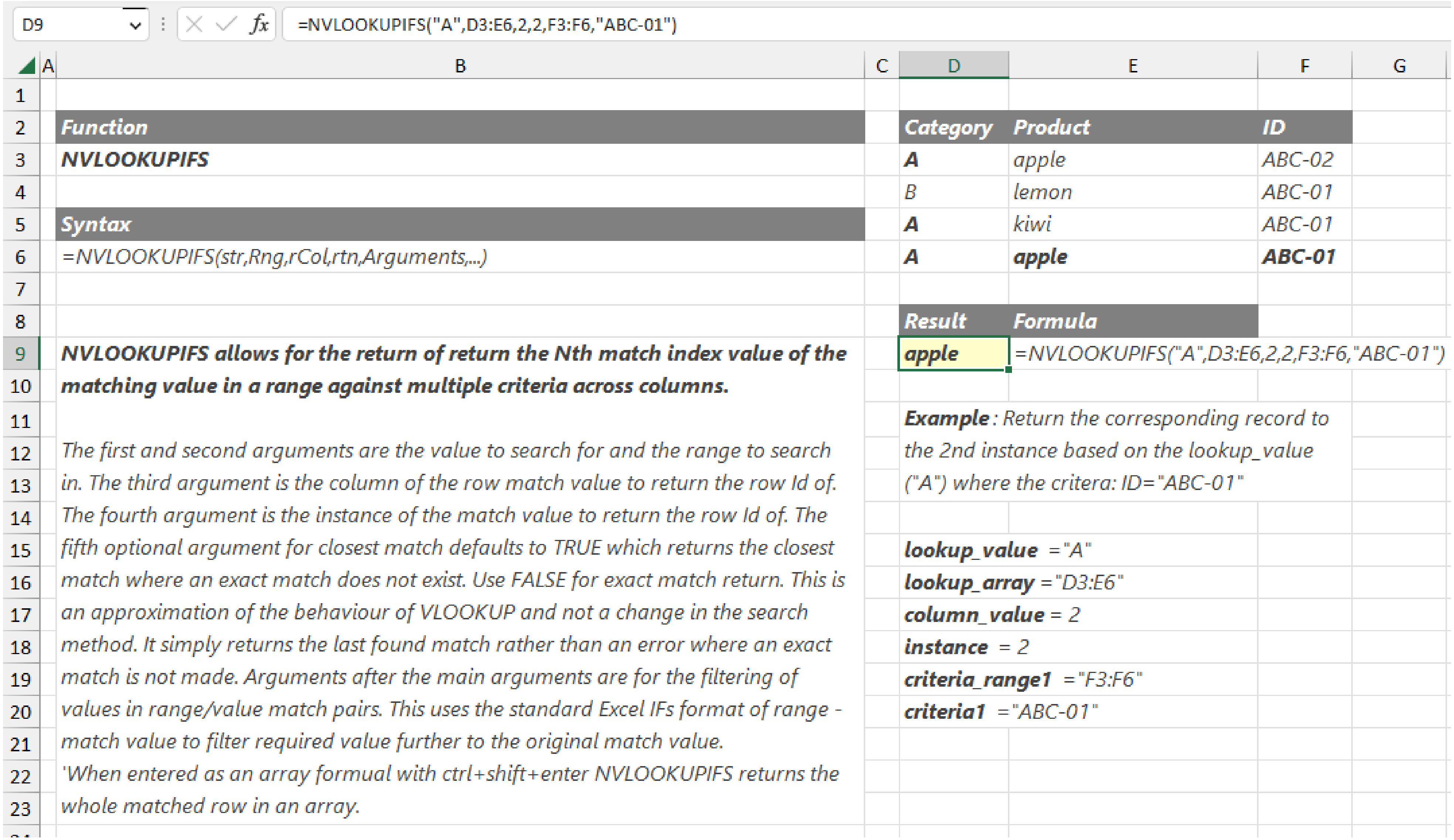
***lookup_value** ="A"*

***lookup_array** ="D3:E6"*

***column_value** = 2*

***instance** = 3*

NVLOOKUPIFS



D9

```
=NVLOOKUPIFS("A",D3:E6,2,2,F3:F6,"ABC-01")
```

Function

NVLOOKUPIFS

Syntax

=NVLOOKUPIFS(str,Rng,rCol,rtn,Arguments,...)

