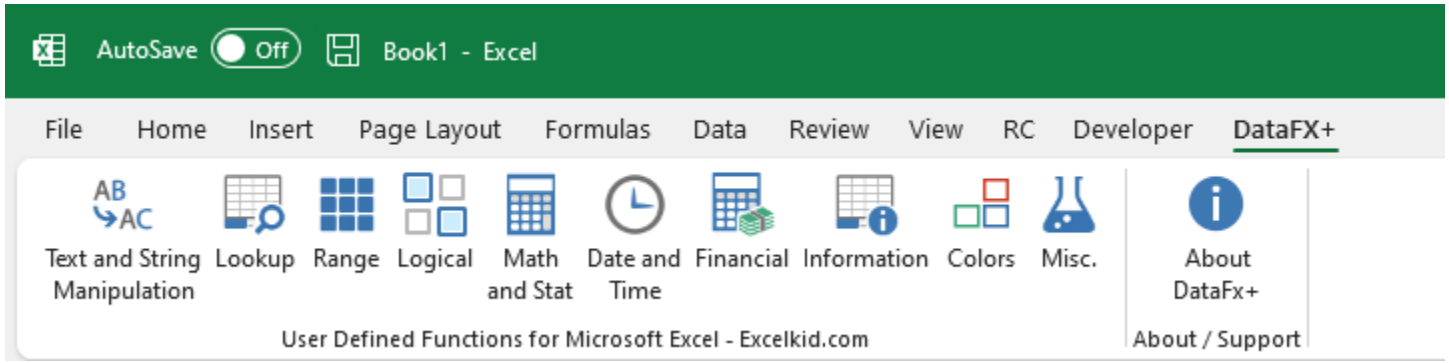
DataFX for Excel

String manipulation functions



ABBREVIATE

	A	B	C	D	E	F
1						
2		***Function***		*Return the abbreviation for the supplied string*		
3		***ABBREVIATE***				
4						
5		***Syntax***				
6		*=ABBREVIATE(string1)*				
7						
8						
9						
10		***Examples***		***Result***	***Formula***	
11		*Advanced Micro Devices*		*AMD*	*=ABBREVIATE(B11)*	
12		*International Business Machines*		*IBM*	*=ABBREVIATE(B12)*	
13						

CAMEL_CASE

	A	B	C	D	E	F
1						
2		**Function**				
3		**CAMEL_CASE**				
4				*Transform all your text characters to camel case. Camel Case also known as camel caps or more formally as medial capitals.*		
5		**Syntax**				
6		=CAMEL_CASE(string1)				
7						
8						
9						
10		**Examples**		**Result**	**Formula**	
11		Hello world		helloWorld	=CAMEL_CASE(B11)	
12		Have A great Day		haveAGreatDay	=CAMEL_CASE(B12)	
13		good morning		goodMorning	=CAMEL_CASE(B13)	
14						
15						
16						

CAPITALIZE

	A	B	C	D	E	F
1						
2		**Function**				
3		**CAPITALIZE**				
4				CAPITALIZE takes a string and returns the same string with the first		
5		**Syntax**		character capitalized and all other characters lowercased.		
6		=CAPITALIZE(string1)				
7						
8						
9						
10		**Examples**		**Result**	**Formula**	
11		Hello World		Hello world	=CAPITALIZE(B11)	
12		Have A GrEAT Day		Have a great day	=CAPITALIZE(B12)	
13		good morning		Good morning	=CAPITALIZE(B13)	
14						
15						
16						

COMPANY_CASE

	Function
	COMPANY_CASE
	Syntax
	=COMPANY_CASE(string1)

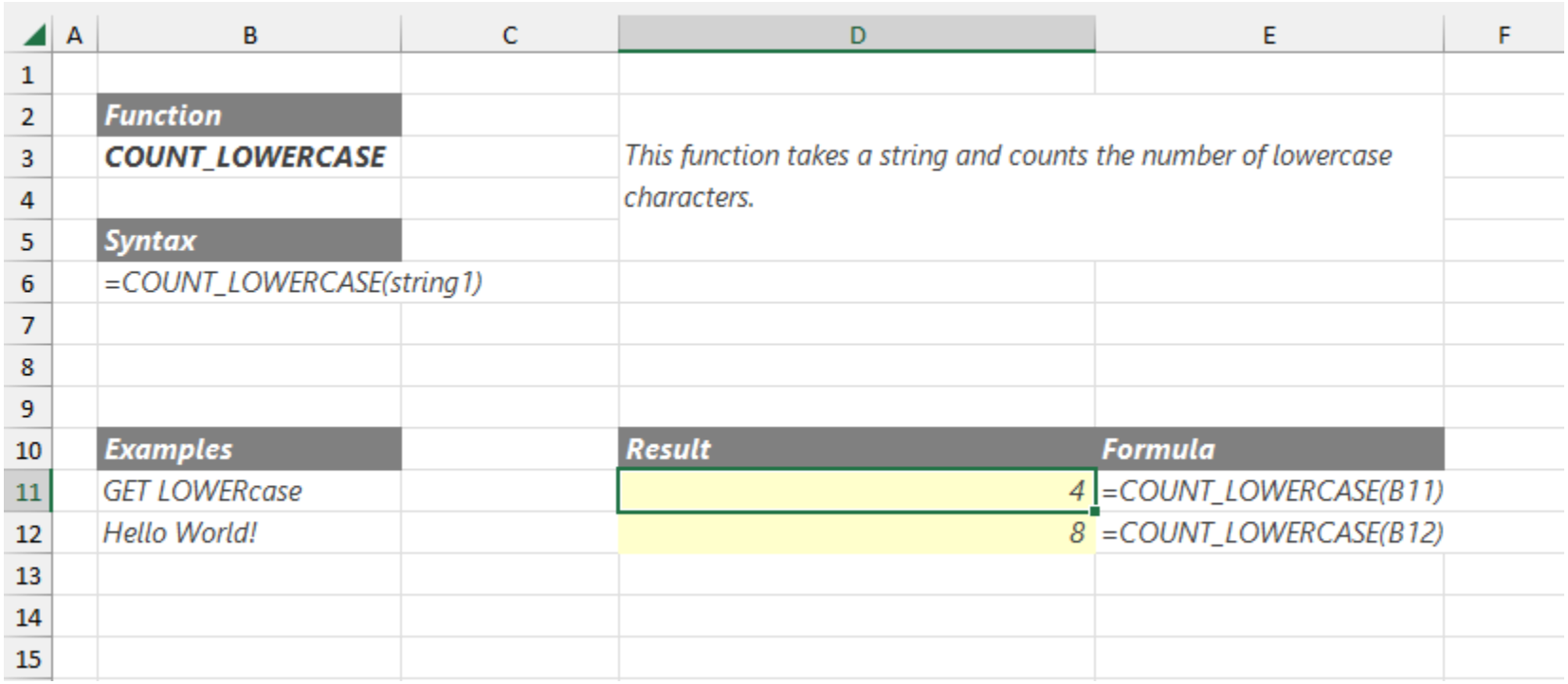
This function takes a string and uses an algorithm to return the string in Company Case. The standard =PROPER() function in Excel will not capitalize company names properly, as it only capitalizes based on space characters, so a name like "j.p. morgan" will be incorrectly formatted as "J.p. Morgan" instead of the correct "J.P. Morgan". Additionally, =PROPER() may incorrectly lowercase company abbreviations, such as the last "H" in "GmbH", as =PROPER() returns "Gmbh" instead of the correct "GmbH". This function attempts to adjust for these issues when a string is a company name.

Examples	Result	Formula
hello world	Hello World	=COMPANY_CASE(B13)
x.y.z company & co.	X.Y.Z Company & Co.	=COMPANY_CASE(B14)
x.y.z plc	X.Y.Z PLC	=COMPANY_CASE(B15)
one company gmbh	One Company GmbH	=COMPANY_CASE(B16)
two company a. en p.	Two Company A. en P.	=COMPANY_CASE(B17)
three company s. en n.c.	Three Company S. en N.C.	=COMPANY_CASE(B18)
FOUR COMPANY SPOL S.R.O.	Four Company spol s.r.o.	=COMPANY_CASE(B19)
five company bvba	Five Company BVBA	=COMPANY_CASE(B20)
Amazon.com Inc.	Amazon.com Inc.	=COMPANY_CASE(B21)
amazon.com inc.	Amazon.Com Inc.	=COMPANY_CASE(B22)

CONCAT_365

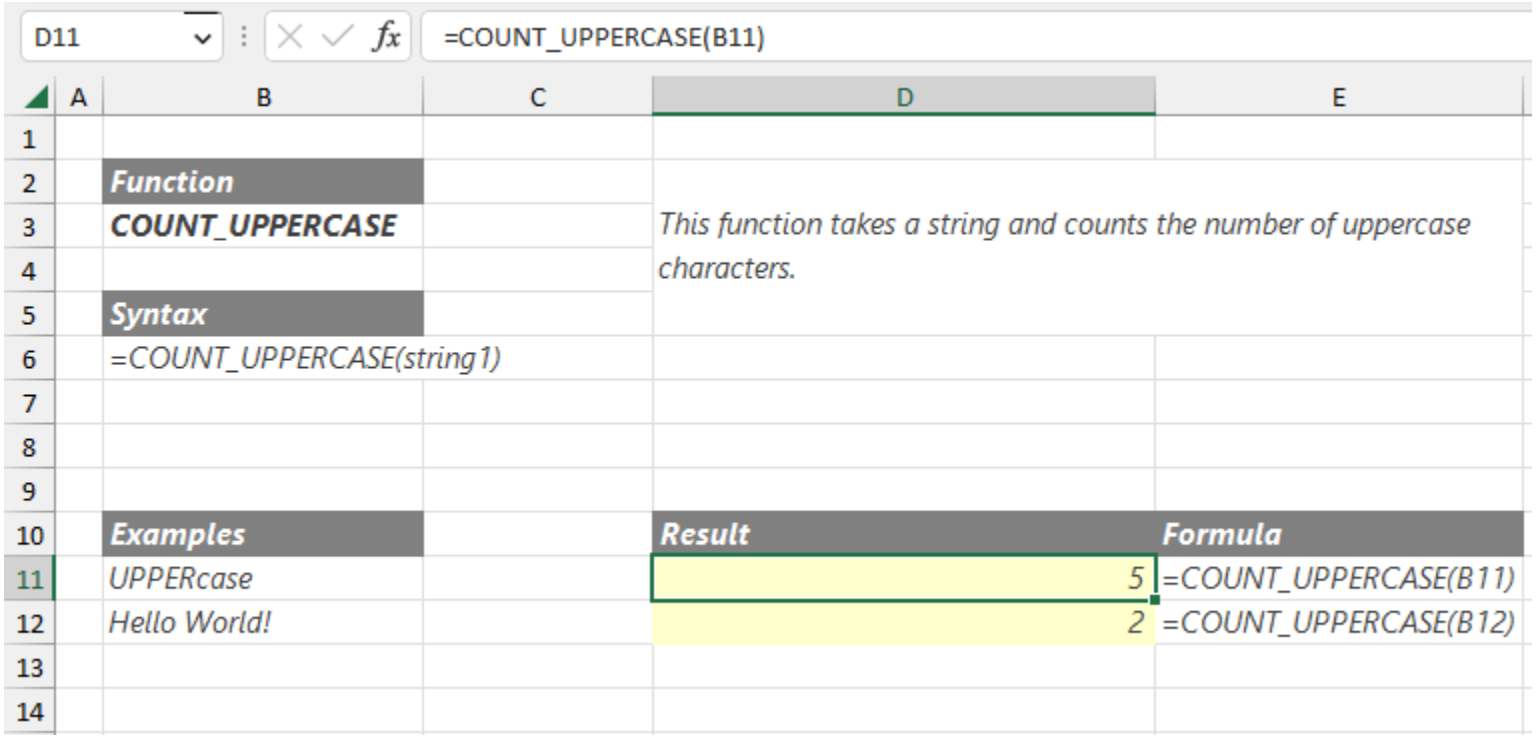
	A	B	C	D	E	F
1						
2		**Function**		*This function takes multiple ranges and strings and concatenates all of them together. It provides the CONCAT() function for all Excel versions.*		
3		**CONCAT_365**				
4						
5		**Syntax**				
6		=CONCAT_365(text1,...)				
7						
8						
9						
10		**Examples**		**Result**	**Formula**	
11		Hello		Hello World	=CONCAT_365(B11," ",B12)	
12		World				
13						
14						

COUNT_LOWERCASE



	A	B	C	D	E	F
1						
2		**Function**				
3		**COUNT_LOWERCASE**		*This function takes a string and counts the number of lowercase characters.*		
4						
5		**Syntax**				
6		=COUNT_LOWERCASE(string1)				
7						
8						
9						
10		**Examples**		**Result**	**Formula**	
11		GET LOWERcase		4	=COUNT_LOWERCASE(B11)	
12		Hello World!		8	=COUNT_LOWERCASE(B12)	
13						
14						
15						

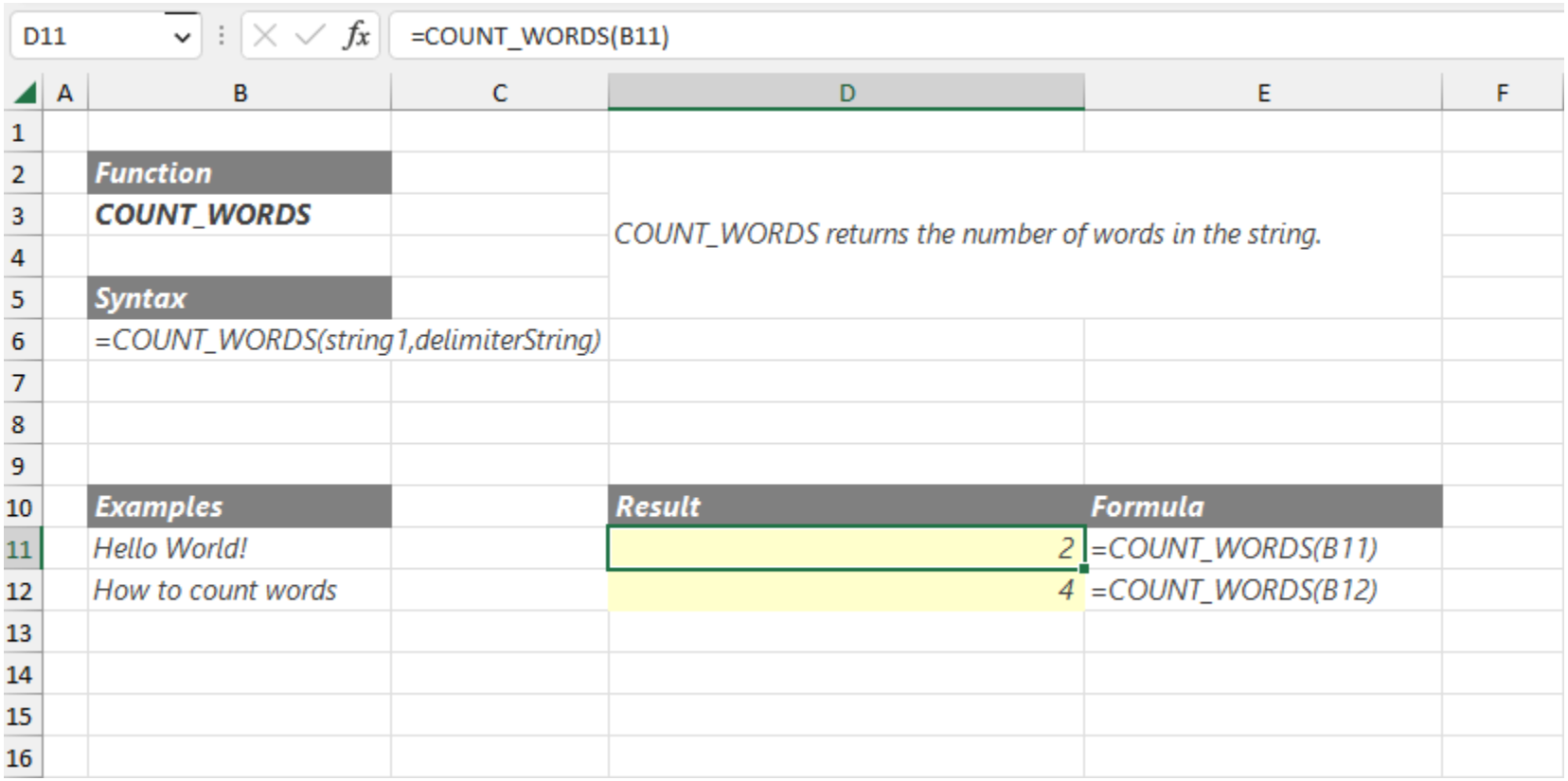
COUNT_UPPERCASE



D11 | =COUNT_UPPERCASE(B11)

	A	B	C	D	E
1					
2		**Function**			
3		**COUNT_UPPERCASE**		*This function takes a string and counts the number of uppercase characters.*	
4					
5		**Syntax**			
6		*=COUNT_UPPERCASE(string1)*			
7					
8					
9					
10		**Examples**		**Result**	**Formula**
11		*UPPERcase*		*5*	*=COUNT_UPPERCASE(B11)*
12		*Hello World!*		*2*	*=COUNT_UPPERCASE(B12)*
13					
14					

COUNT_WORDS



D11 | =COUNT_WORDS(B11)

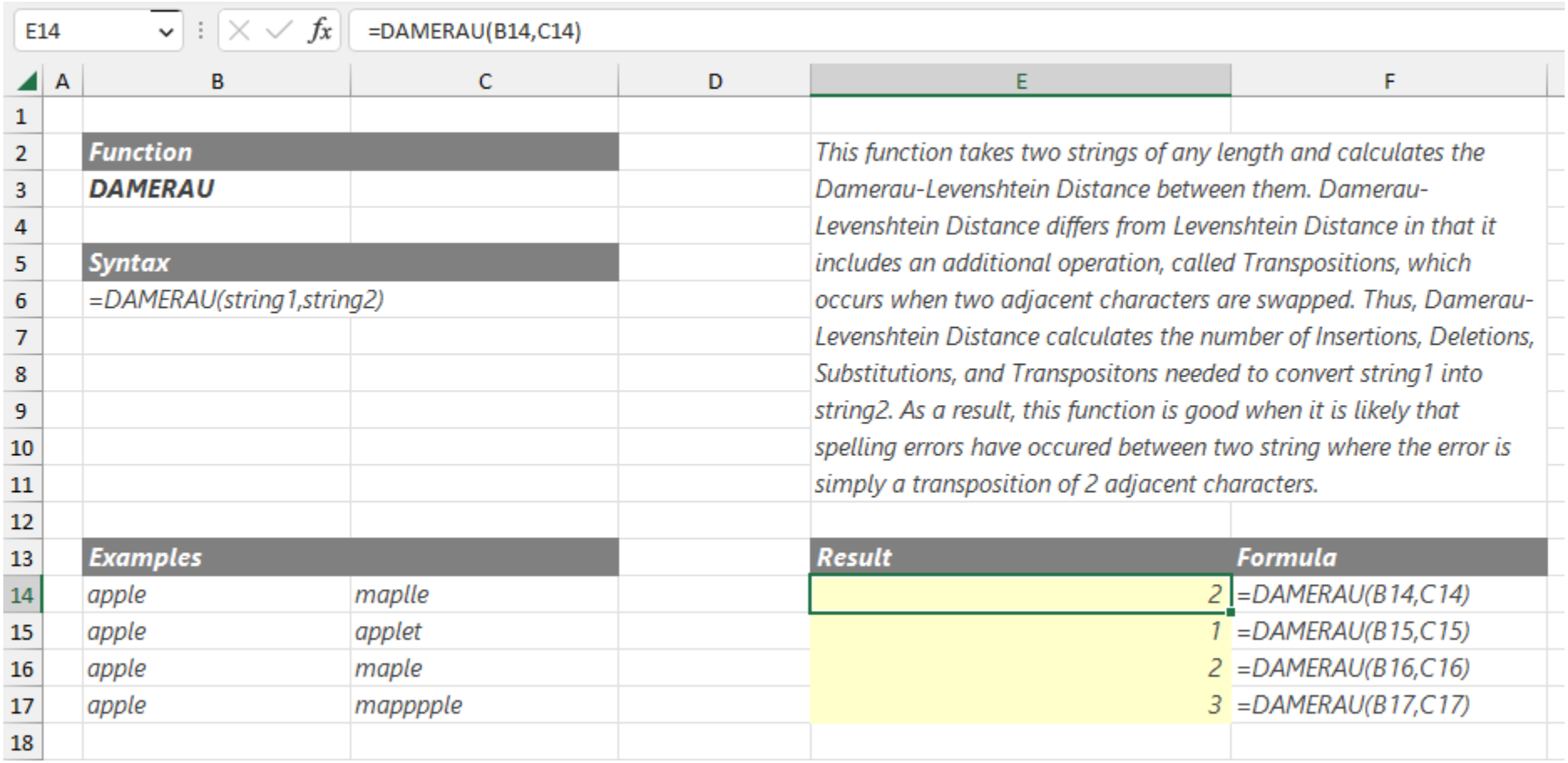
	A	B	C	D	E	F
1						
2		***Function***		*COUNT_WORDS returns the number of words in the string.*		
3		***COUNT_WORDS***				
4						
5		***Syntax***				
6		*=COUNT_WORDS(string1,delimiterString)*				
7						
8						
9						
10		***Examples***		***Result***	***Formula***	
11		*Hello World!*		*2*	*=COUNT_WORDS(B11)*	
12		*How to count words*		*4*	*=COUNT_WORDS(B12)*	
13						
14						
15						
16						

DAM_STR

E11 =DAM_STR(B11,C11:C14)

	A	B	C	D	E	F
1						
2		***Function***			*This function takes two ranges and calculates the string resulting from the lowest Damerau-Levenshtein Distance. The first range is a single cell which will be compared to all other cells in the second range, and whichever value produces the lowest Damerau-Levenshtein Distance will be returned. Returns the string that is closest from the rangeArray.*	
3		***DAM_STR***				
4						
5		***Syntax***				
6		*=DAM_STR(range1,rangeArray)*				
7						
8						
9						
10		***Examples***			***Result***	***Formula***
11		*apple*	*maplle*		*applet*	*=DAM_STR(B11,C11:C14)*
12			*applet*			
13			*maple*			
14			*mapppple*			
15						

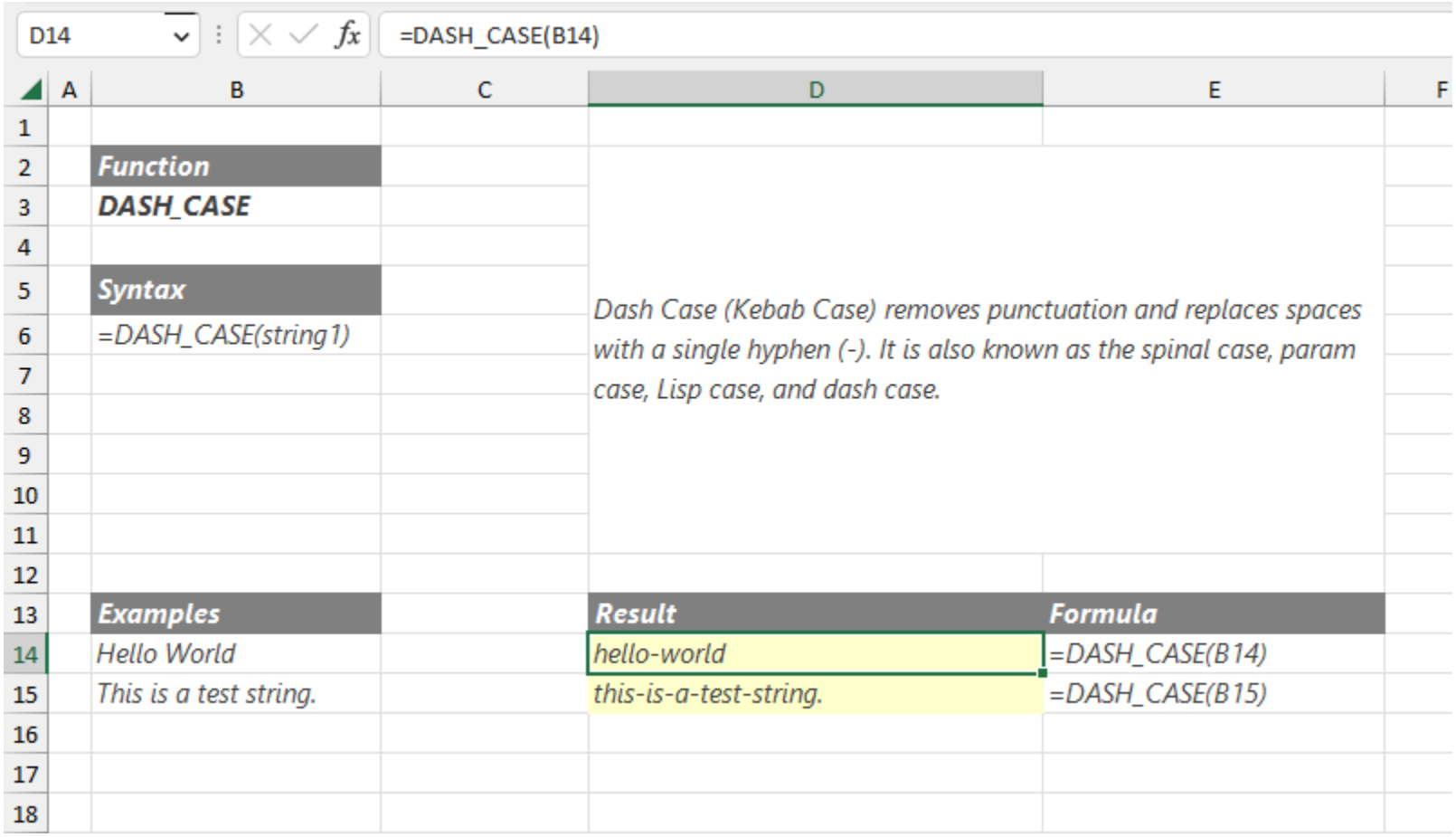
DAMERAU



E14 =DAMERAU(B14,C14)