NVLOOKUPIFS allows for the return of return the Nth match index value of the matching value in a range against multiple criteria across columns.

The first and second arguments are the value to search for and the range to search in. The third argument is the column of the row match value to return the row Id of. The fourth argument is the instance of the match value to return the row Id of. The fifth optional argument for closest match defaults to TRUE which returns the closest match where an exact match does not exist. Use FALSE for exact match return. This is an approximation of the behaviour of VLOOKUP and not a change in the search method. It simply returns the last found match rather than an error where an exact match is not made. Arguments after the main arguments are for the filtering of values in range/value match pairs. This uses the standard Excel IFs format of range - match value to filter required value further to the original match value.

'When entered as an array formual with ctrl+shift+enter NVLOOKUPIFS returns the whole matched row in an array.

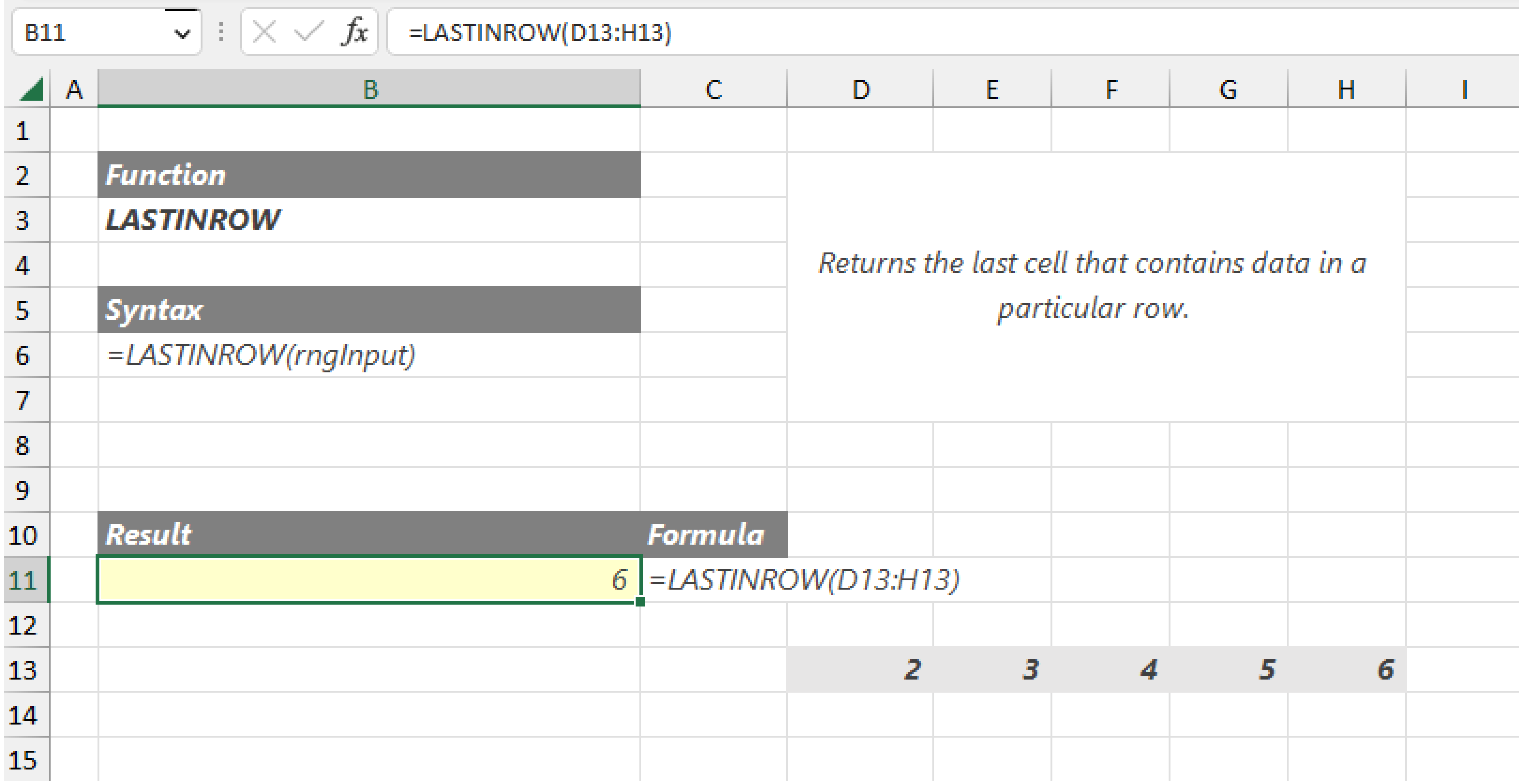
Category	Product	ID
A	*apple*	*ABC-02*
B	*lemon*	*ABC-01*
A	*kiwi*	*ABC-01*
A	***apple***	***ABC-01***