	A	B	C	D	E	F
1						
2		**Function**			*This function takes two strings of any length and calculates the*	
3		**DAMERAU**			*Damerau-Levenshtein Distance between them. Damerau-*	
4					*Levenshtein Distance differs from Levenshtein Distance in that it*	
5		**Syntax**			*includes an additional operation, called Transpositions, which*	
6		*=DAMERAU(string1,string2)*			*occurs when two adjacent characters are swapped. Thus, Damerau-*	
7					*Levenshtein Distance calculates the number of Insertions, Deletions,*	
8					*Substitutions, and Transpositons needed to convert string1 into*	
9					*string2. As a result, this function is good when it is likely that*	
10					*spelling errors have occured between two string where the error is*	
11					*simply a transposition of 2 adjacent characters.*	
12						
13		**Examples**			**Result**	**Formula**
14		*apple*	*maplle*		*2*	*=DAMERAU(B14,C14)*
15		*apple*	*applet*		*1*	*=DAMERAU(B15,C15)*
16		*apple*	*maple*		*2*	*=DAMERAU(B16,C16)*
17		*apple*	*mappppple*		*3*	*=DAMERAU(B17,C17)*
18						

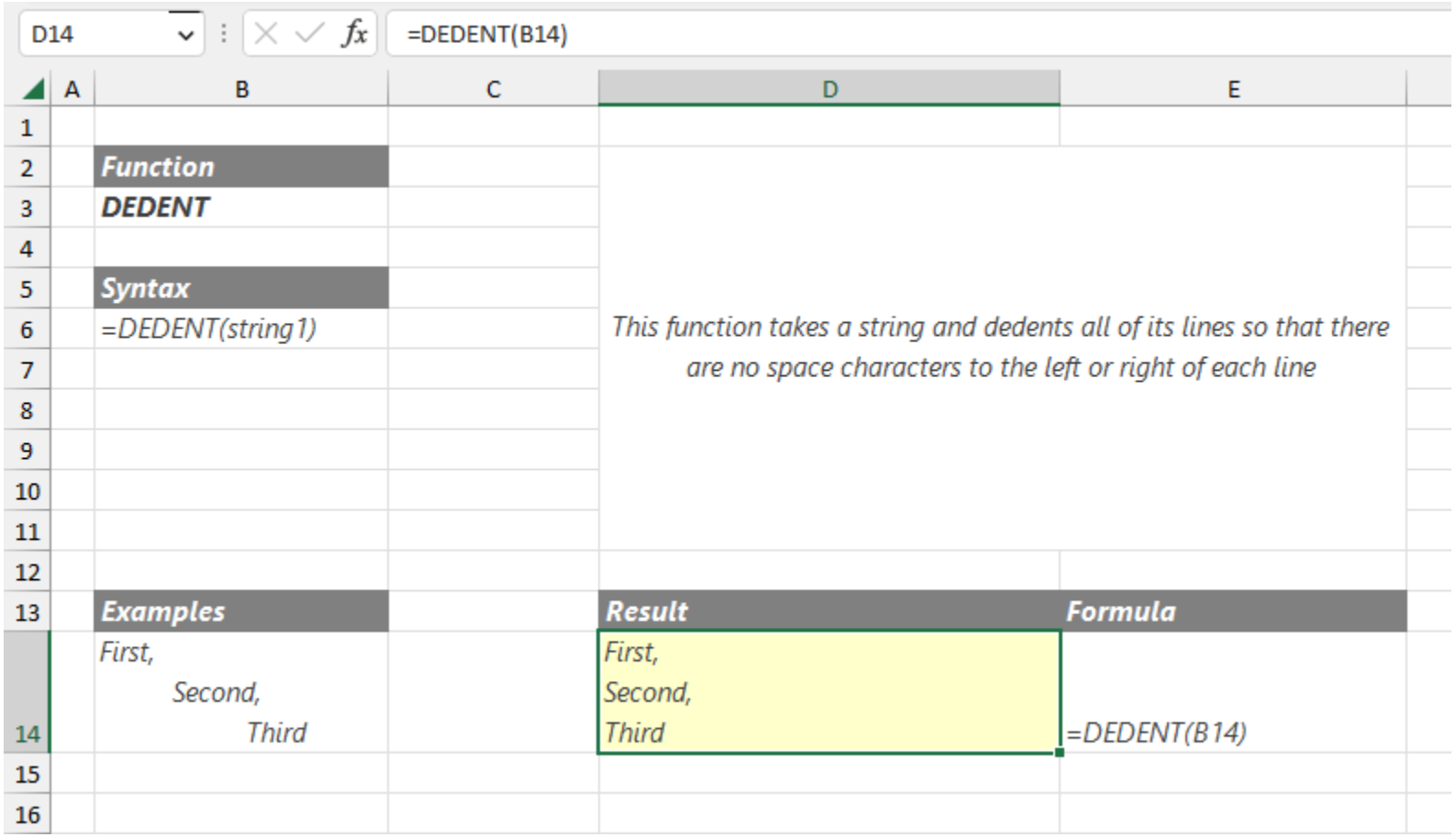
DASH_CASE



D14 =DASH_CASE(B14)

	A	B	C	D	E	F
1						
2		**Function**				
3		**DASH_CASE**				
4						
5		**Syntax**		*Dash Case (Kebab Case) removes punctuation and replaces spaces with a single hyphen (-). It is also known as the spinal case, param case, Lisp case, and dash case.*		
6		=DASH_CASE(string1)				
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		Hello World		hello-world	=DASH_CASE(B14)	
15		This is a test string.		this-is-a-test-string.	=DASH_CASE(B15)	
16						
17						
18						

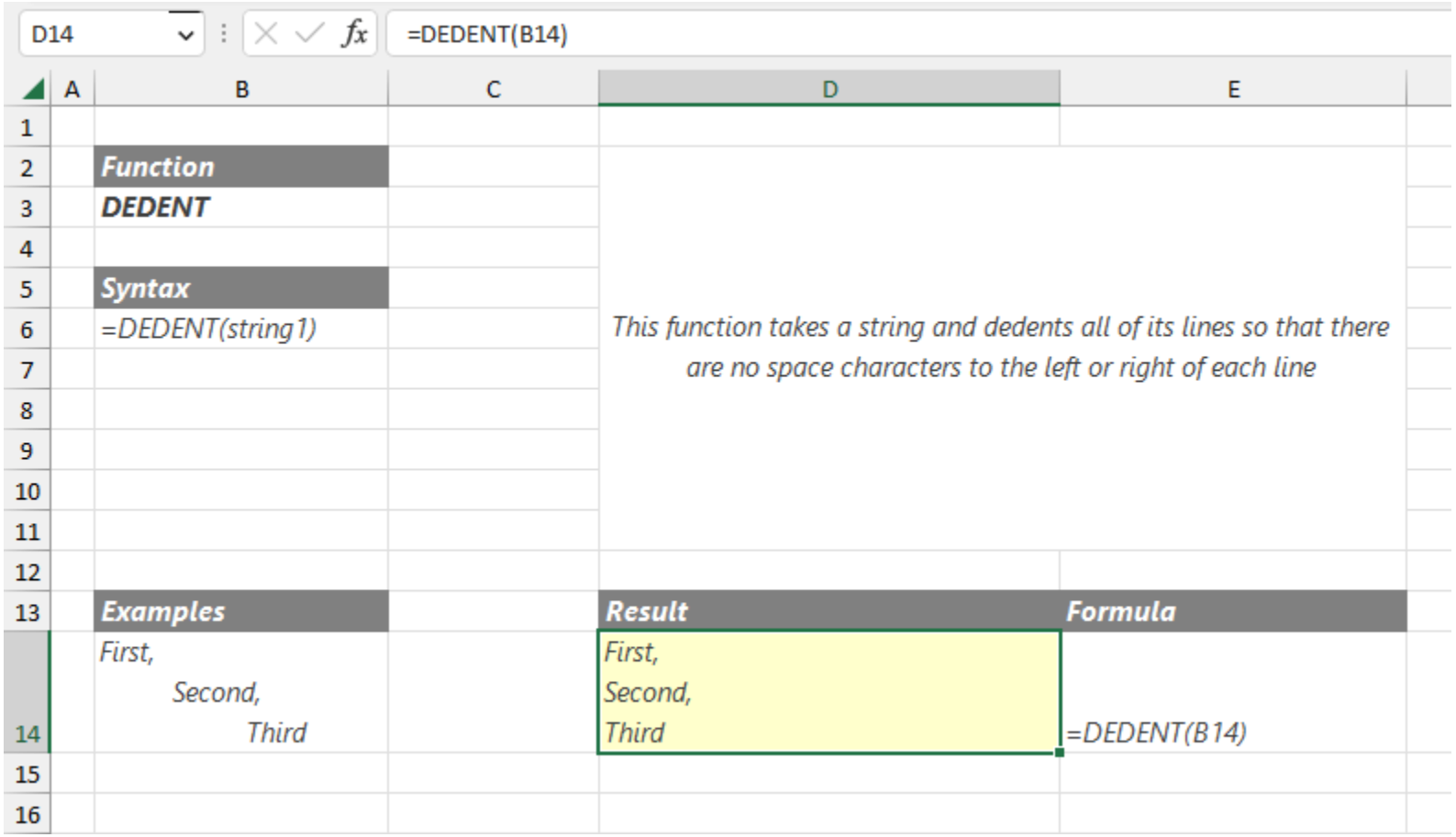
DEDENT



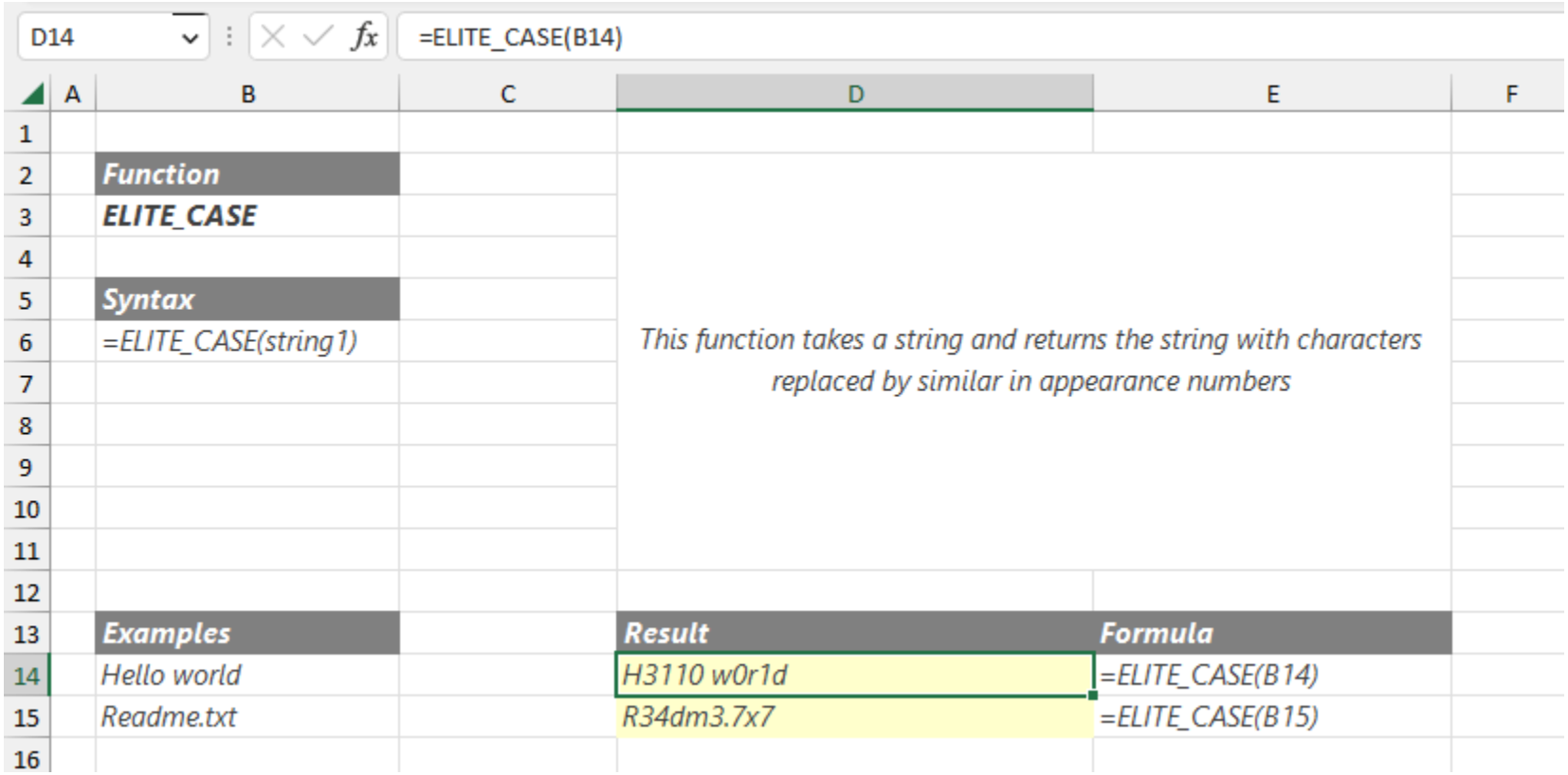
D14 | =DEDENT(B14)

	A	B	C	D	E
1					
2		**Function**			
3		**DEDENT**			
4					
5		**Syntax**			
6		=DEDENT(string1)		*This function takes a string and dedents all of its lines so that there are no space characters to the left or right of each line*	
7					
8					
9					
10					
11					
12					
13		**Examples**		**Result**	**Formula**
14		First, Second, Third		First, Second, Third	=DEDENT(B14)
15					
16					

DELIMSTR


D14
=DEDENT(B14)
Function
DEDENT
Syntax
=DEDENT(string1)
This function takes a string and dedents all of its lines so that there are no space characters to the left or right of each line
Examples
Result
Formula
First,
Second,
Third
First,
Second,
Third
=DEDENT(B14)

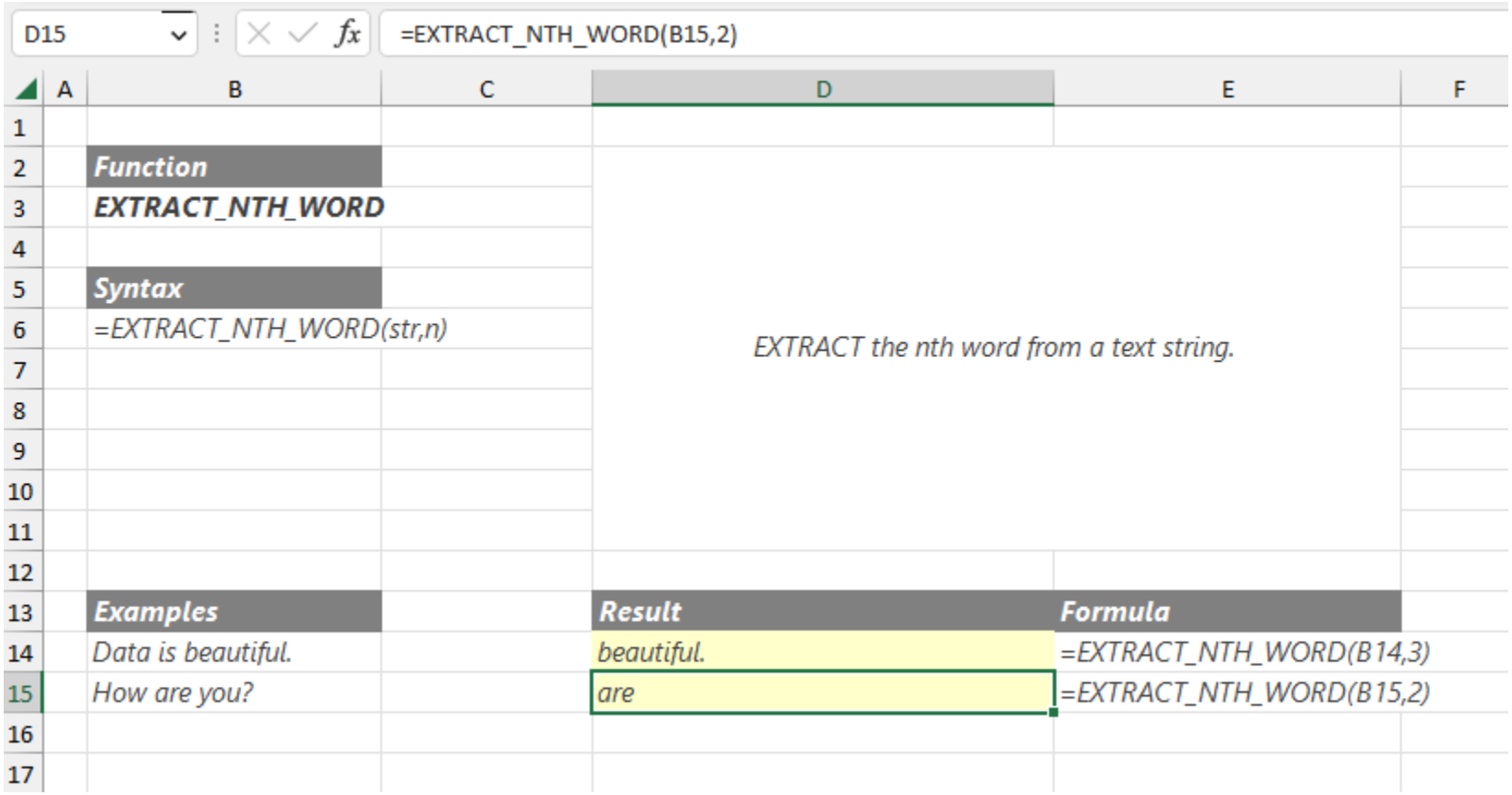
ELITE_CASE



D14 | =ELITE_CASE(B14)

	A	B	C	D	E	F
1						
2		**Function**		*This function takes a string and returns the string with characters replaced by similar in appearance numbers*		
3		**ELITE_CASE**				
4						
5		**Syntax**				
6		=ELITE_CASE(string1)				
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		Hello world		H3110 w0r1d	=ELITE_CASE(B14)	
15		Readme.txt		R34dm3.7x7	=ELITE_CASE(B15)	
16						

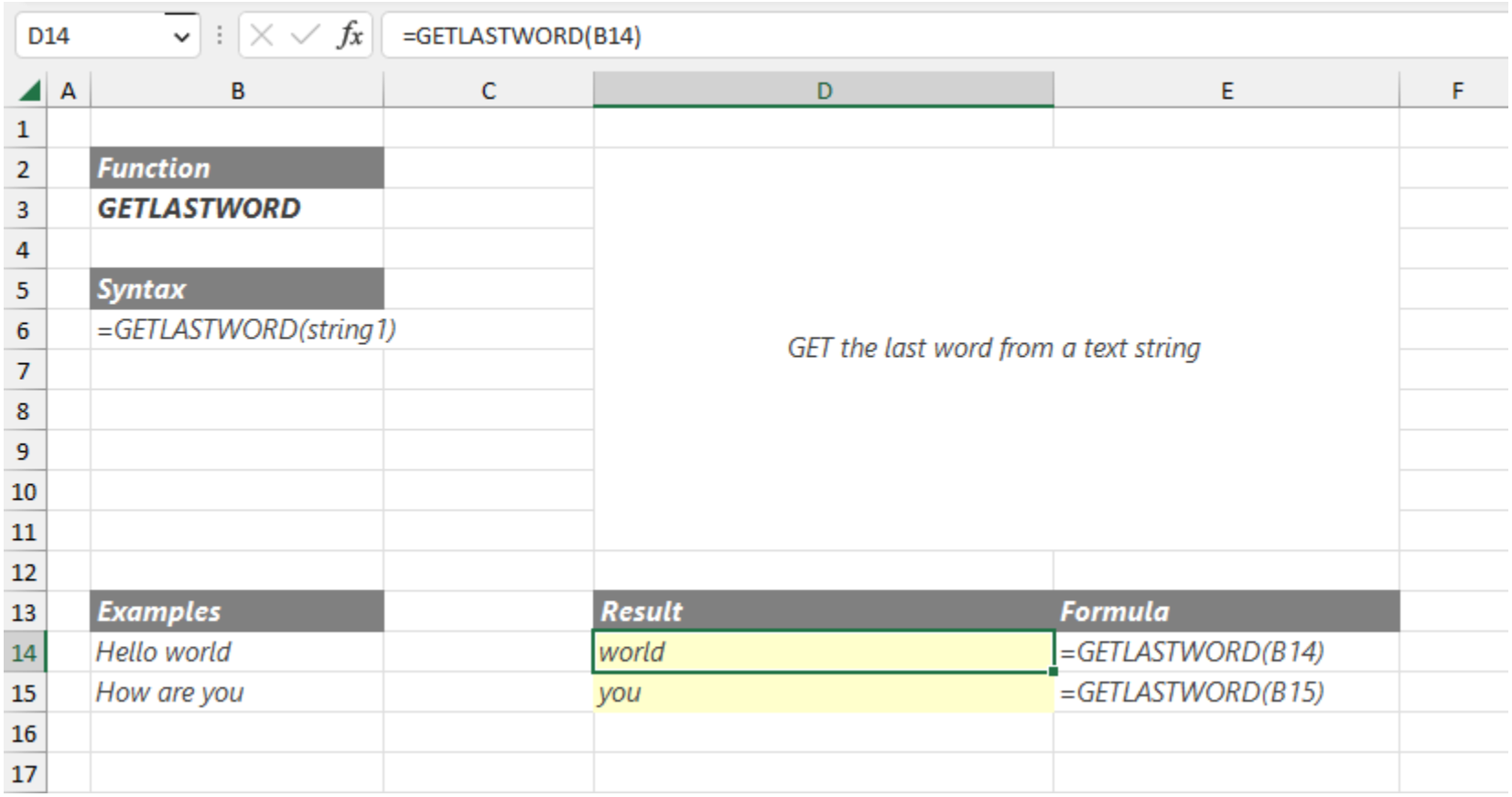
EXTRACT_NTH_WORD



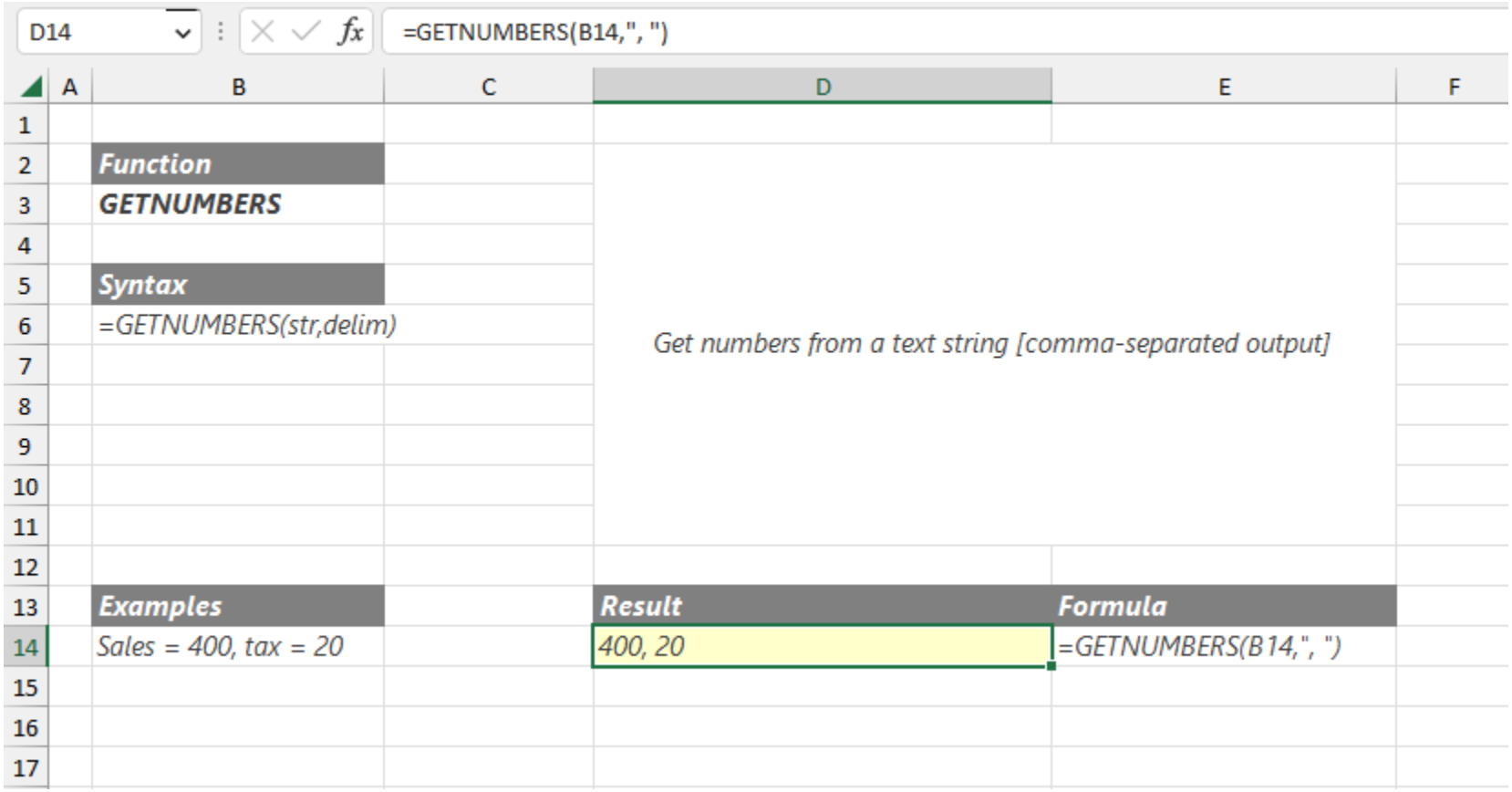
	A	B	C	D	E	F
1						
2		**Function**				
3		**EXTRACT_NTH_WORD**				
4						
5		**Syntax**				
6		=EXTRACT_NTH_WORD(str,n)				
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		Data is beautiful.		beautiful.	=EXTRACT_NTH_WORD(B14,3)	
15		How are you?		are	=EXTRACT_NTH_WORD(B15,2)	
16						
17						

EXTRACT the nth word from a text string.

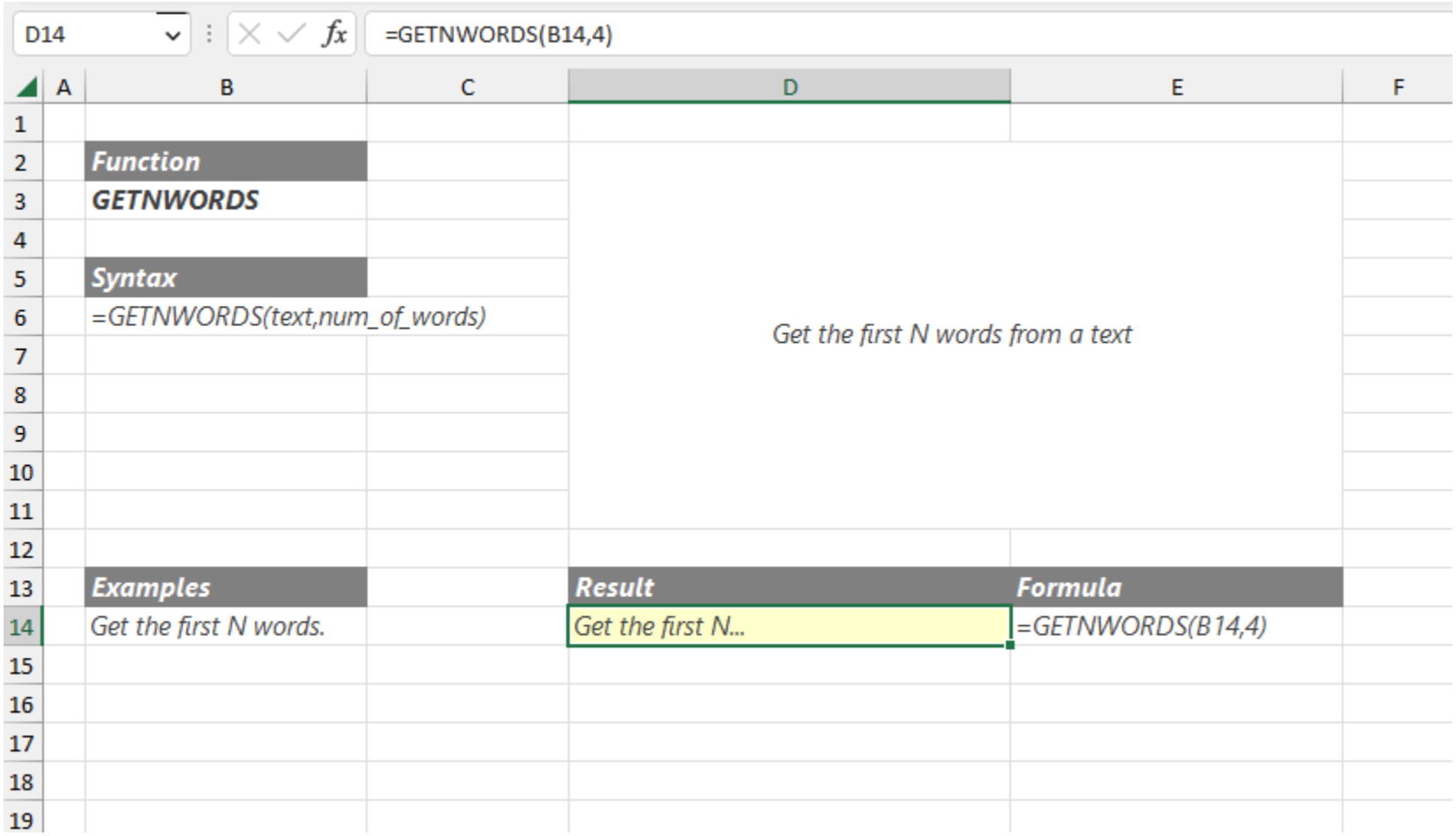
GETLASTWORD



GETNUMBERS


D14
=GETNUMBERS(B14,", ")
Function
GETNUMBERS
Syntax
=GETNUMBERS(str,delim)
Get numbers from a text string [comma-separated output]
Examples
Result
Formula
Sales = 400, tax = 20
400, 20
=GETNUMBERS(B14,", ")

GETNWORDS

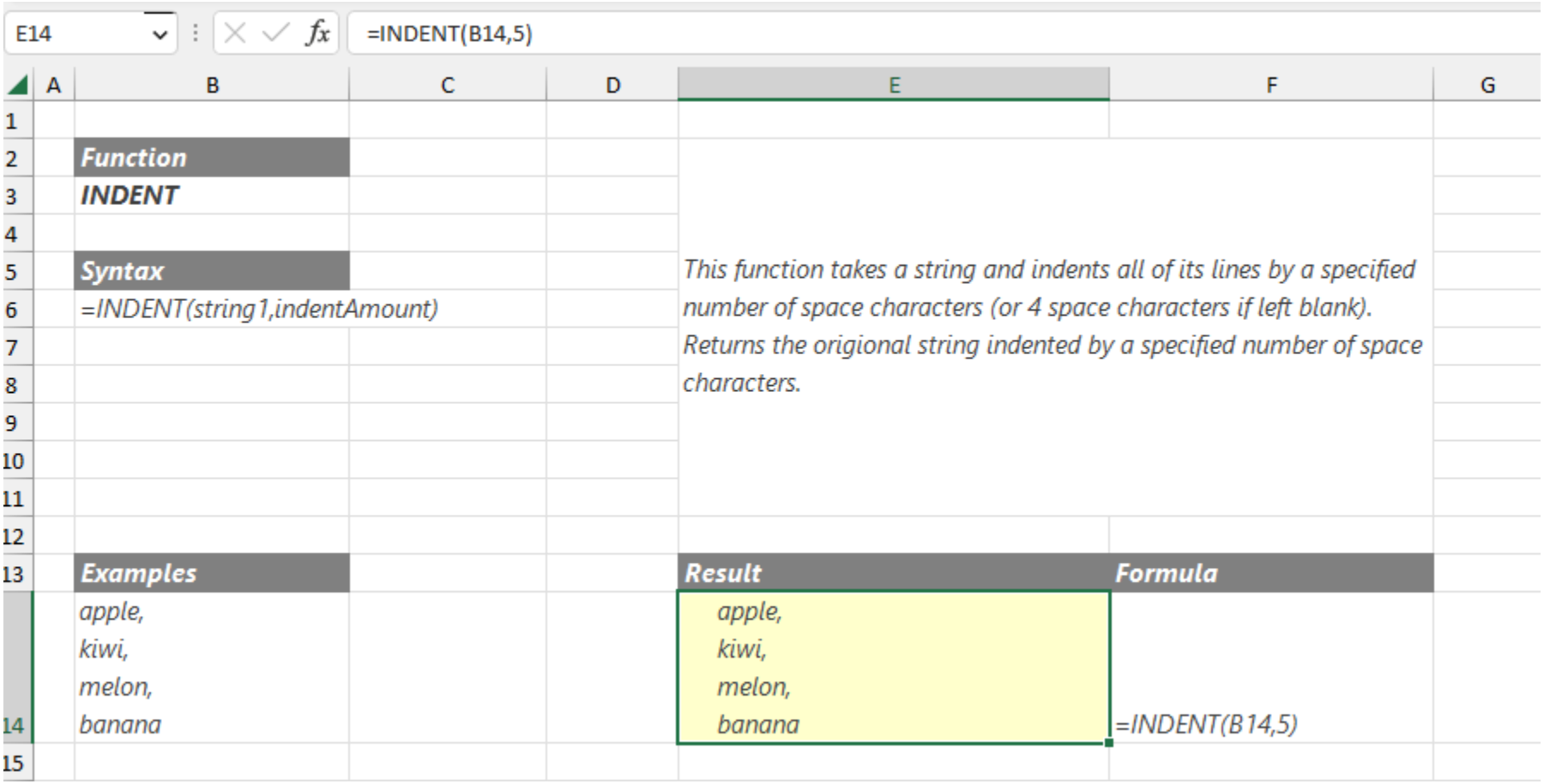

D14
=GETNWORDS(B14,4)
Function
GETNWORDS
Syntax
=GETNWORDS(text,num_of_words)
Get the first N words from a text
Examples
Result
Formula
Get the first N words.
Get the first N...
=GETNWORDS(B14,4)

HAMMING

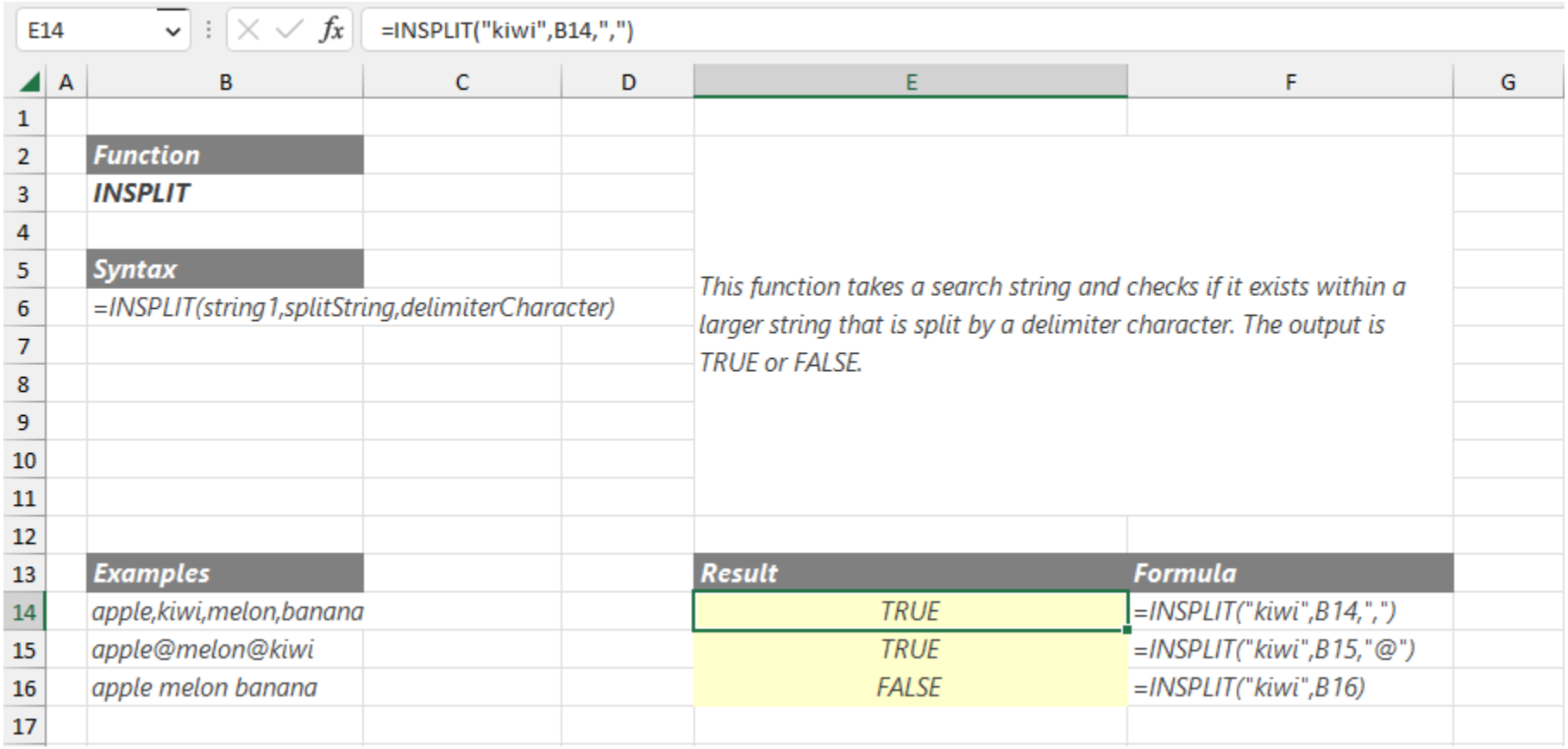
E14 =HAMMING(B14,C14)

	A	B	C	D	E	F	G
1							
2		**Function**					
3		**HAMMING**			*This function* ***takes two strings of the same length*** *and*		
4					*calculates the Hamming Distance between them.*		
5		**Syntax**					
6		*=HAMMING(string1,string2)*			*Hamming Distance measures how close two strings are by*		
7					*checking how many Substitutions are needed to turn one string into*		
8					*the other. Lower numbers mean the strings are closer than high*		
9					*numbers. Returns an integer of the Hamming Distance between two*		
10					*string.*		
11							
12							
13		**Examples**			**Result**	**Formula**	
14		*lego*	*gleo*		*3*	*=HAMMING(B14,C14)*	
15		*apple*	*pleap*		*5*	*=HAMMING(B15,C15)*	
16		*apple*	*maple*		*2*	*=HAMMING(B16,C16)*	
17		*excel*	*aaaaa*		*5*	*=HAMMING(B17,C17)*	
18							
19							

INDENT


E14
=INDENT(B14,5)
Function
INDENT
Syntax
=INDENT(string1,indentAmount)
This function takes a string and indents all of its lines by a specified number of space characters (or 4 space characters if left blank). Returns the origional string indented by a specified number of space characters.
Examples
Result
Formula
apple,
kiwi,
melon,
banana
apple,
kiwi,
melon,
banana
=INDENT(B14,5)

INSPLIT

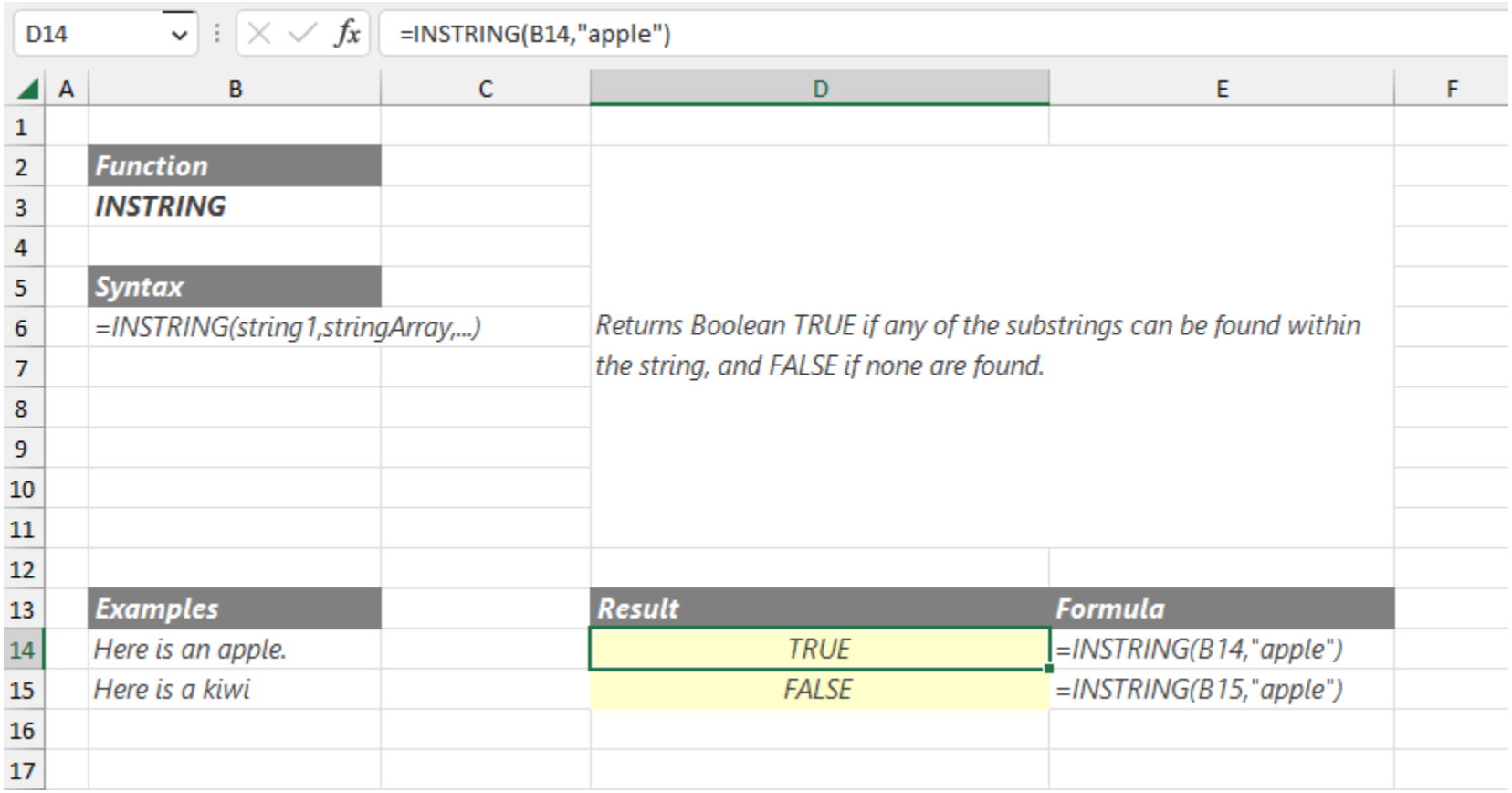


E14 =INSPLIT("kiwi",B14,",")

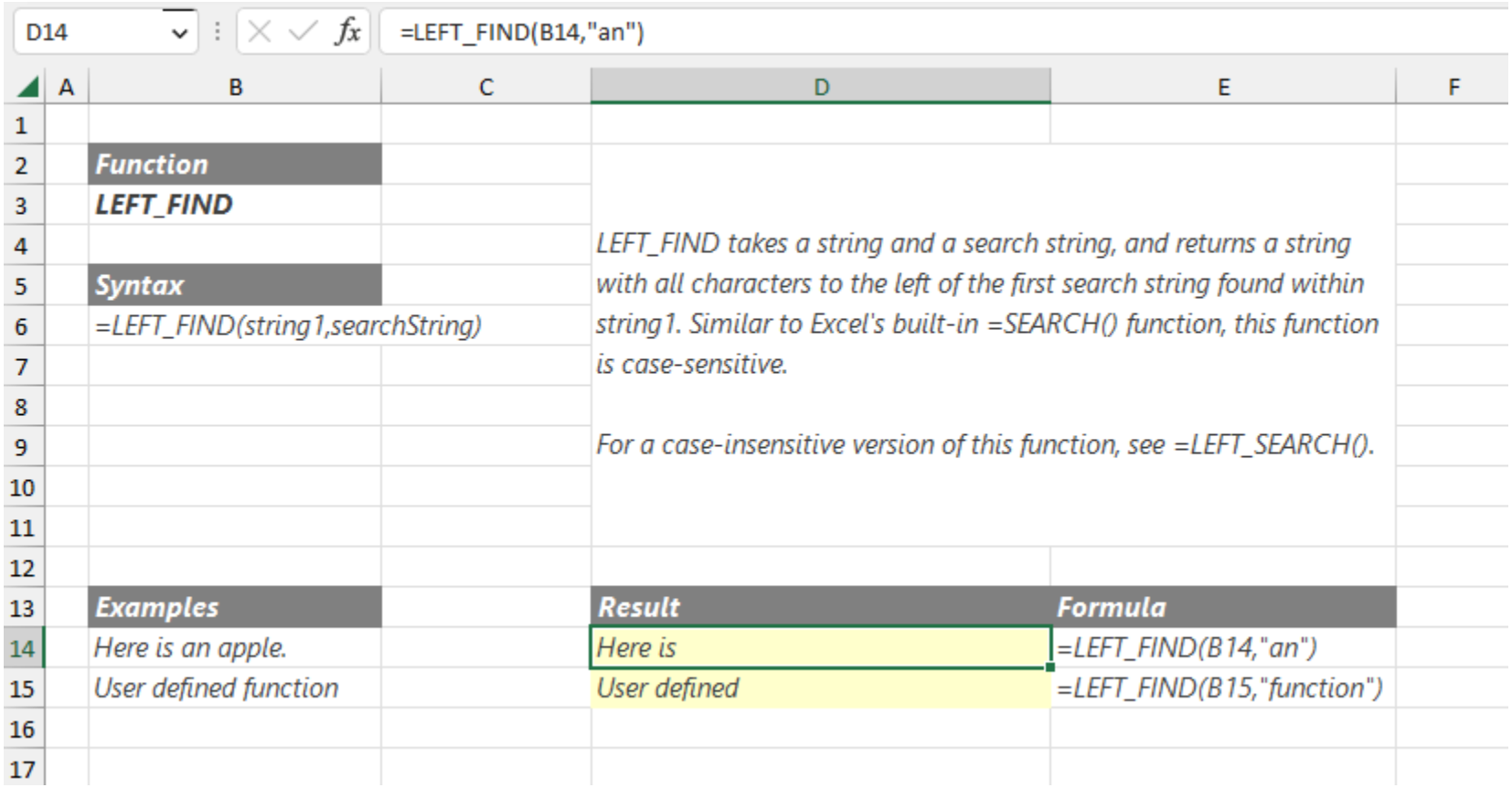
	A	B	C	D	E	F	G
1							
2		**Function**					
3		**INSPLIT**					
4							
5		**Syntax**					
6		=INSPLIT(string1,splitString,delimiterCharacter)					
7							
8							
9							
10							
11							
12							
13		**Examples**			**Result**	**Formula**	
14		apple,kiwi,melon,banana			TRUE	=INSPLIT("kiwi",B14,",")	
15		apple@melon@kiwi			TRUE	=INSPLIT("kiwi",B15,"@")	
16		apple melon banana			FALSE	=INSPLIT("kiwi",B16)	
17							

This function takes a search string and checks if it exists within a larger string that is split by a delimiter character. The output is TRUE or FALSE.