Result	Formula
apple	*=NVLOOKUPIFS("A",D3:E6,2,2,F3:F6,"ABC-01")*

***Example** : Return the corresponding record to the 2nd instance based on the lookup_value ("A") where the critera: ID="ABC-01"*

lookup_value *="A"*
lookup_array *="D3:E6"*
column_value *= 2*
instance *= 2*
criteria_range1 *="F3:F6"*
criteria1 *="ABC-01"*

LASTINROW



=LASTINROW(D13:H13)

Function

LASTINROW

Syntax

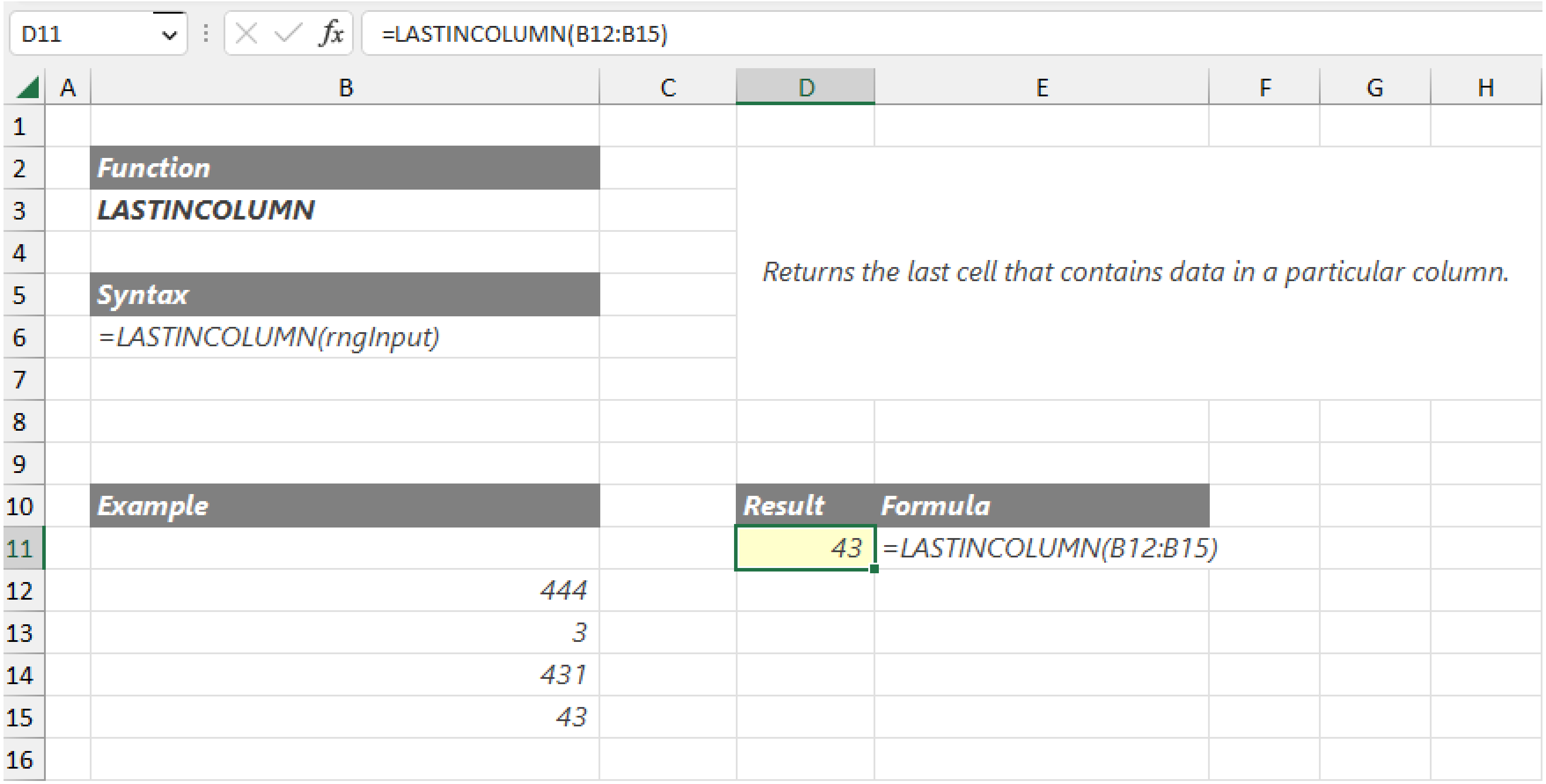
=LASTINROW(rngInput)