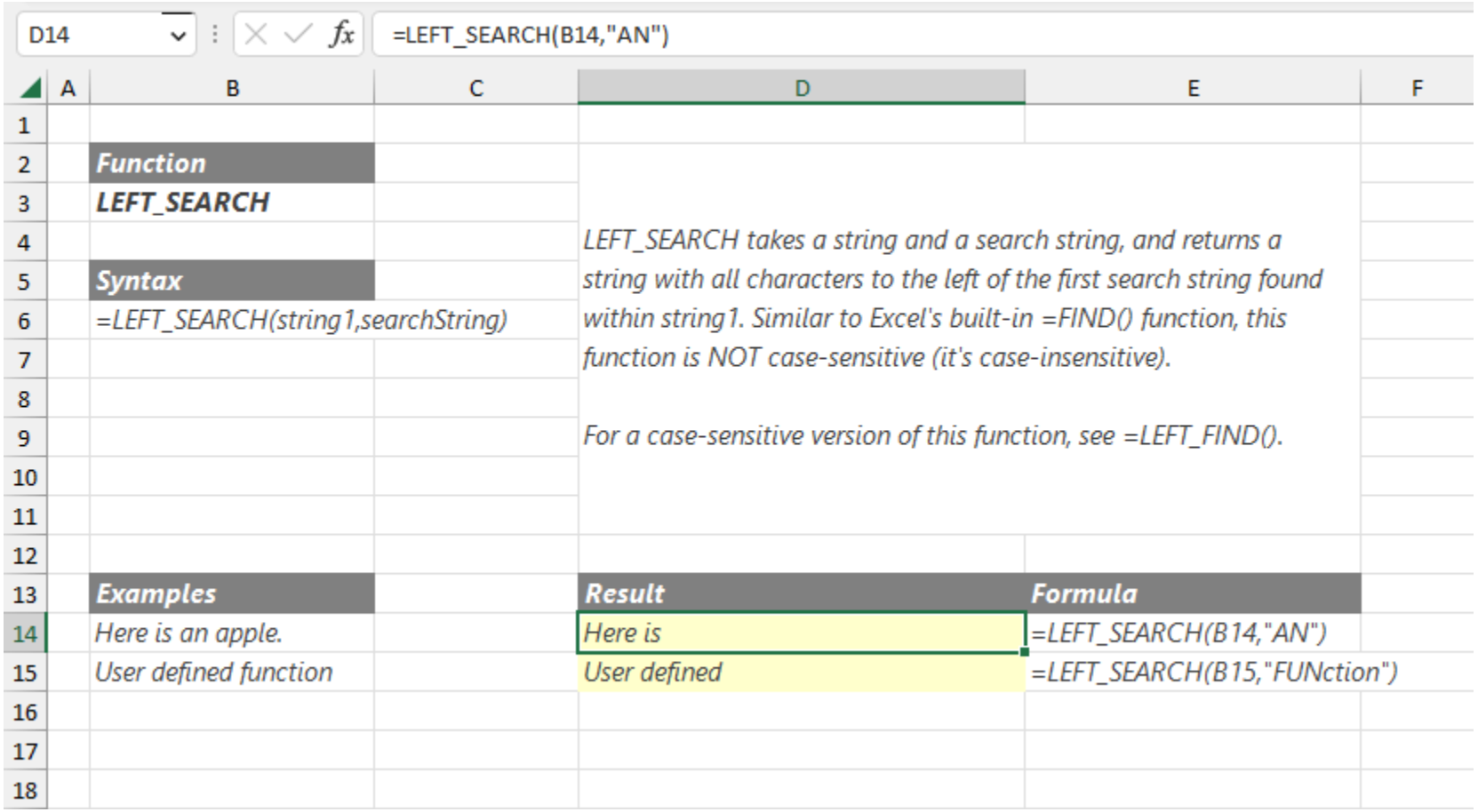
INSTRING


D14
=INSTRING(B14,"apple")
Function
INSTRING
Syntax
=INSTRING(string1,stringArray,...)
Returns Boolean TRUE if any of the substrings can be found within the string, and FALSE if none are found.
Examples
Result
Formula
Here is an apple.
TRUE
=INSTRING(B14,"apple")
Here is a kiwi
FALSE
=INSTRING(B15,"apple")

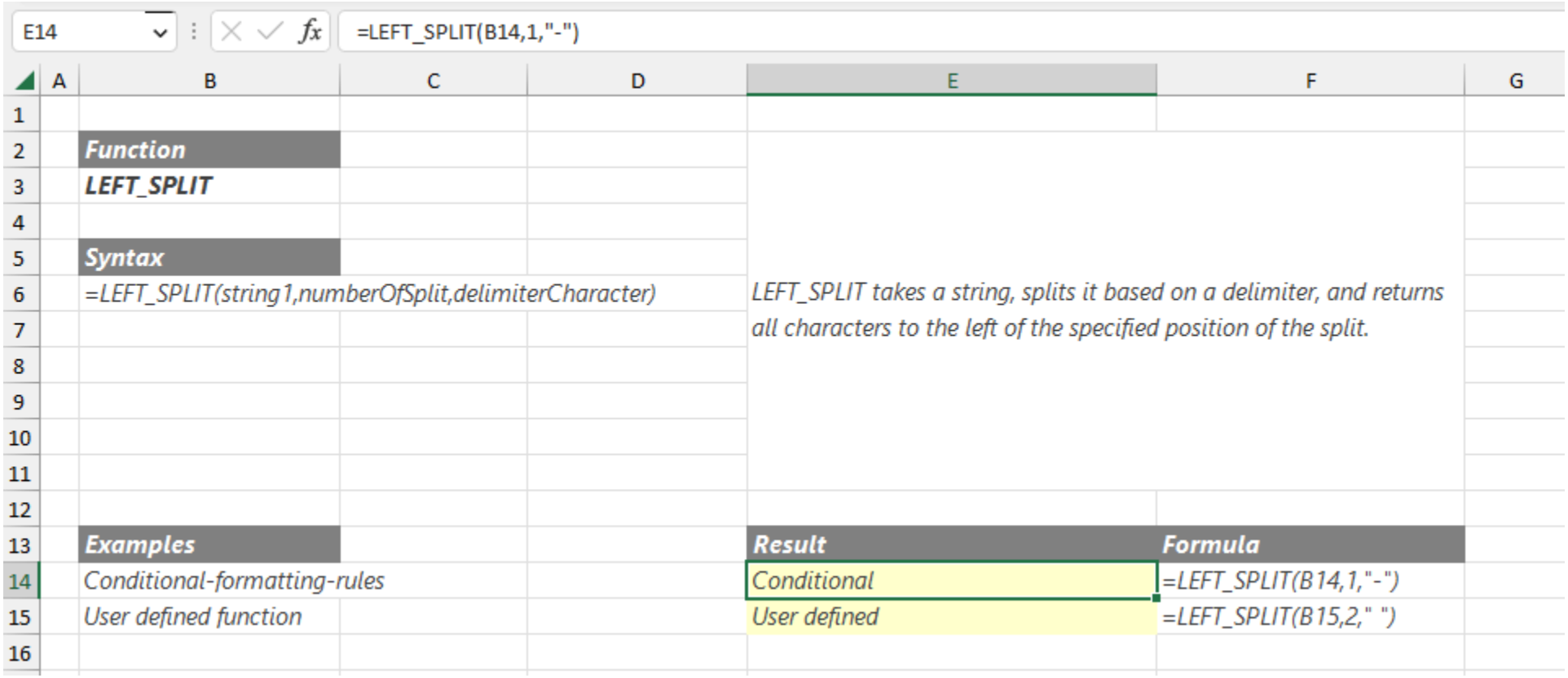
LEFT_FIND


D14
=LEFT_FIND(B14,"an")
Function
LEFT_FIND
Syntax
=LEFT_FIND(string1,searchString)
LEFT_FIND takes a string and a search string, and returns a string with all characters to the left of the first search string found within string1. Similar to Excel's built-in =SEARCH() function, this function is case-sensitive.
For a case-insensitive version of this function, see =LEFT_SEARCH().
Examples
Result
Formula
Here is an apple.
Here is
=LEFT_FIND(B14,"an")
User defined function
User defined
=LEFT_FIND(B15,"function")

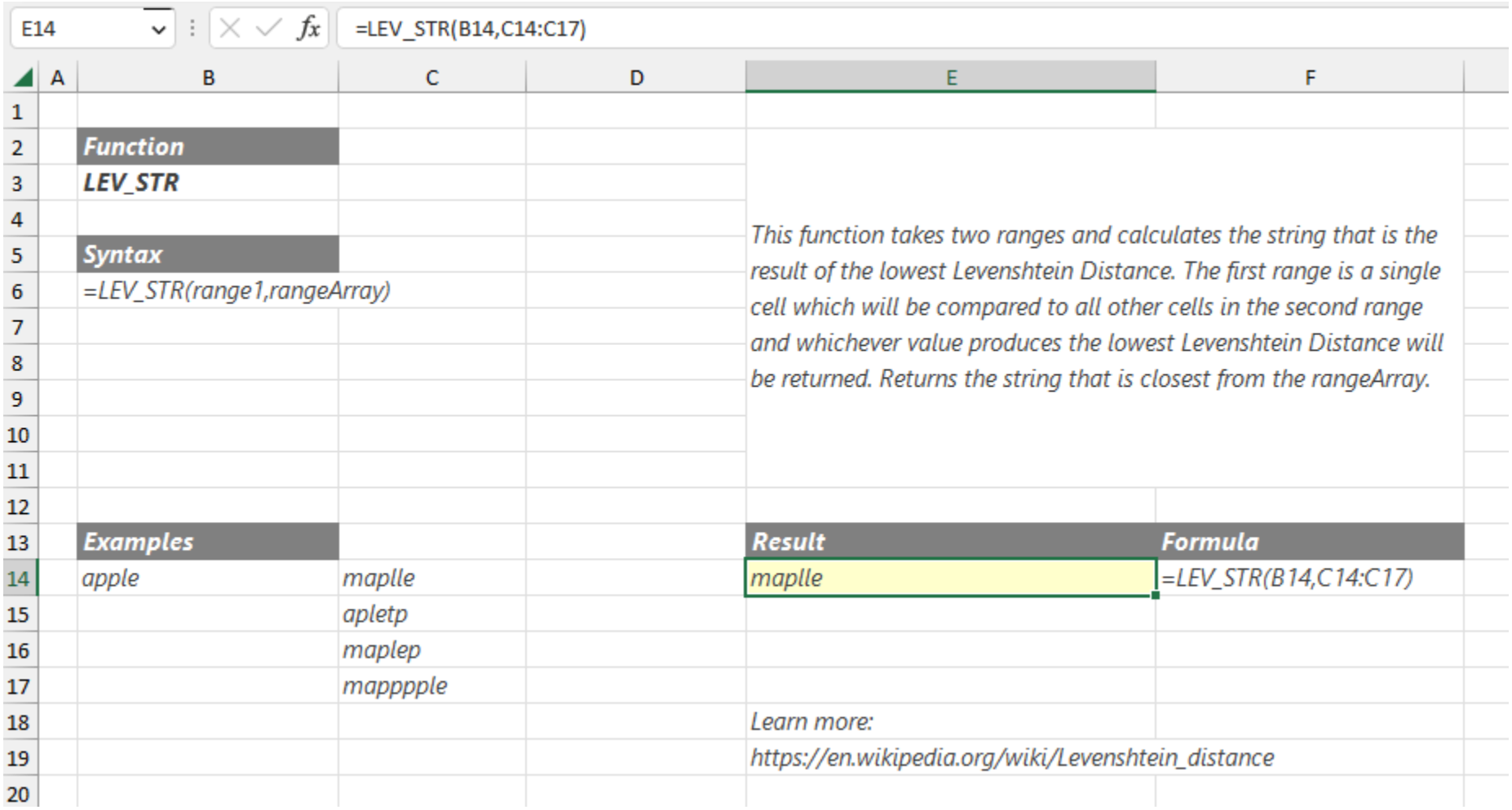
LEFT_SEARCH


D14
=LEFT_SEARCH(B14,"AN")
Function
LEFT_SEARCH
Syntax
=LEFT_SEARCH(string1,searchString)
LEFT_SEARCH takes a string and a search string, and returns a string with all characters to the left of the first search string found within string1. Similar to Excel's built-in =FIND() function, this function is NOT case-sensitive (it's case-insensitive).
For a case-sensitive version of this function, see =LEFT_FIND().
Examples
Result
Formula
Here is an apple.
Here is
=LEFT_SEARCH(B14,"AN")
User defined function
User defined
=LEFT_SEARCH(B15,"FUNction")

LEFT_SPLIT


E14
=LEFT_SPLIT(B14,1,"-")
Function
LEFT_SPLIT
Syntax
=LEFT_SPLIT(string1,numberOfSplit,delimiterCharacter)
LEFT_SPLIT takes a string, splits it based on a delimiter, and returns all characters to the left of the specified position of the split.
Examples
Result
Formula
Conditional-formatting-rules
Conditional
=LEFT_SPLIT(B14,1,"-")
User defined function
User defined
=LEFT_SPLIT(B15,2," ")

LEV_STR


E14
=LEV_STR(B14,C14:C17)
Function
LEV_STR
Syntax
=LEV_STR(range1,rangeArray)
This function takes two ranges and calculates the string that is the result of the lowest Levenshtein Distance. The first range is a single cell which will be compared to all other cells in the second range and whichever value produces the lowest Levenshtein Distance will be returned. Returns the string that is closest from the rangeArray.
Examples
apple
maplle
apletp
maplep
mapppple
Result
maplle
Formula
=LEV_STR(B14,C14:C17)
Learn more:
https://en.wikipedia.org/wiki/Levenshtein_distance

LEVENSHTEIN

E14 =LEVENSHTEIN(B14,C14)

	A	B	C	D	E	F	G
1							
2		**Function**			*This function takes two strings of any length and calculates the Levenshtein Distance between them. Levenshtein Distance measures how close two strings are by checking how many Insertions, Deletions, or Substitutions are needed to turn one string into the other. Lower numbers mean the strings are closer than high numbers. Unlike Hamming Distance, Levenshtein Distance works for strings of any length and includes 2 more operations. However, calculation time will be slower than Hamming Distance for same length strings, so if you know the two strings are the same length, its preferred to use Hamming Distance. Returns an integer of the Levenshtein Distance between two string.*		
3		**LEVENSHTEIN**					
4							
5		**Syntax**					
6		*=LEVENSHTEIN(string1,string2)*					
7							
8							
9							
10							
11							
12							
13		**Examples**			**Result**	**Formula**	
14		*apple*	*maplle*		*2*	*=LEVENSHTEIN(B14,C14)*	
15		*apple*	*apletp*		*3*	*=LEVENSHTEIN(B15,C15)*	
16		*apple*	*Pxmaplep*		*5*	*=LEVENSHTEIN(B16,C16)*	
17		*apple*	*mappxpple*		*4*	*=LEVENSHTEIN(B17,C17)*	
18							

REMOVE_CHARACTERS

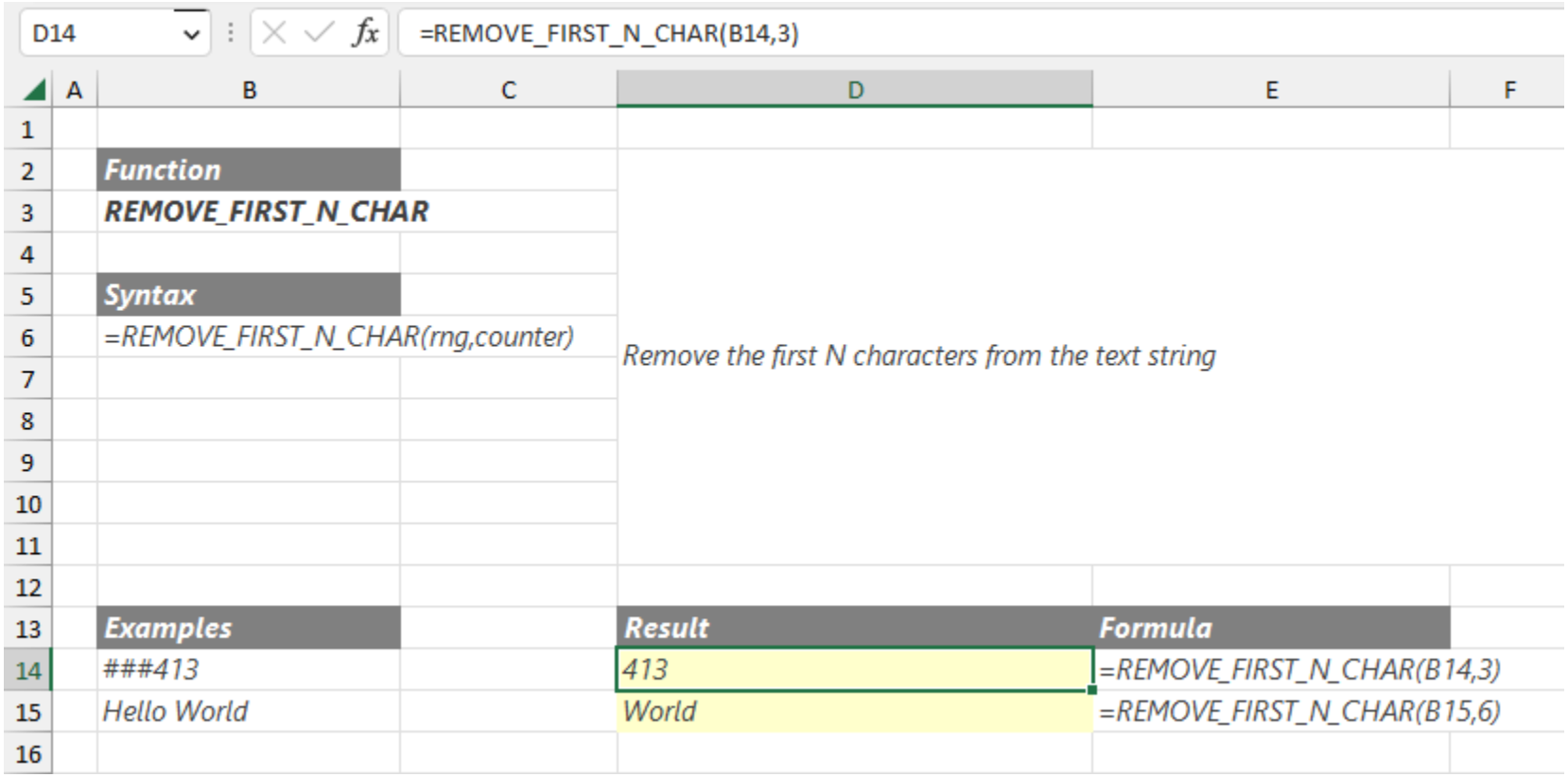
E14 =REMOVE_CHARACTERS(B14,"e","W")

	Function
	REMOVE_CHARACTERS
	Syntax
	=REMOVE_CHARACTERS(string1,removedCharacters,...)

REMOVE_CHARACTERS takes a string and either another string or multiple strings and removes all characters from the first string that are in the second string.

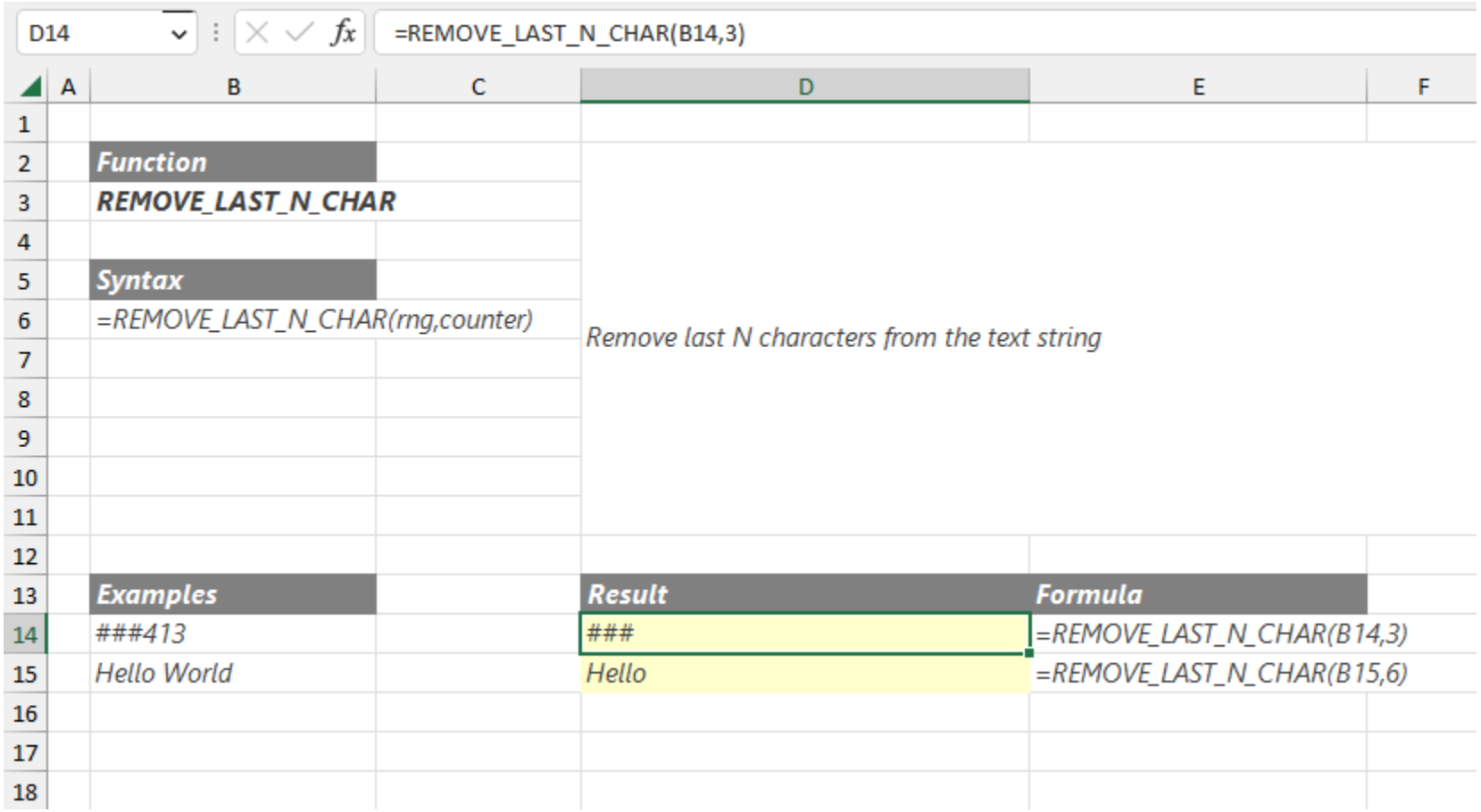
Examples	Result	Formula
Hello World	Hllo orld	=REMOVE_CHARACTERS(B14,"e","W")