Returns the last cell that contains data in a particular row.

Result	**Formula**
6	*=LASTINROW(D13:H13)*

D	E	F	G	H
2	***3***	***4***	***5***	***6***

LASTINCOLUMN



D11

=LASTINCOLUMN(B12:B15)

Function

LASTINCOLUMN

Syntax

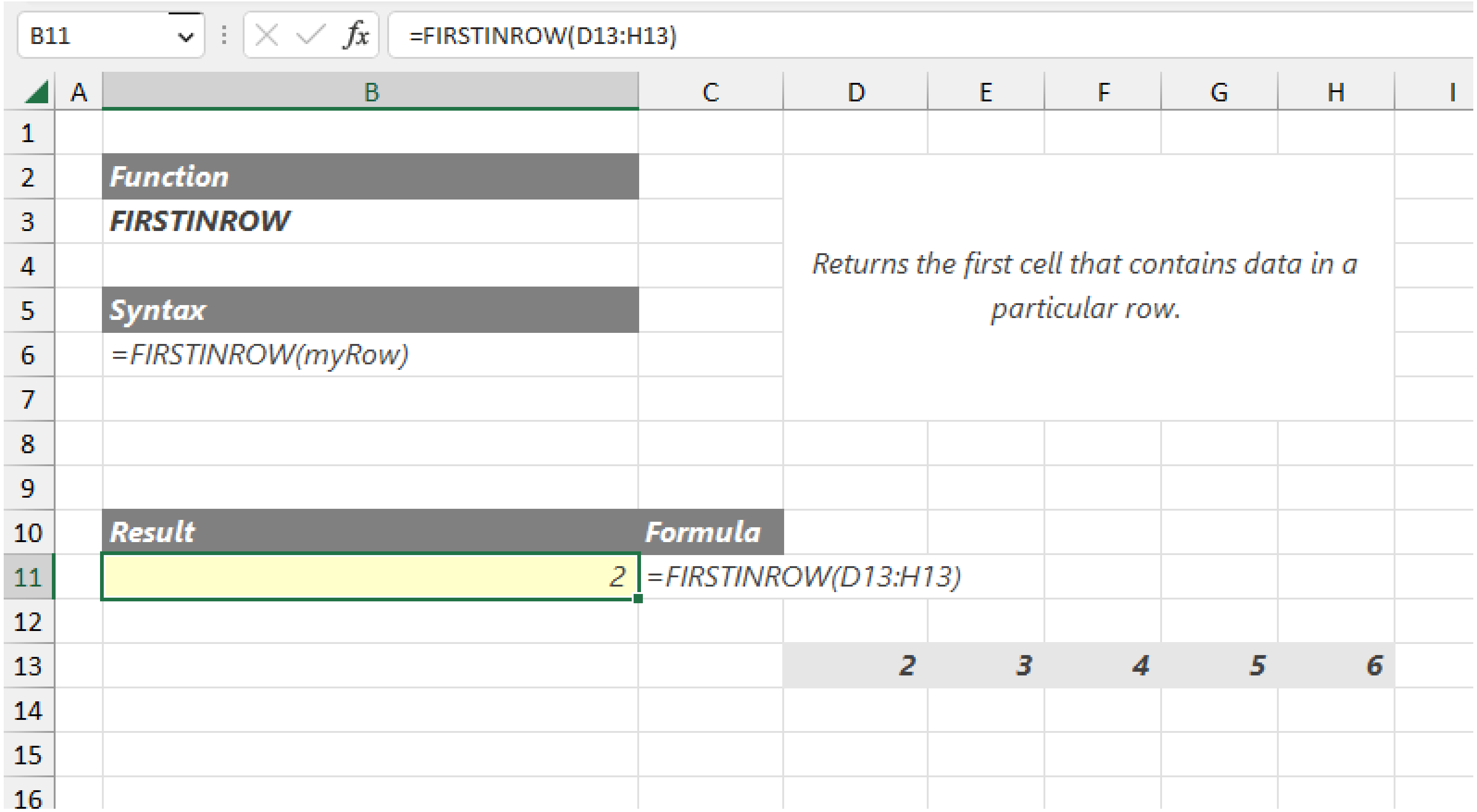
=LASTINCOLUMN(rngInput)

Returns the last cell that contains data in a particular column.

Example

Row	B	D	E
10		**Result**	**Formula**
11		*43*	*=LASTINCOLUMN(B12:B15)*
12	*444*		
13	*3*		
14	*431*		
15	*43*		

FIRSTINROW



B11 =FIRSTINROW(D13:H13)