REMOVE_FIRST_N_CHAR

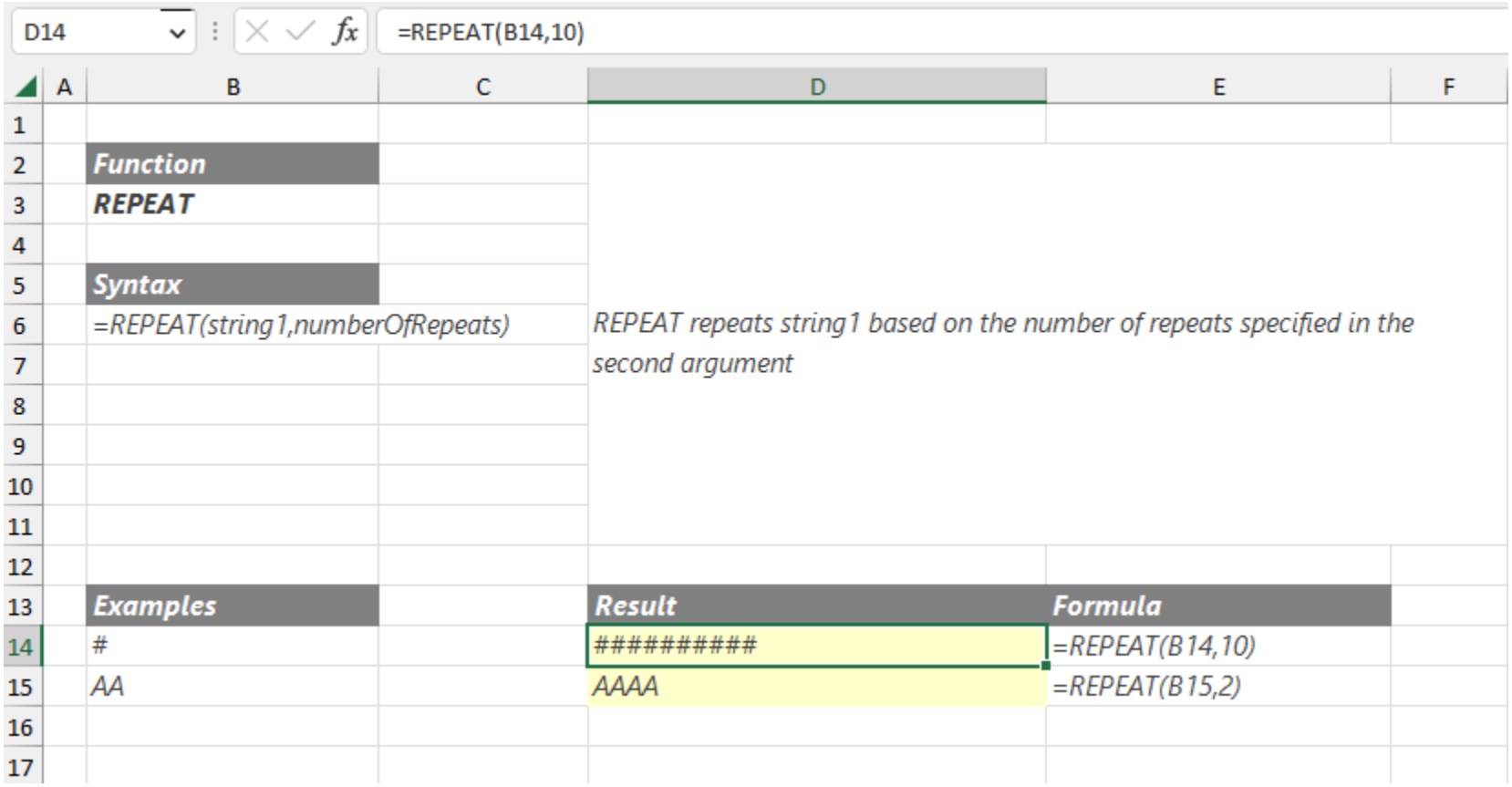


	A	B	C	D	E	F
1						
2		**Function**				
3		**REMOVE_FIRST_N_CHAR**				
4						
5		**Syntax**				
6		=REMOVE_FIRST_N_CHAR(rng,counter)		Remove the first N characters from the text string		
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		###413		413	=REMOVE_FIRST_N_CHAR(B14,3)	
15		Hello World		World	=REMOVE_FIRST_N_CHAR(B15,6)	
16						

REMOVE_LAST_N_CHAR



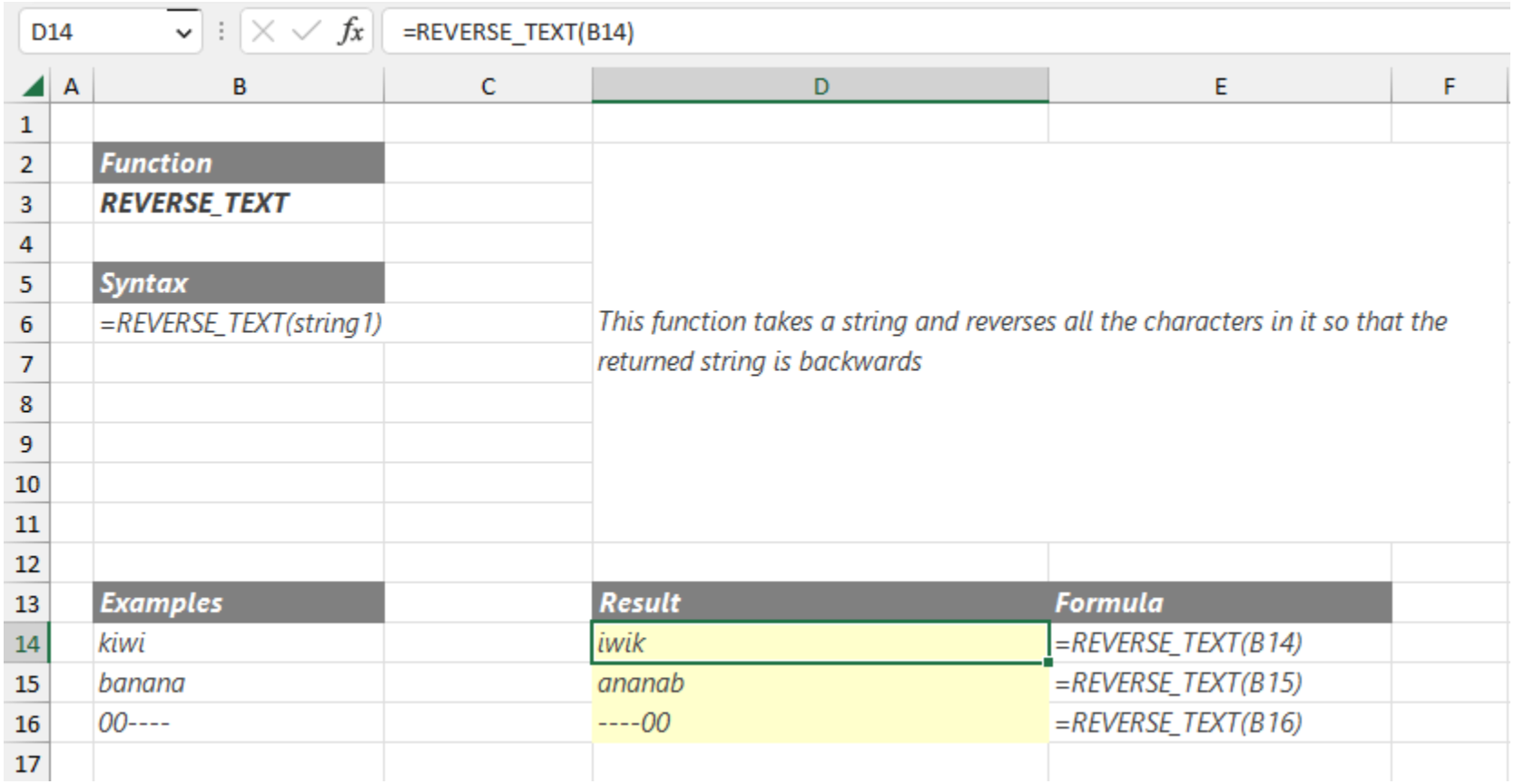
REPEAT



D14 | =REPEAT(B14,10)

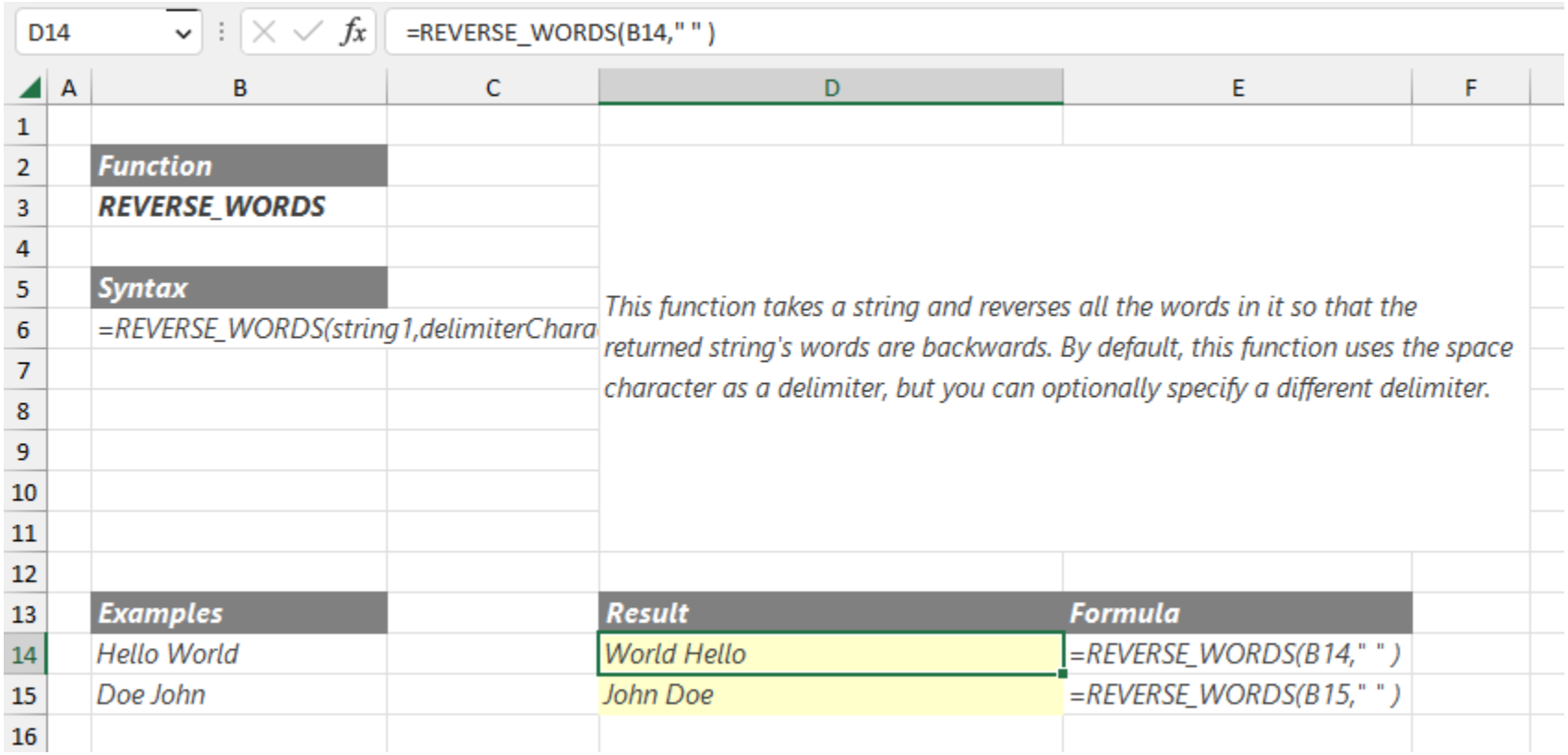
	A	B	C	D	E	F
1						
2		**Function**				
3		**REPEAT**				
4						
5		**Syntax**				
6		=REPEAT(string1,numberOfRepeats)		REPEAT repeats string1 based on the number of repeats specified in the		
7				second argument		
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		#		##########	=REPEAT(B14,10)	
15		AA		AAAA	=REPEAT(B15,2)	
16						
17						

REVERSE_TEXT

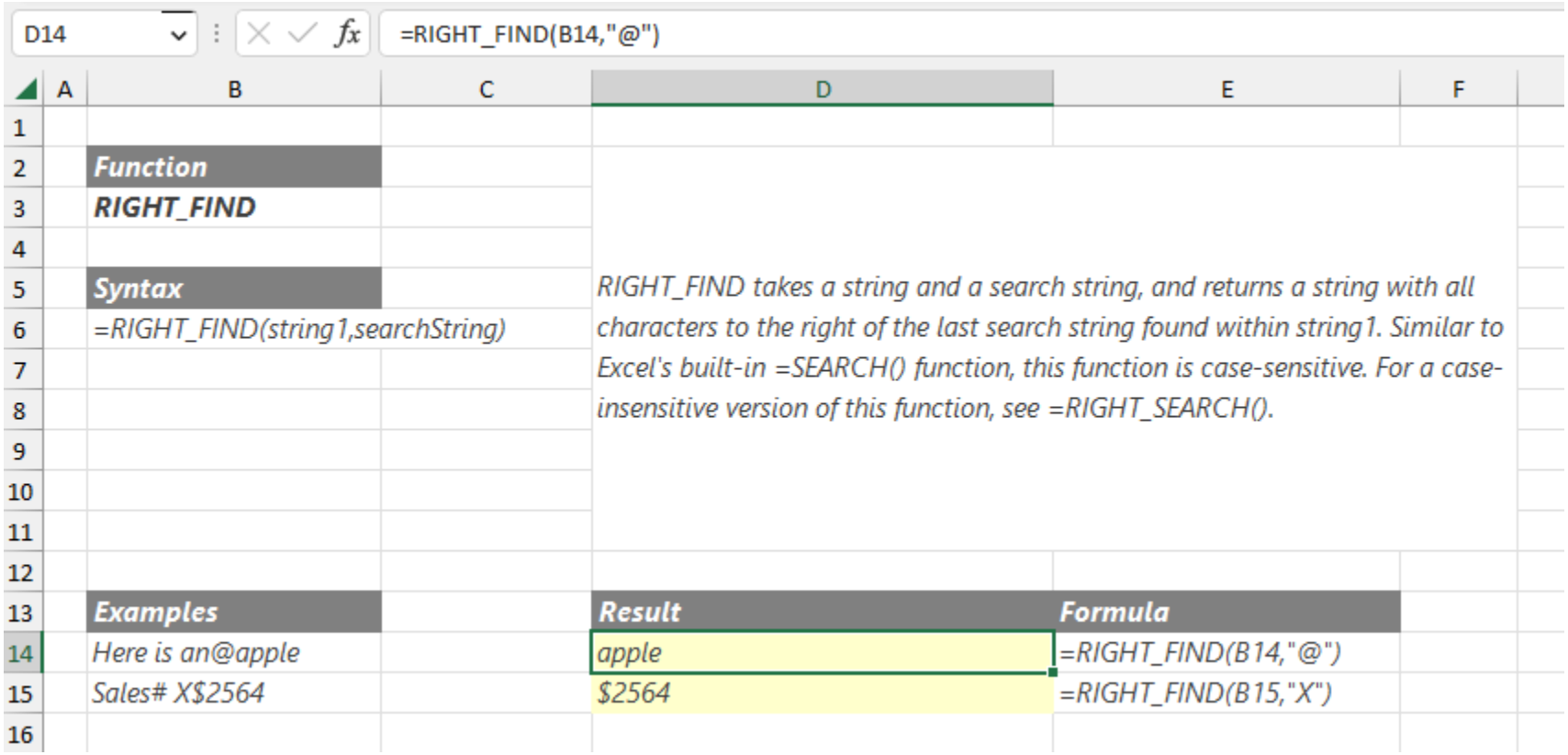


	A	B	C	D	E	F
1						
2		**Function**				
3		**REVERSE_TEXT**				
4						
5		**Syntax**				
6		=REVERSE_TEXT(string1)		This function takes a string and reverses all the characters in it so that the		
7				returned string is backwards		
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		kiwi		iwik	=REVERSE_TEXT(B14)	
15		banana		ananab	=REVERSE_TEXT(B15)	
16		00----		----00	=REVERSE_TEXT(B16)	
17						

REVERSE_WORDS

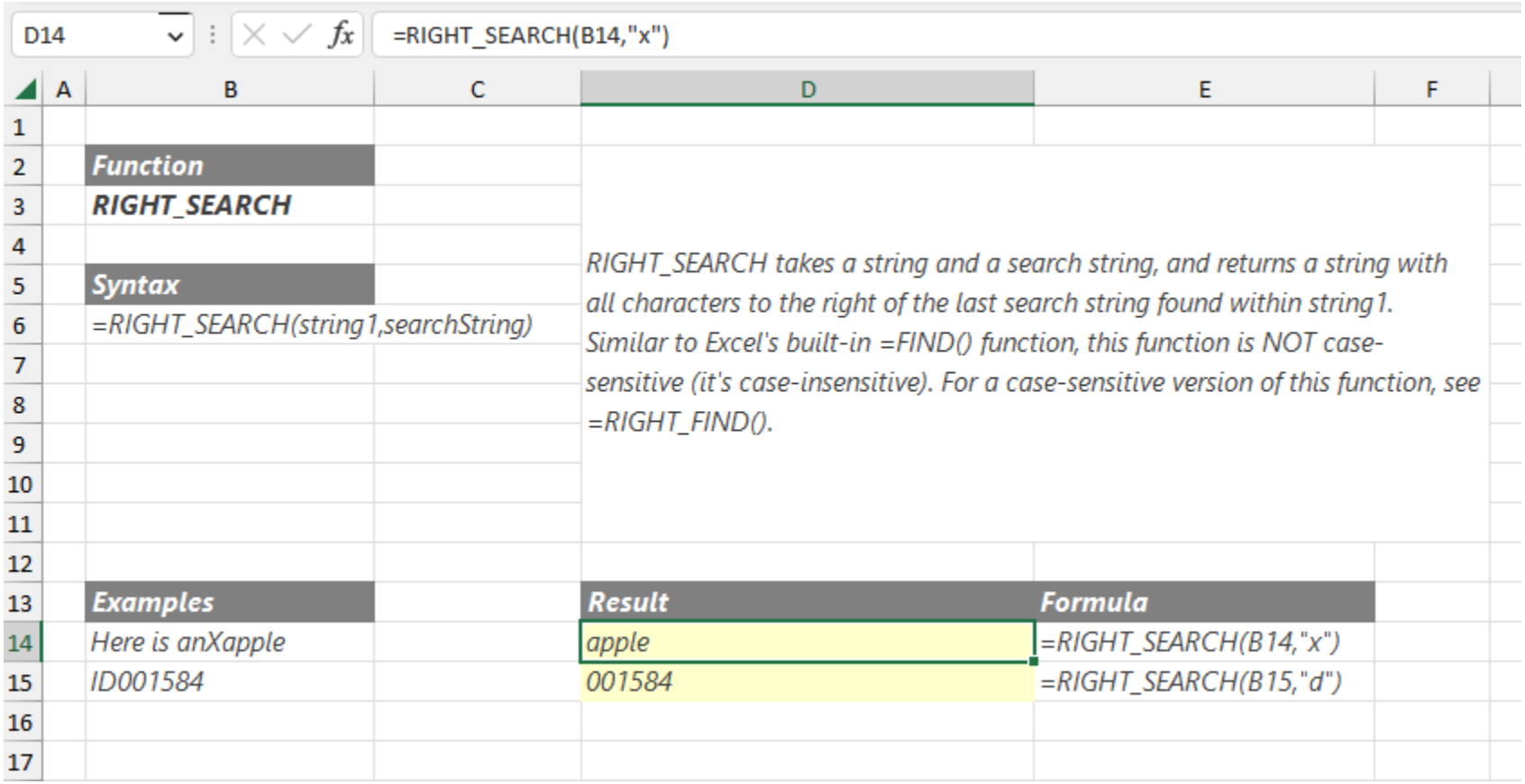


RIGHT_FIND

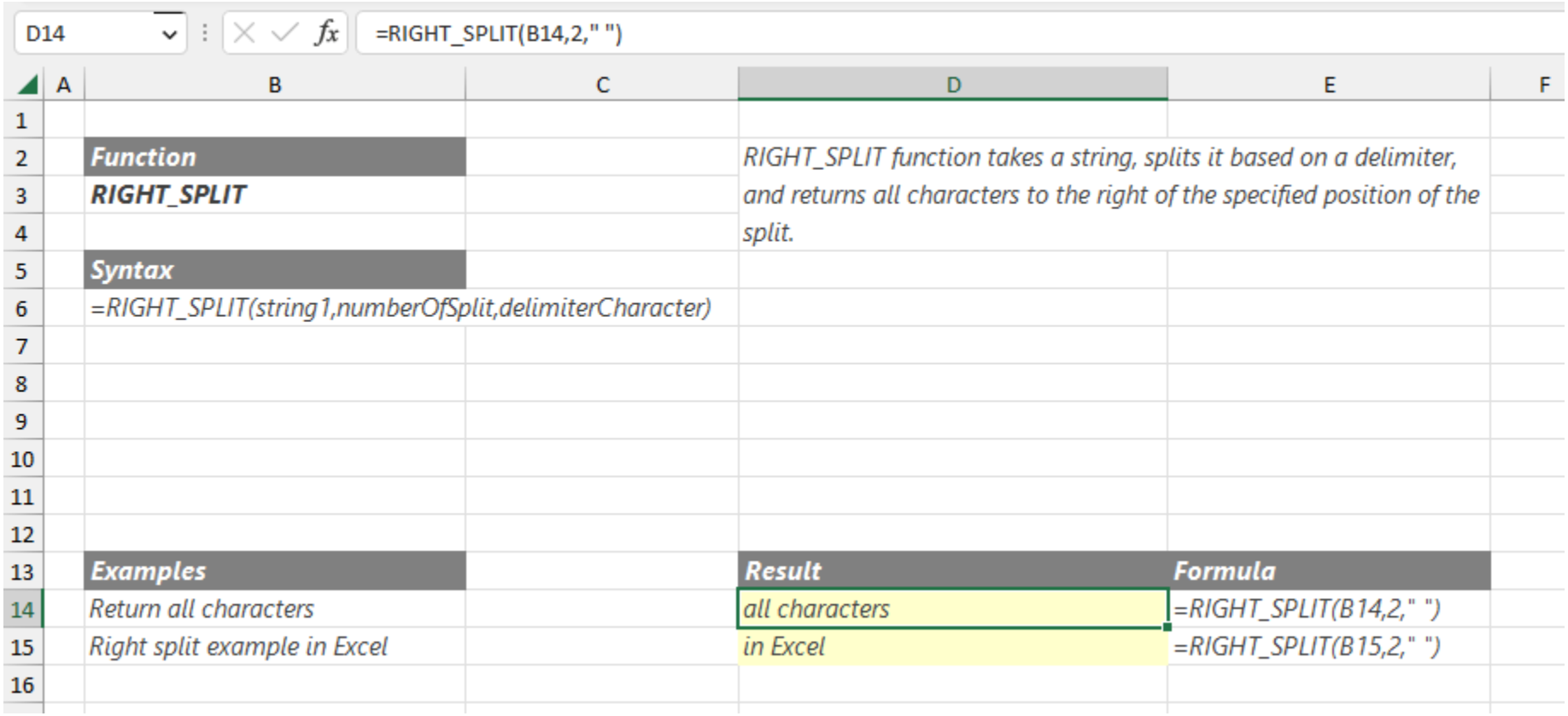


	A	B	C	D	E	F
1						
2		***Function***				
3		***RIGHT_FIND***				
4						
5		***Syntax***		*RIGHT_FIND takes a string and a search string, and returns a string with all*		
6		*=RIGHT_FIND(string1,searchString)*		*characters to the right of the last search string found within string1. Similar to*		
7				*Excel's built-in =SEARCH() function, this function is case-sensitive. For a case-*		
8				*insensitive version of this function, see =RIGHT_SEARCH().*		
9						
10						
11						
12						
13		***Examples***		***Result***	***Formula***	
14		*Here is an@apple*		*apple*	*=RIGHT_FIND(B14,"@")*	
15		*Sales# X$2564*		*$2564*	*=RIGHT_FIND(B15,"X")*	
16						

RIGHT_SEARCH



RIGHT_SPLIT



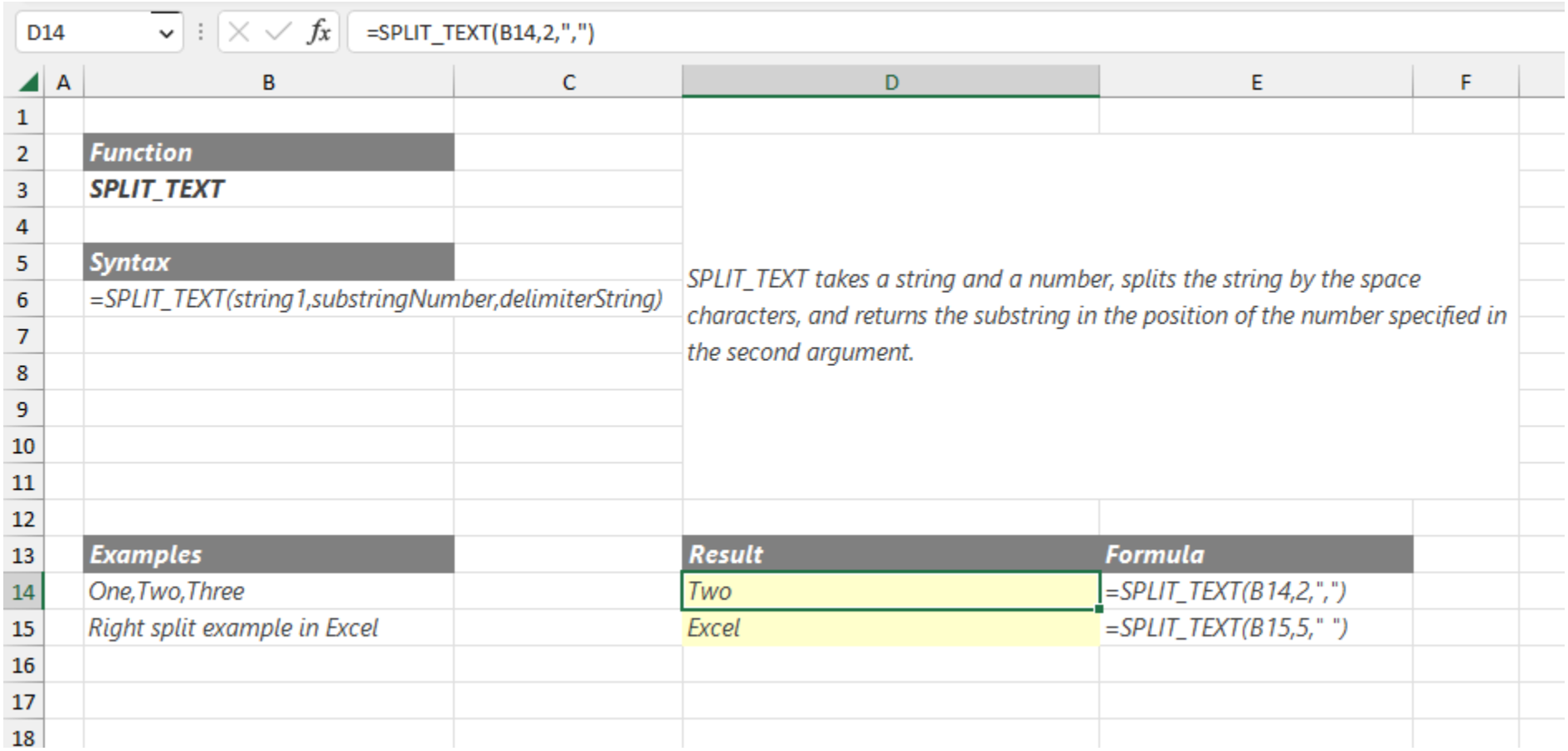
	A	B	C	D	E	F
1						
2		**Function**		*RIGHT_SPLIT function takes a string, splits it based on a delimiter,*		
3		***RIGHT_SPLIT***		*and returns all characters to the right of the specified position of the*		
4				*split.*		
5		**Syntax**				
6		*=RIGHT_SPLIT(string1,numberOfSplit,delimiterCharacter)*				
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		*Return all characters*		*all characters*	*=RIGHT_SPLIT(B14,2," ")*	
15		*Right split example in Excel*		*in Excel*	*=RIGHT_SPLIT(B15,2," ")*	
16						

SHORTEN

D14 =SHORTEN(B14,10,"....")

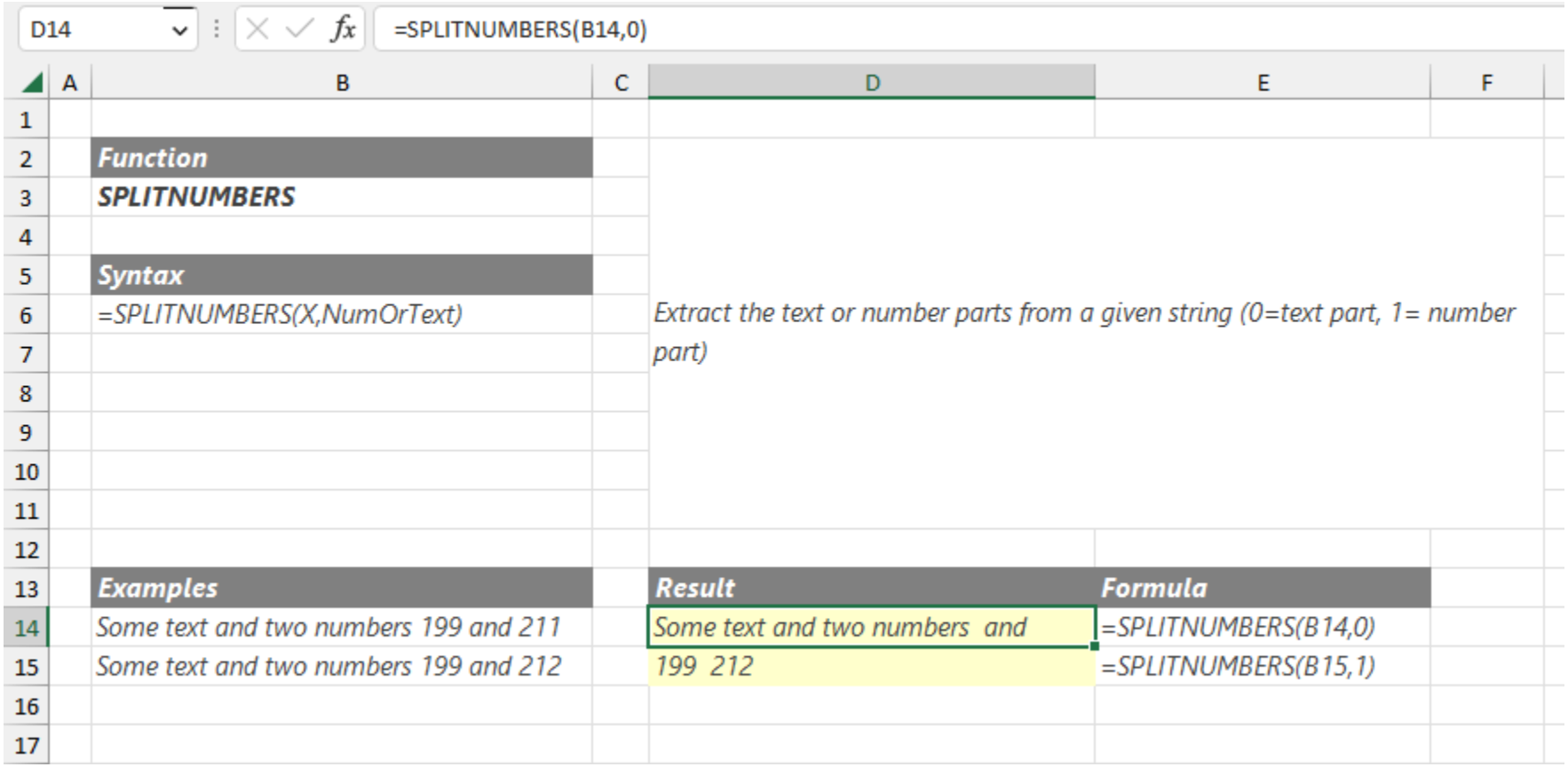
	A	B	C	D	E	F
1						
2		***Function***				
3		***SHORTEN***				
4						
5		***Syntax***				
6		*=SHORTEN(string1,shortenWidth,placeholderText,delimiterCharacter)*		*This function takes a string and shortens it with placeholder text so that it is*		
7				*no longer in length than the specified width.*		
8						
9						
10						
11						
12						
13		***Examples***		***Result***	***Formula***	
14		*Return all characters*		*Return ....*	*=SHORTEN(B14,10,"....")*	
15		*Right split example in Excel*		*Right split ..*	*=SHORTEN(B15,20,"..")*	
16						

SPLIT_TEXT

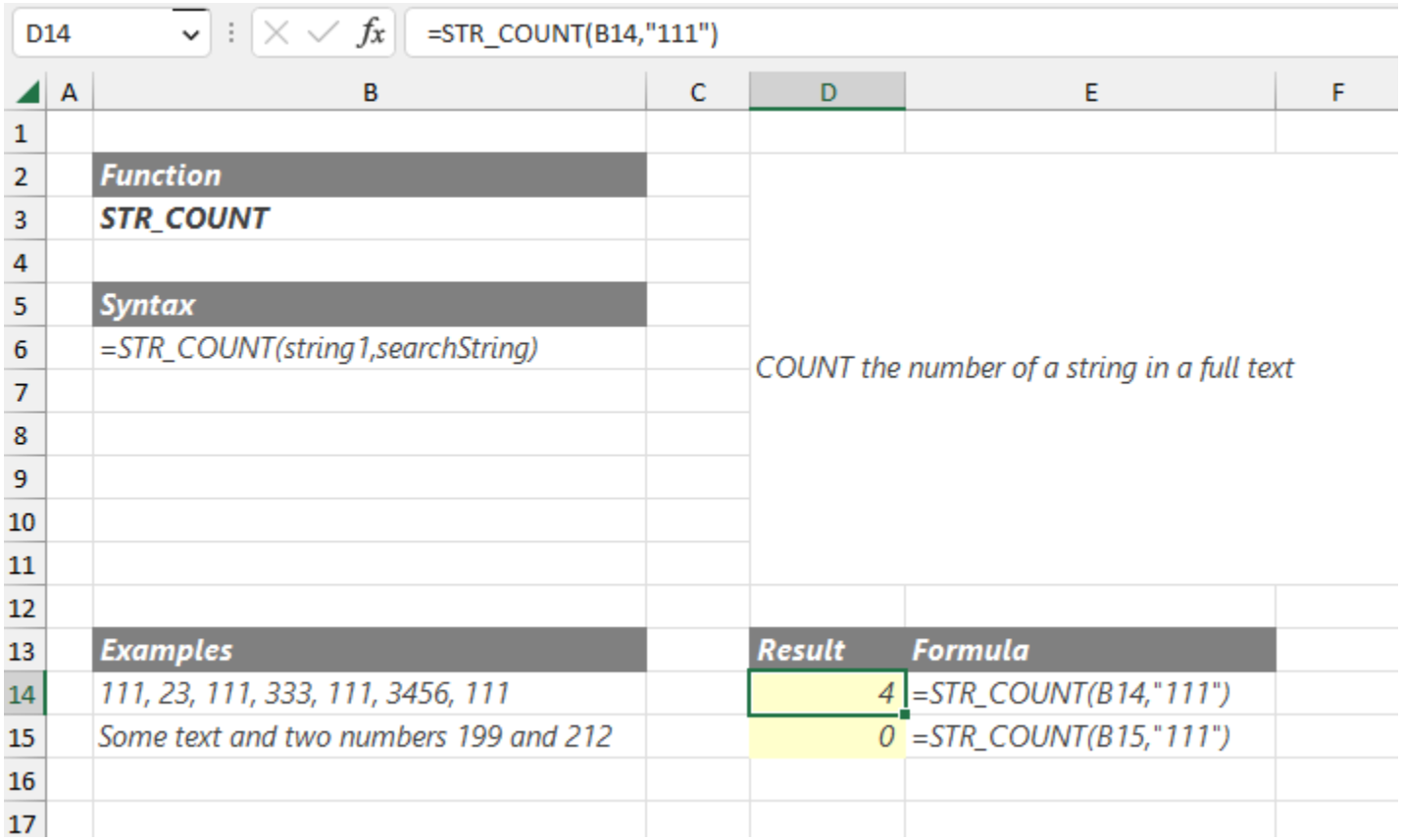


	A	B	C	D	E	F
1						
2		**Function**				
3		**SPLIT_TEXT**				
4						
5		**Syntax**				
6		=SPLIT_TEXT(string1,substringNumber,delimiterString)		SPLIT_TEXT takes a string and a number, splits the string by the space characters, and returns the substring in the position of the number specified in the second argument.		
7						
8						
9						
10						
11						
12						
13		**Examples**		**Result**	**Formula**	
14		One,Two,Three		Two	=SPLIT_TEXT(B14,2,",")	
15		Right split example in Excel		Excel	=SPLIT_TEXT(B15,5," ")	
16						
17						
18						

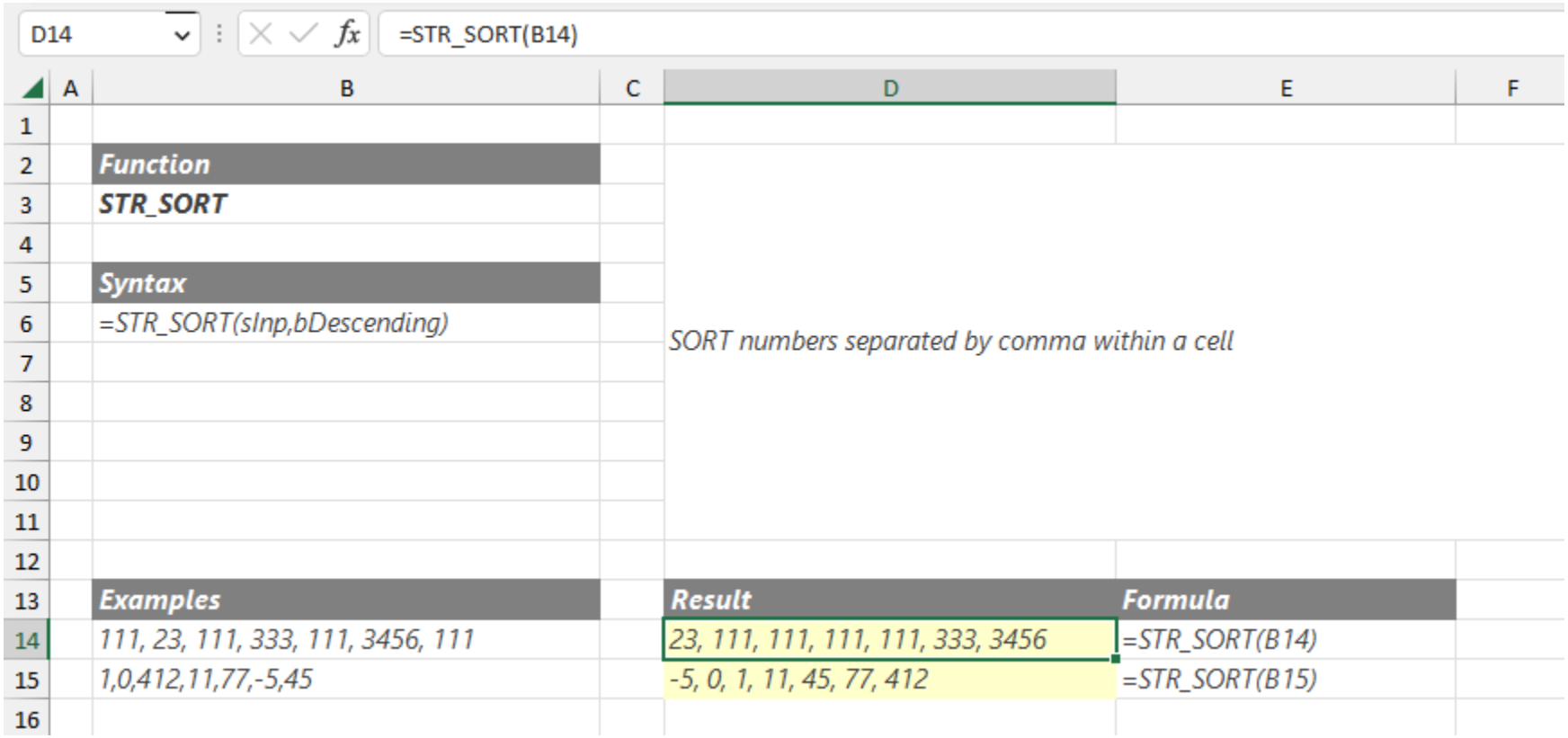
SPLITNUMBERS



STR_COUNT


D14
=STR_COUNT(B14,"111")
Function
STR_COUNT
Syntax
=STR_COUNT(string1,searchString)
COUNT the number of a string in a full text
Examples
Result
Formula
111, 23, 111, 333, 111, 3456, 111
4
=STR_COUNT(B14,"111")
Some text and two numbers 199 and 212
0
=STR_COUNT(B15,"111")

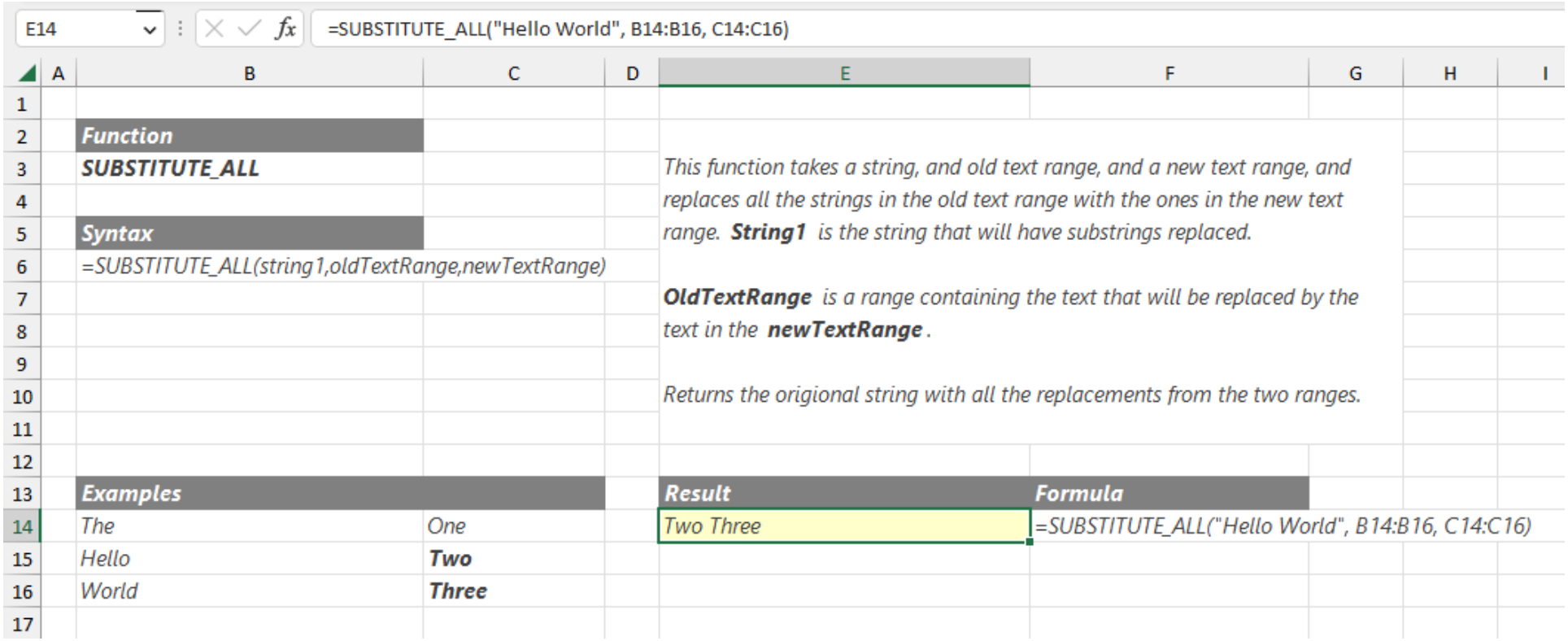
STR_SORT



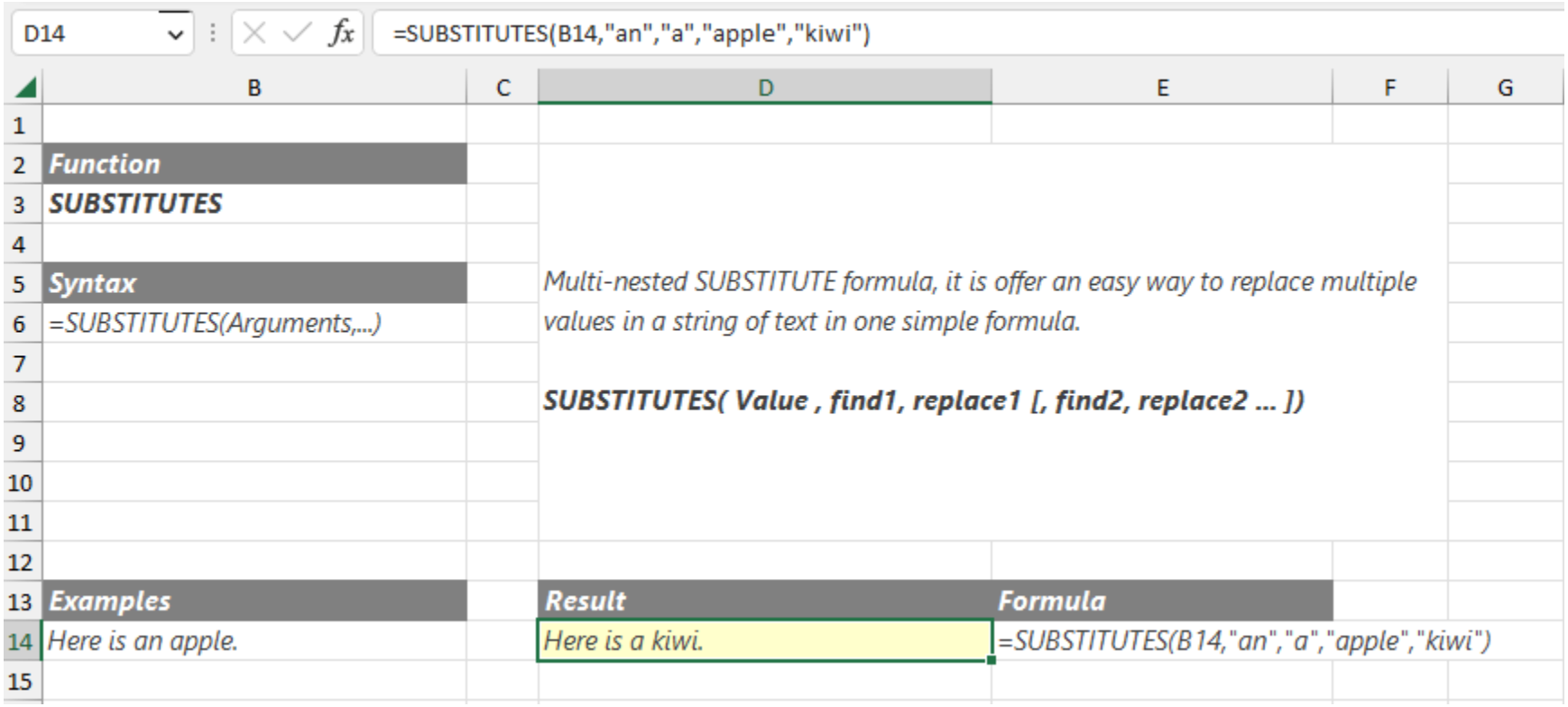
D14 | =STR_SORT(B14)

	A	B	C	D	E	F
1						
2		***Function***				
3		***STR_SORT***				
4						
5		***Syntax***				
6		*=STR_SORT(sInp,bDescending)*		*SORT numbers separated by comma within a cell*		
7						
8						
9						
10						
11						
12						
13		***Examples***		***Result***	***Formula***	
14		*111, 23, 111, 333, 111, 3456, 111*		*23, 111, 111, 111, 111, 333, 3456*	*=STR_SORT(B14)*	
15		*1,0,412,11,77,-5,45*		*-5, 0, 1, 11, 45, 77, 412*	*=STR_SORT(B15)*	
16						