Function

FIRSTINROW

Syntax

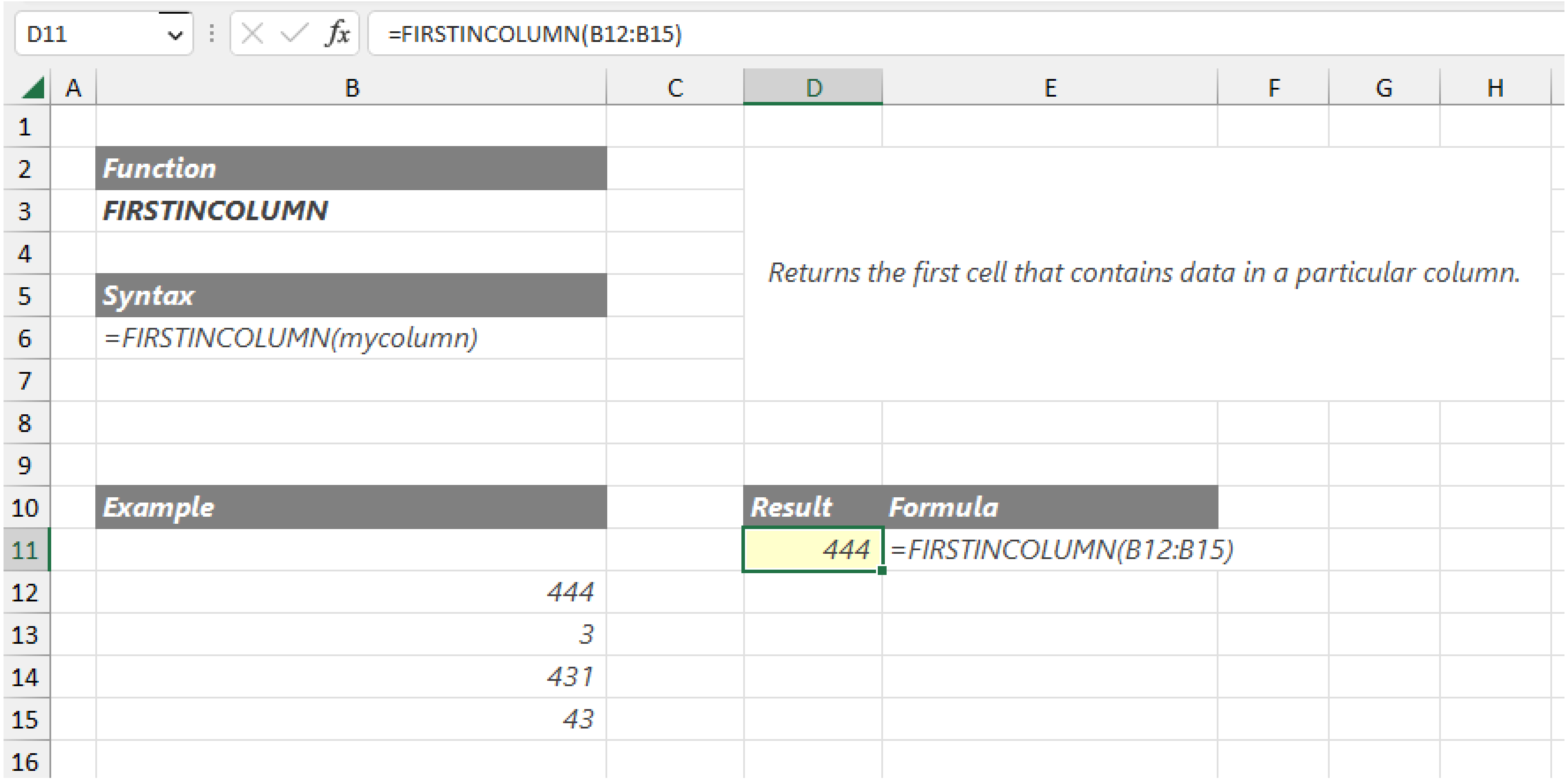
=FIRSTINROW(myRow)

Returns the first cell that contains data in a particular row.

Result	Formula
2	*=FIRSTINROW(D13:H13)*

D	E	F	G	H
2	***3***	***4***	***5***	***6***

FIRSTINCOLUMN



D11

=FIRSTINCOLUMN(B12:B15)

Function

FIRSTINCOLUMN

Syntax

=FIRSTINCOLUMN(mycolumn)