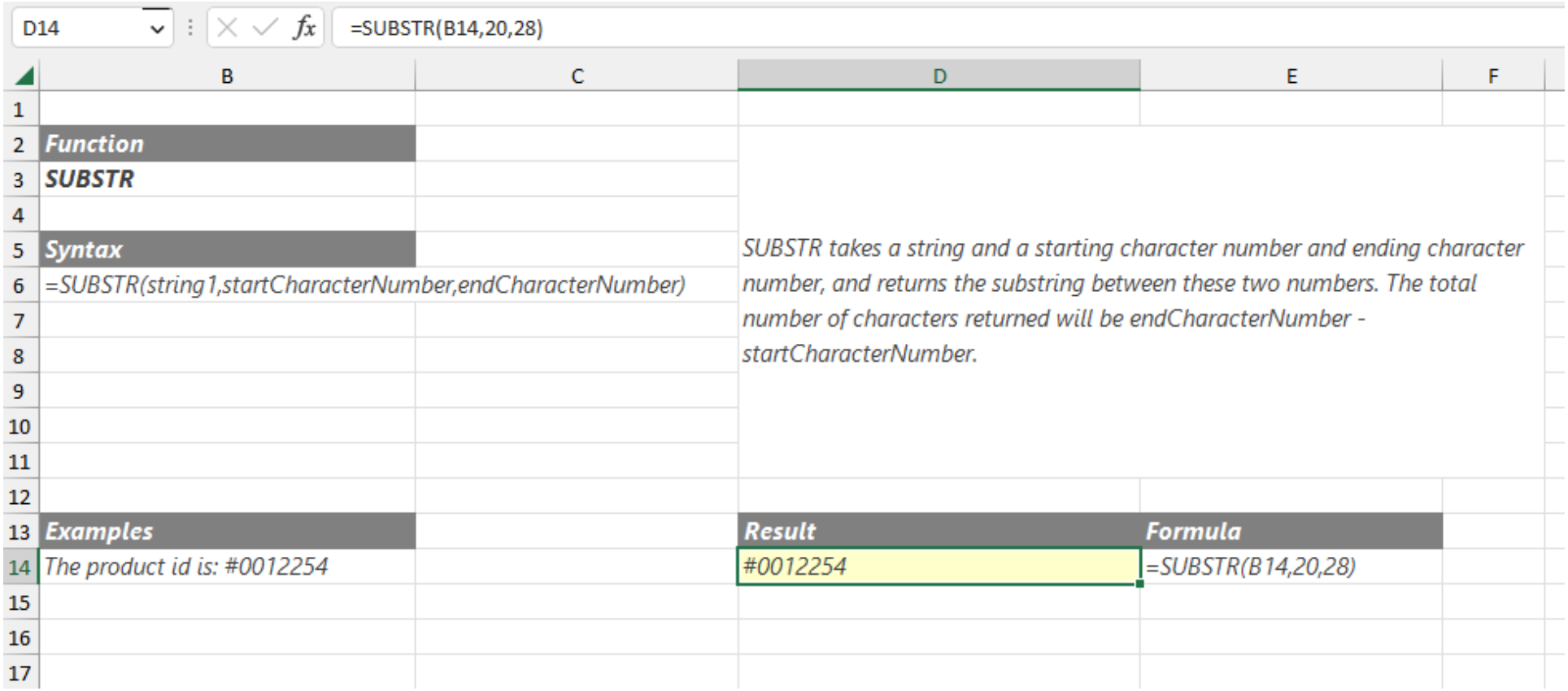
SUBSTITUTE_ALL



SUBSTITUTES

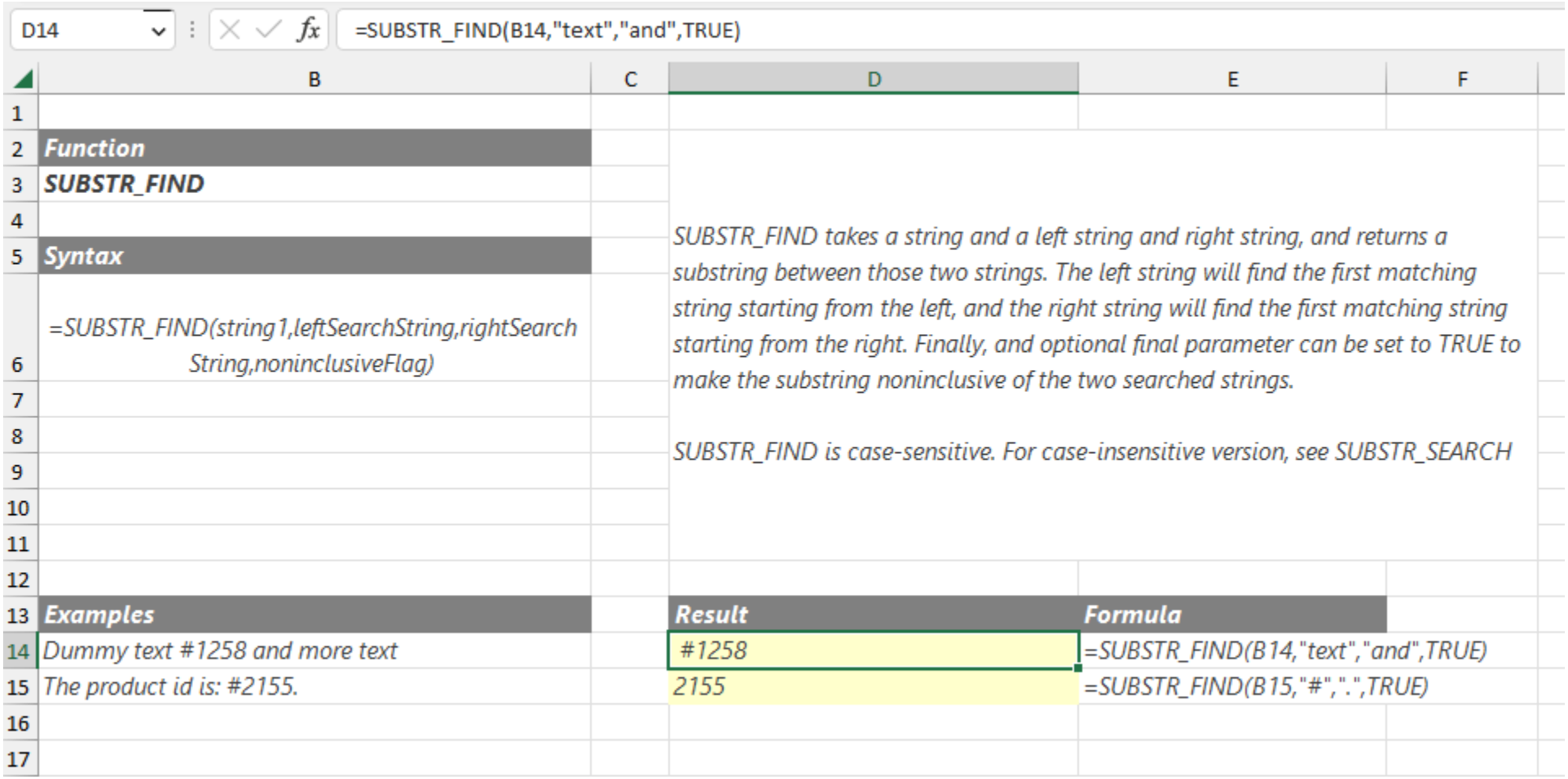


SUBSTR

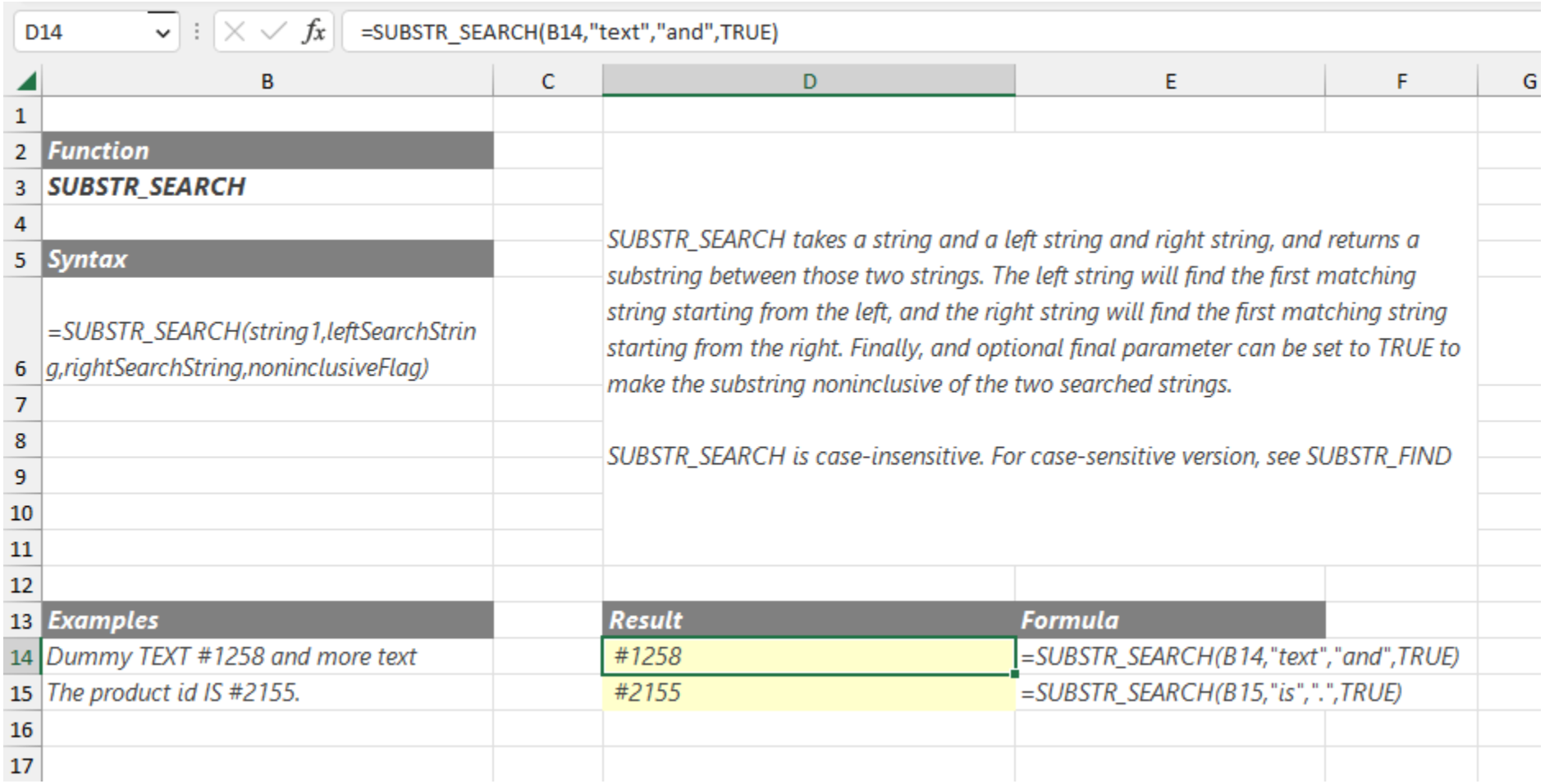


	B	C	D	E	F
1					
2	**Function**				
3	**SUBSTR**				
4					
5	**Syntax**		*SUBSTR takes a string and a starting character number and ending character*		
6	*=SUBSTR(string1,startCharacterNumber,endCharacterNumber)*		*number, and returns the substring between these two numbers. The total*		
7			*number of characters returned will be endCharacterNumber -*		
8			*startCharacterNumber.*		
9					
10					
11					
12					
13	**Examples**		**Result**	**Formula**	
14	*The product id is: #0012254*		*#0012254*	*=SUBSTR(B14,20,28)*	
15					
16					
17					

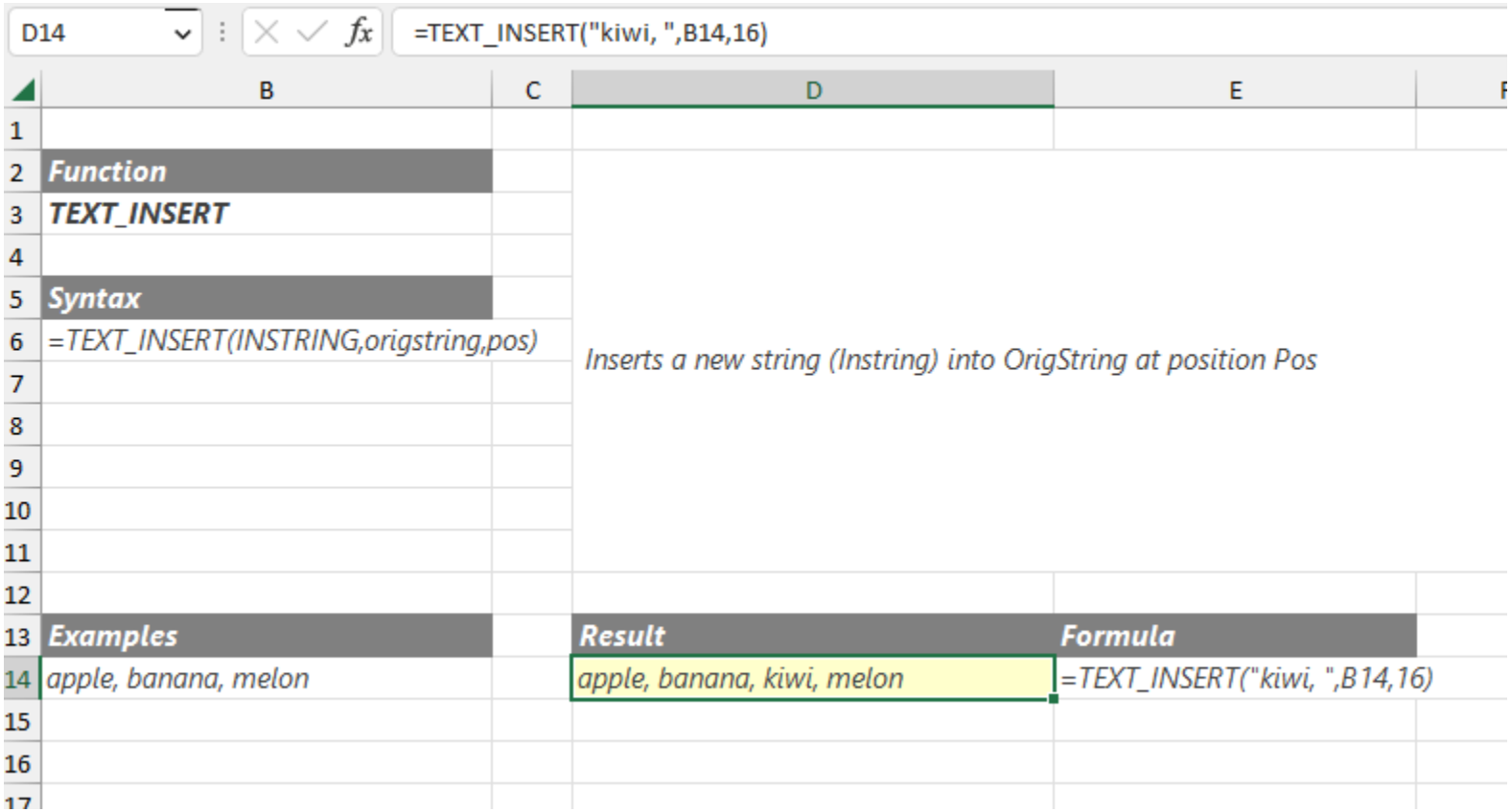
SUBSTR_FIND


D14
=SUBSTR_FIND(B14,"text","and",TRUE)
Function
SUBSTR_FIND
Syntax
=SUBSTR_FIND(string1,leftSearchString,rightSearchString,noninclusiveFlag)
SUBSTR_FIND takes a string and a left string and right string, and returns a substring between those two strings. The left string will find the first matching string starting from the left, and the right string will find the first matching string starting from the right. Finally, and optional final parameter can be set to TRUE to make the substring noninclusive of the two searched strings.
SUBSTR_FIND is case-sensitive. For case-insensitive version, see SUBSTR_SEARCH
Examples
Result
Formula
Dummy text #1258 and more text
#1258
=SUBSTR_FIND(B14,"text","and",TRUE)
The product id is: #2155.
2155
=SUBSTR_FIND(B15,"#",".",TRUE)

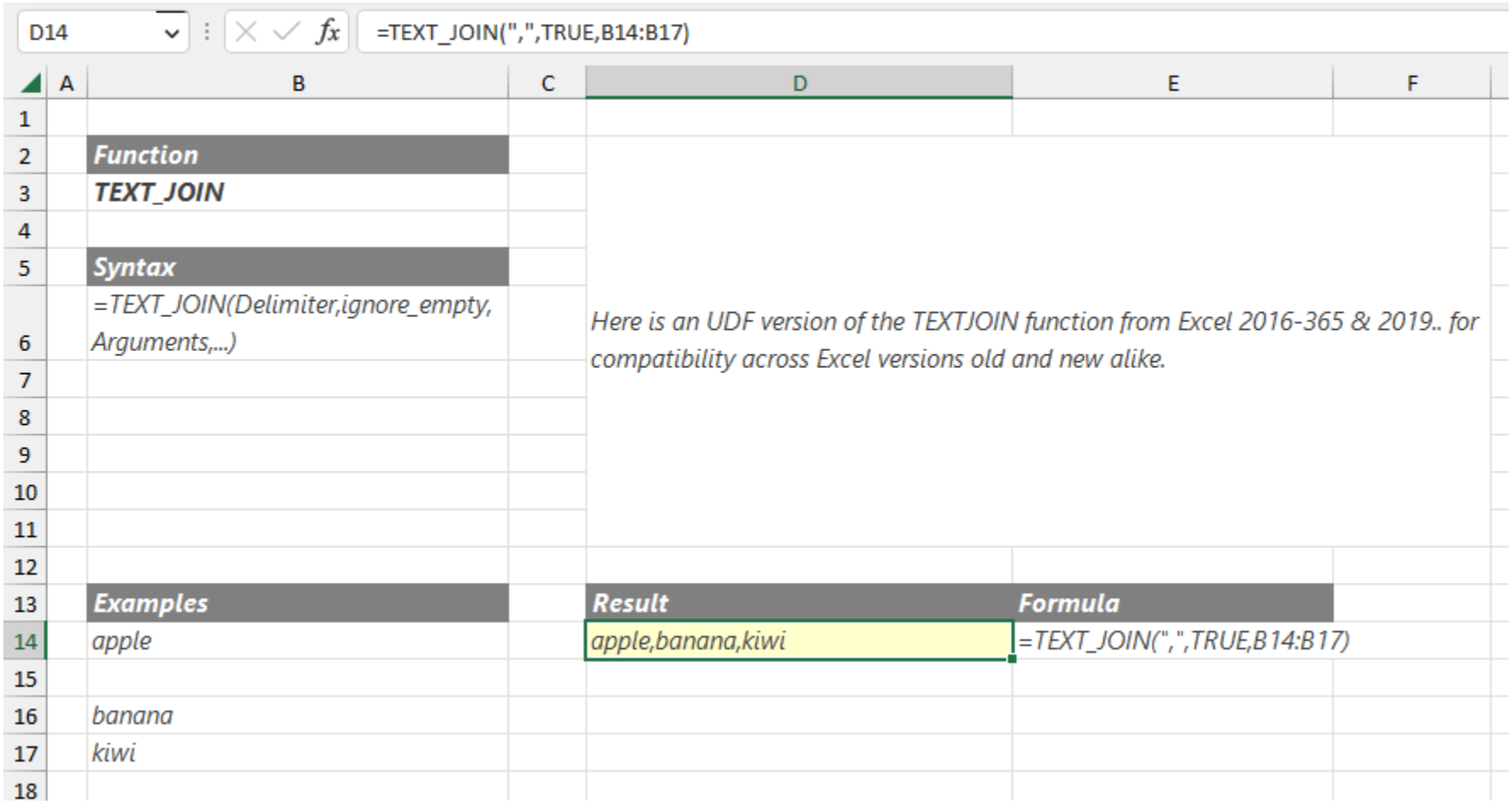
SUBSTR_SEARCH



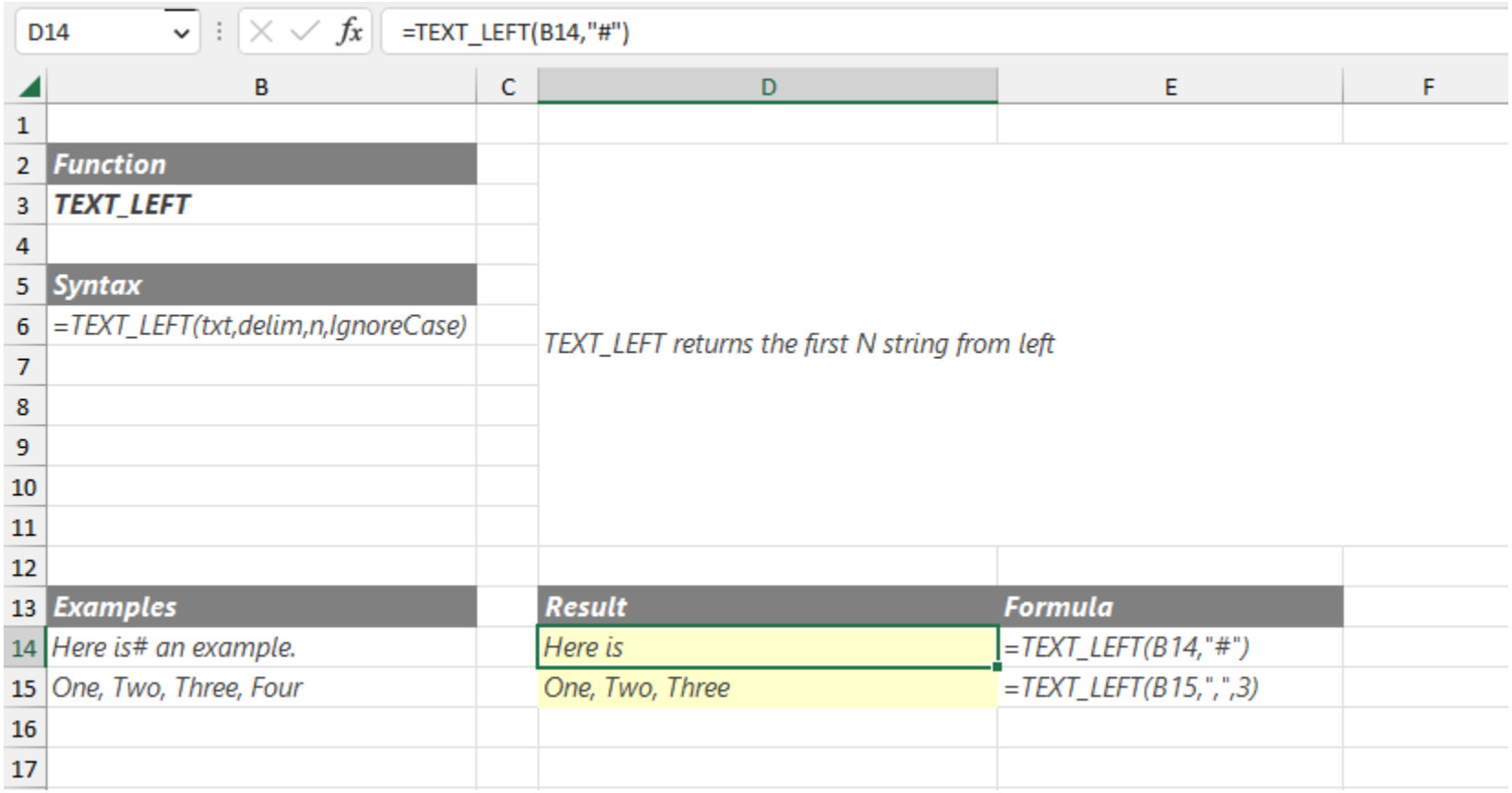
TEXT_INSERT


D14
=TEXT_INSERT("kiwi, ",B14,16)
Function
TEXT_INSERT
Syntax
=TEXT_INSERT(INSTRING,origstring,pos)
Inserts a new string (Instring) into OrigString at position Pos
Examples
Result
Formula
apple, banana, melon
apple, banana, kiwi, melon
=TEXT_INSERT("kiwi, ",B14,16)