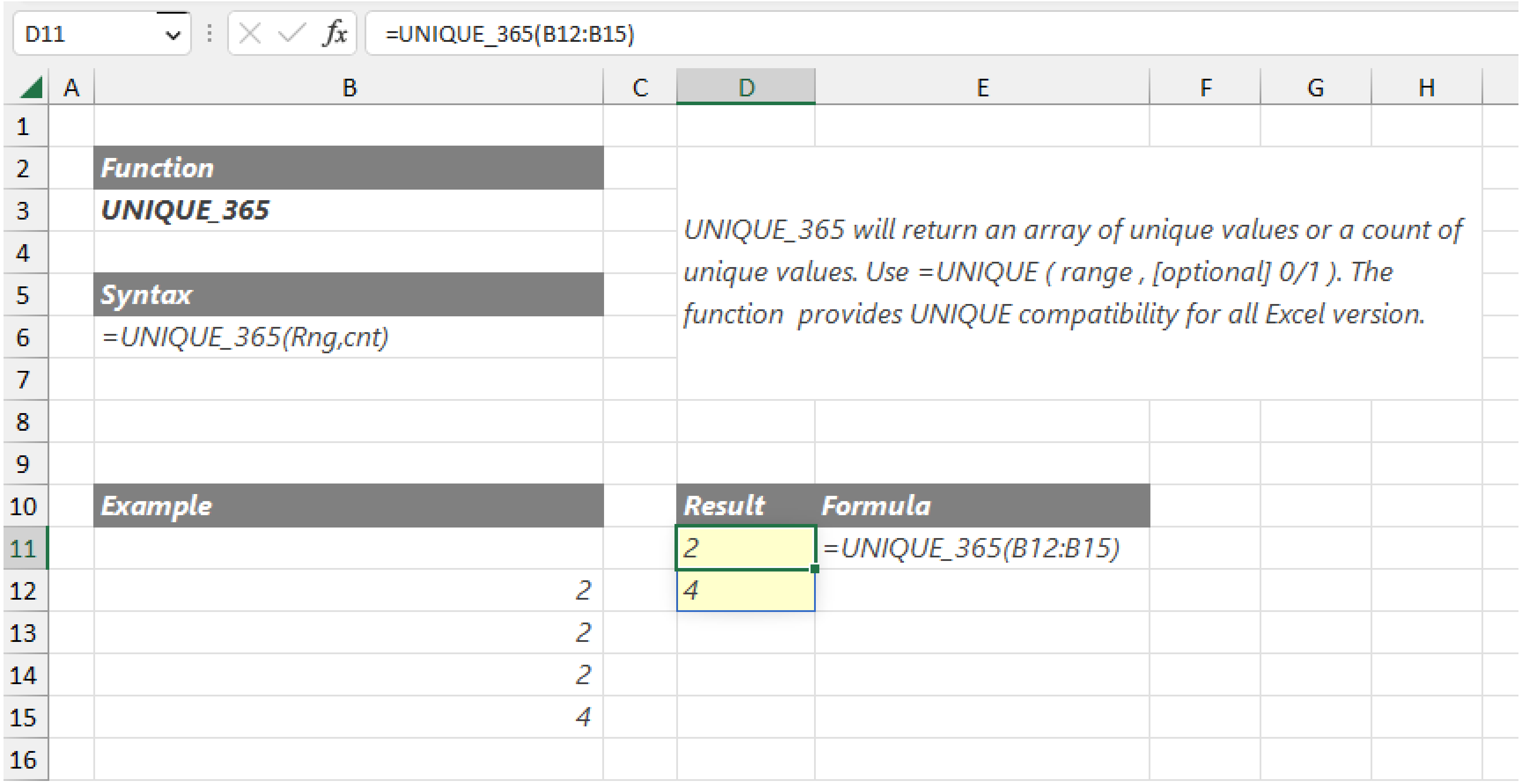
Returns the first cell that contains data in a particular column.

Example

Result	Formula
444	*=FIRSTINCOLUMN(B12:B15)*

444
3
431
43

UNIQUE_365



D11 | =UNIQUE_365(B12:B15)

	A	B	C	D	E	F	G	H
1								
2		***Function***						
3		***UNIQUE_365***		*UNIQUE_365 will return an array of unique values or a count of unique values. Use =UNIQUE (range , [optional] 0/1). The function provides UNIQUE compatibility for all Excel version.*				
4								
5		***Syntax***						
6		*=UNIQUE_365(Rng,cnt)*						
7								
8								
9								
10		***Example***		***Result***	***Formula***			
11				*2*	*=UNIQUE_365(B12:B15)*			
12		*2*		*4*				
13		*2*						
14		*2*						
15		*4*						
16								