TEXT_JOIN

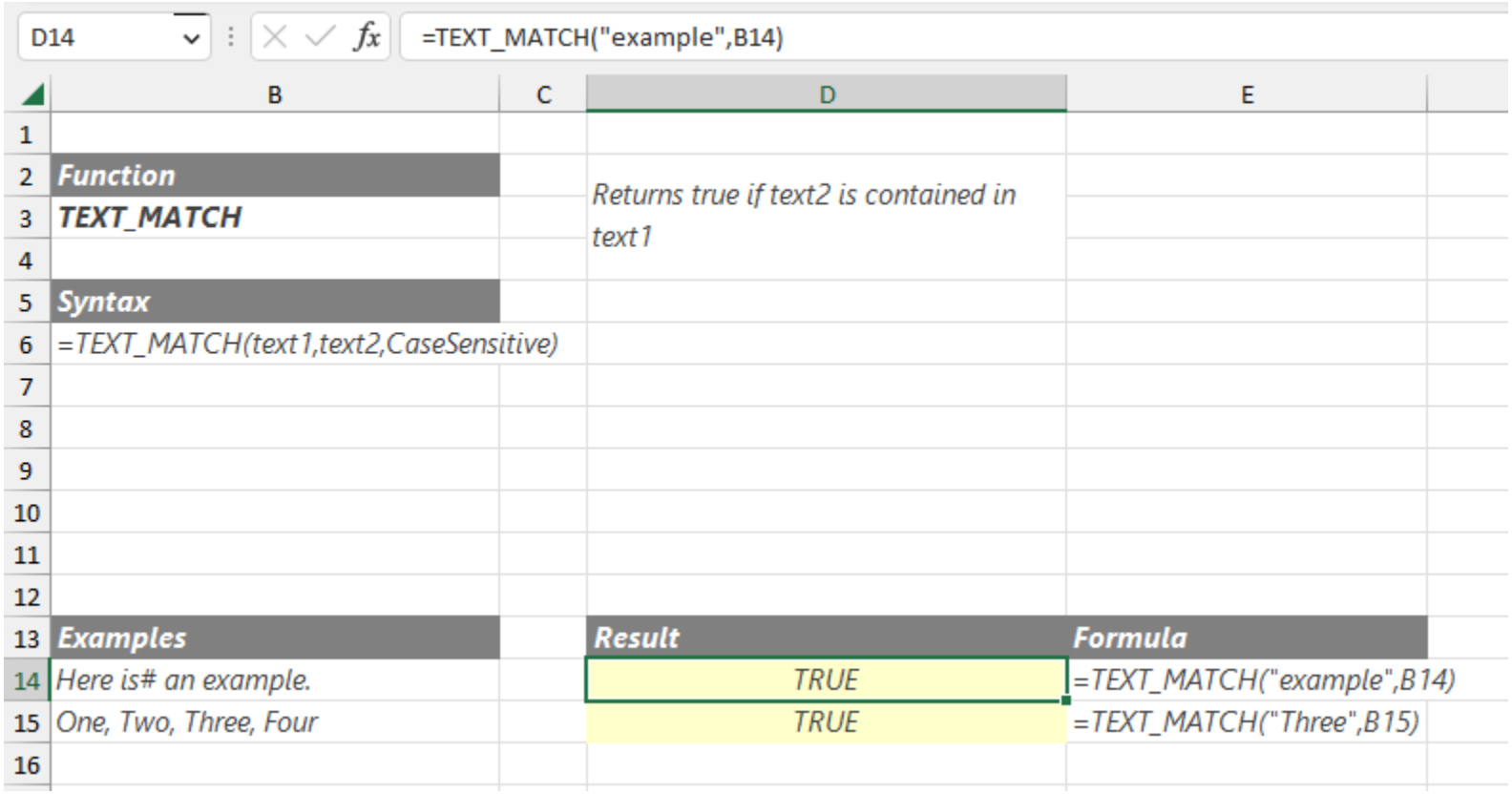

D14
=TEXT_JOIN(",",TRUE,B14:B17)
Function
TEXT_JOIN
Syntax
=TEXT_JOIN(Delimiter,ignore_empty, Arguments,...)
Here is an UDF version of the TEXTJOIN function from Excel 2016-365 & 2019.. for compatibility across Excel versions old and new alike.
Examples
Result
Formula
apple
apple,banana,kiwi
=TEXT_JOIN(",",TRUE,B14:B17)
banana
kiwi

TEXT_LEFT

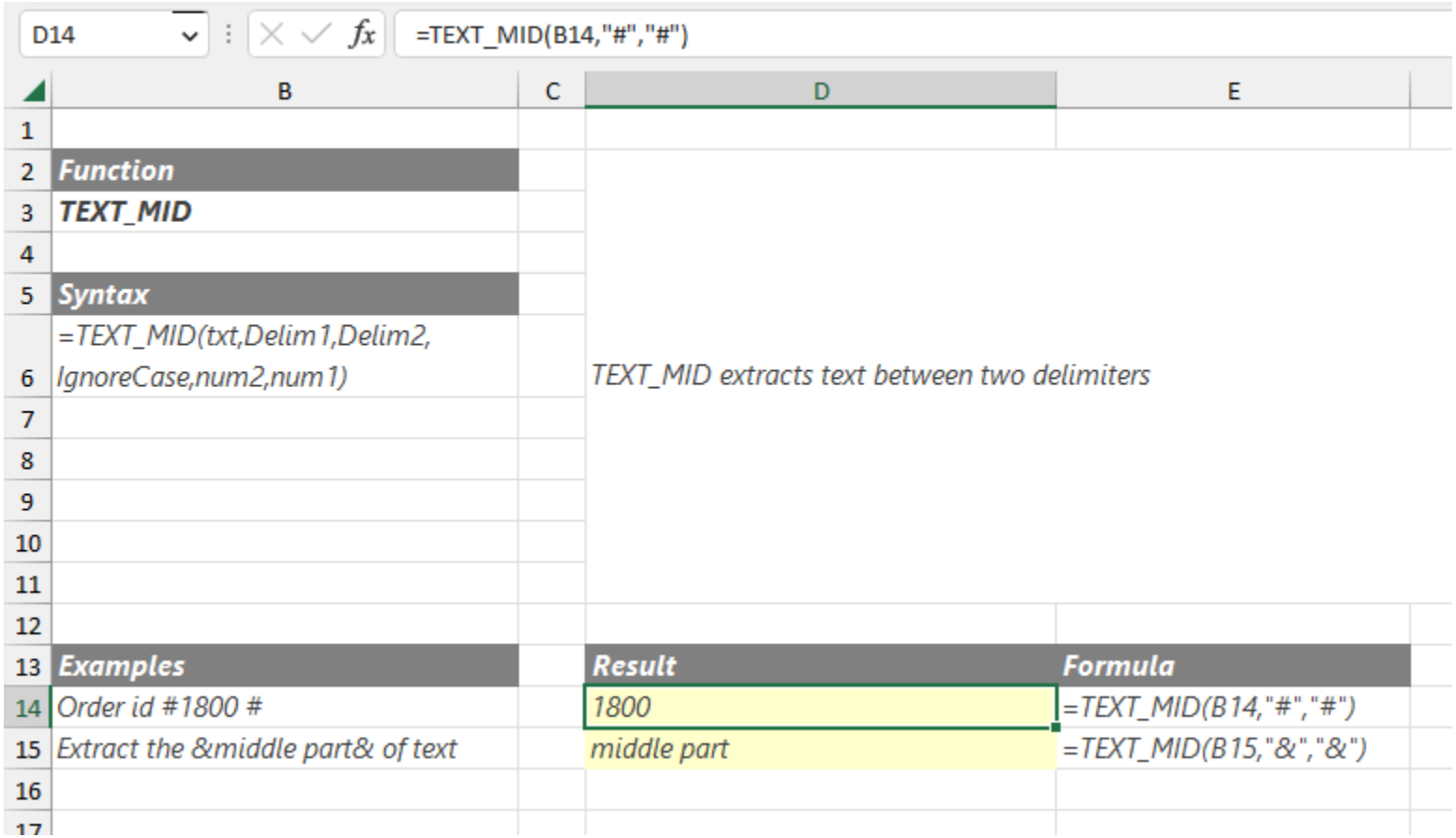


	B	C	D	E	F
1					
2	**Function**				
3	**TEXT_LEFT**				
4					
5	**Syntax**				
6	=TEXT_LEFT(txt,delim,n,IgnoreCase)		TEXT_LEFT returns the first N string from left		
7					
8					
9					
10					
11					
12					
13	**Examples**		**Result**	**Formula**	
14	Here is# an example.		Here is	=TEXT_LEFT(B14,"#")	
15	One, Two, Three, Four		One, Two, Three	=TEXT_LEFT(B15,",",3)	
16					
17					

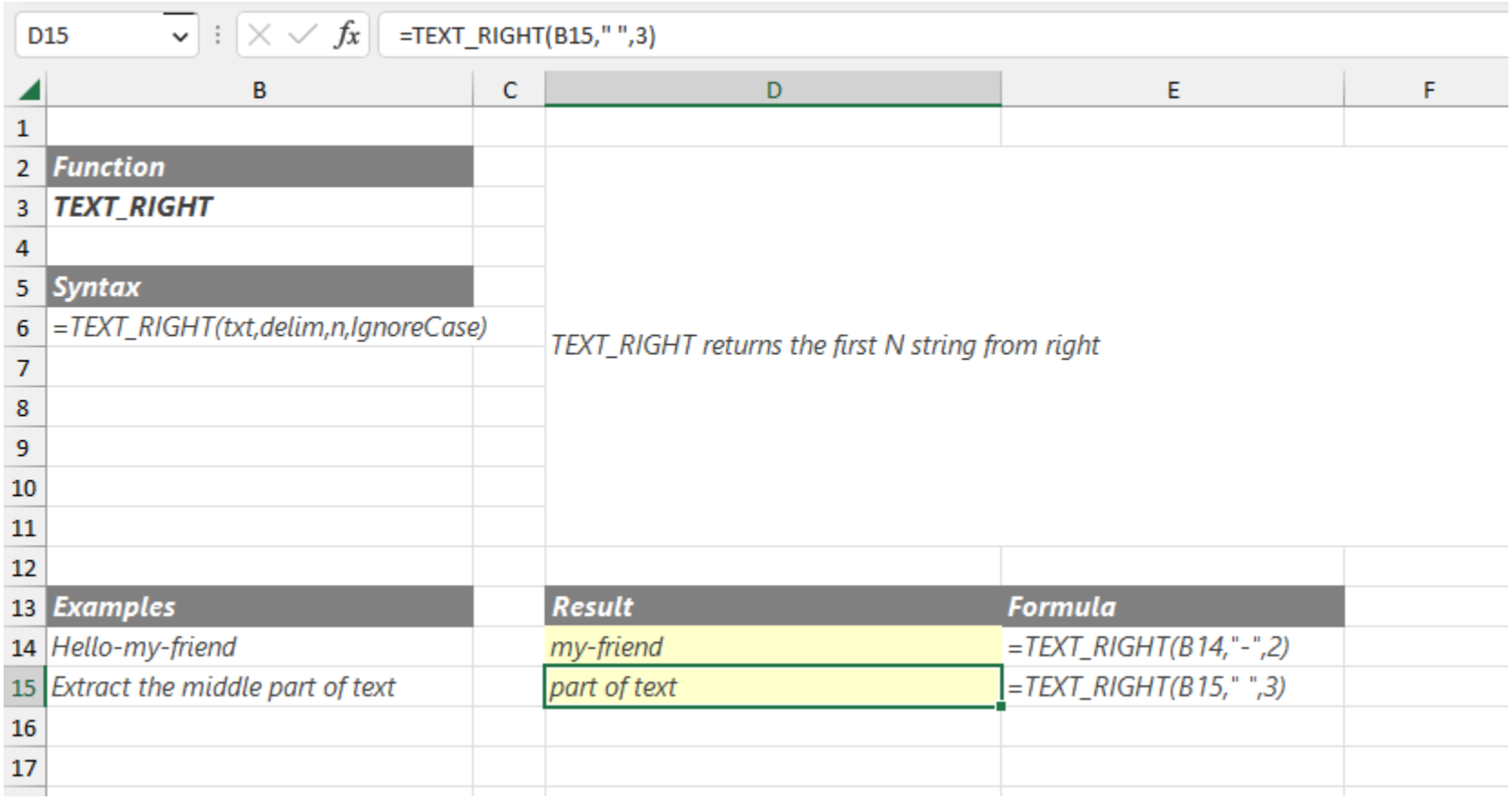
TEXT_MATCH


D14
=TEXT_MATCH("example",B14)
Function
TEXT_MATCH
Returns true if text2 is contained in text1
Syntax
=TEXT_MATCH(text1,text2,CaseSensitive)
Examples
Result
Formula
Here is# an example.
TRUE
=TEXT_MATCH("example",B14)
One, Two, Three, Four
TRUE
=TEXT_MATCH("Three",B15)

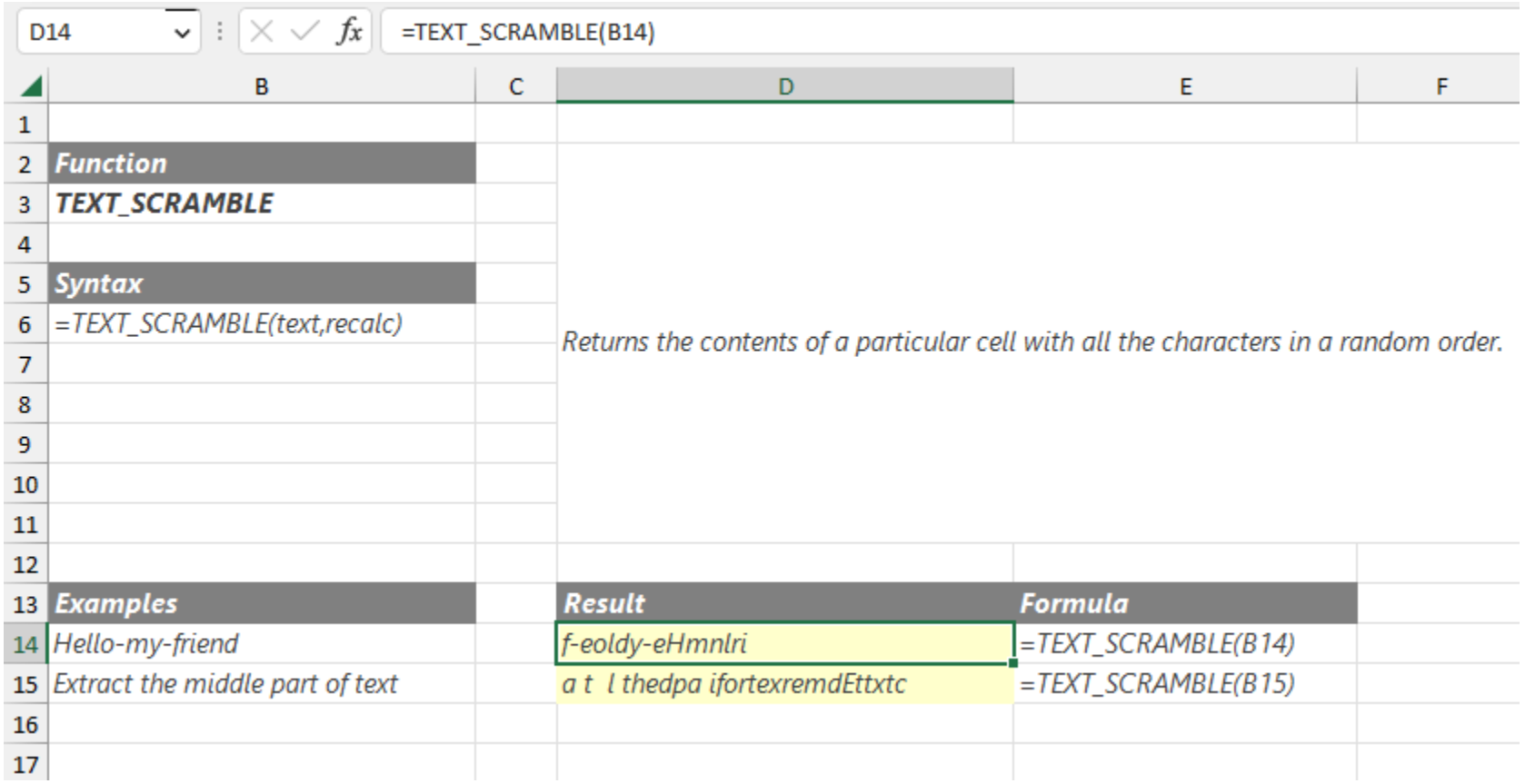
TEXT_MID


D14
=TEXT_MID(B14,"#","#")
B
C
D
E
Function
TEXT_MID
Syntax
=TEXT_MID(txt,Delim1,Delim2,
IgnoreCase,num2,num1)
TEXT_MID extracts text between two delimiters
Examples
Result
Formula
Order id #1800 #
1800
=TEXT_MID(B14,"#","#")
Extract the &middle part& of text
middle part
=TEXT_MID(B15,"&","&")

TEXT_RIGHT


D15
=TEXT_RIGHT(B15," ",3)
Function
TEXT_RIGHT
Syntax
=TEXT_RIGHT(txt,delim,n,IgnoreCase)
TEXT_RIGHT returns the first N string from right
Examples
Result
Formula
Hello-my-friend
my-friend
=TEXT_RIGHT(B14,"-",2)
Extract the middle part of text
part of text
=TEXT_RIGHT(B15," ",3)

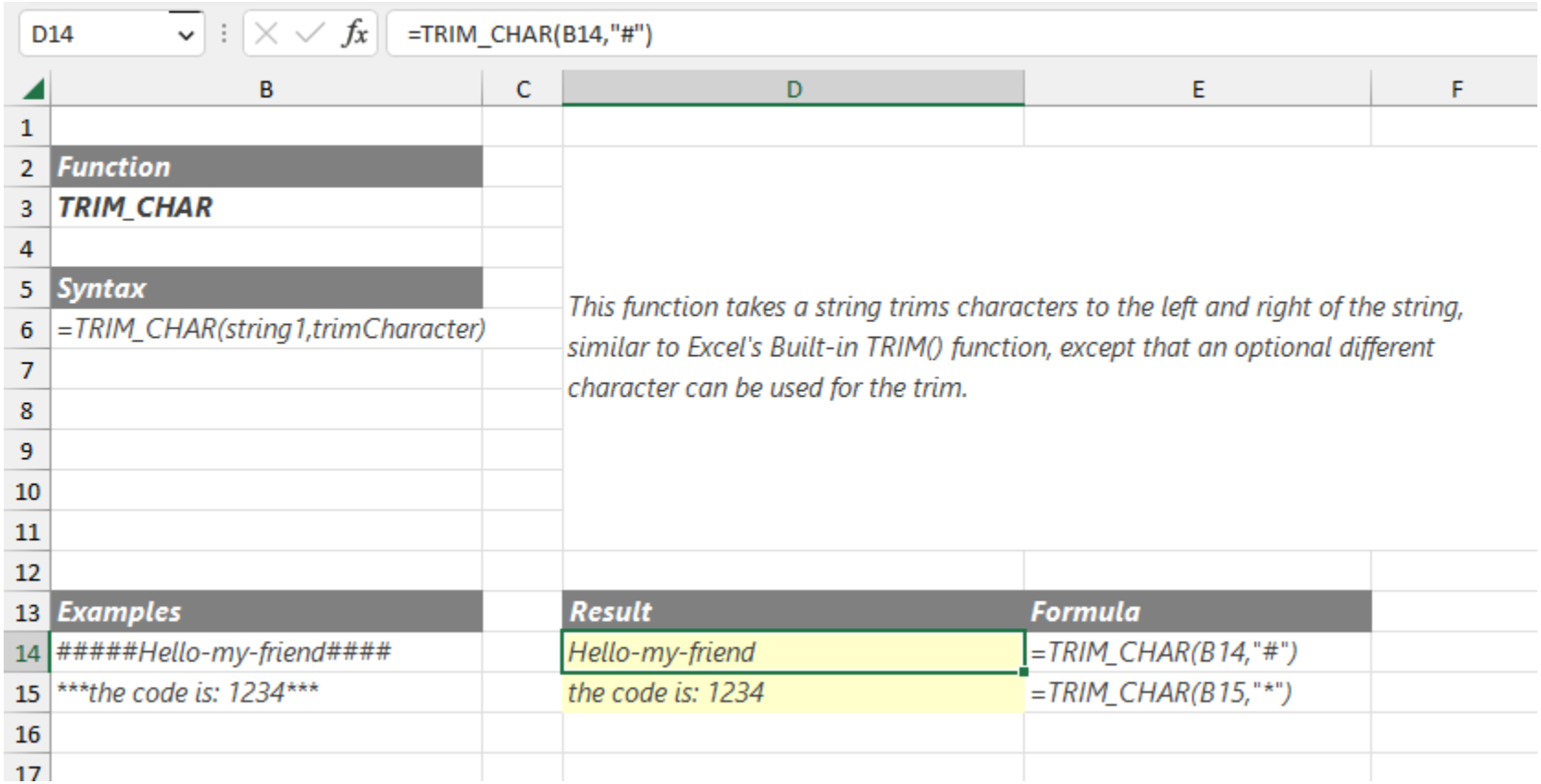
TEXT_SCRAMBLE



D14 | =TEXT_SCRAMBLE(B14)

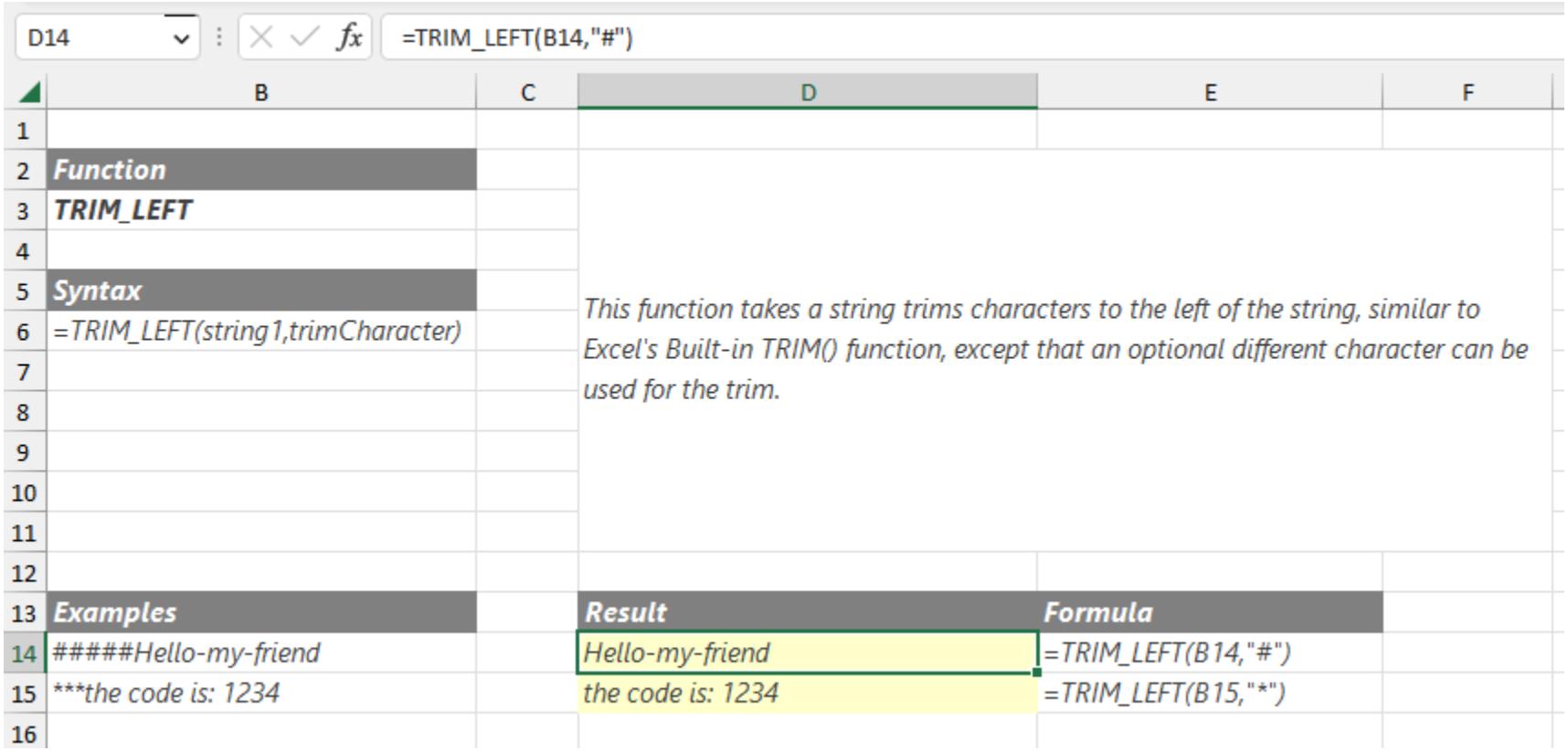
	B	C	D	E	F
1					
2	**Function**				
3	**TEXT_SCRAMBLE**				
4					
5	**Syntax**				
6	=TEXT_SCRAMBLE(text,recalc)		Returns the contents of a particular cell with all the characters in a random order.		
7					
8					
9					
10					
11					
12					
13	**Examples**		**Result**	**Formula**	
14	Hello-my-friend		f-eoldy-eHmnlri	=TEXT_SCRAMBLE(B14)	
15	Extract the middle part of text		a t l thedpa ifortexremdEttxtc	=TEXT_SCRAMBLE(B15)	
16					
17					

TRIM_CHAR



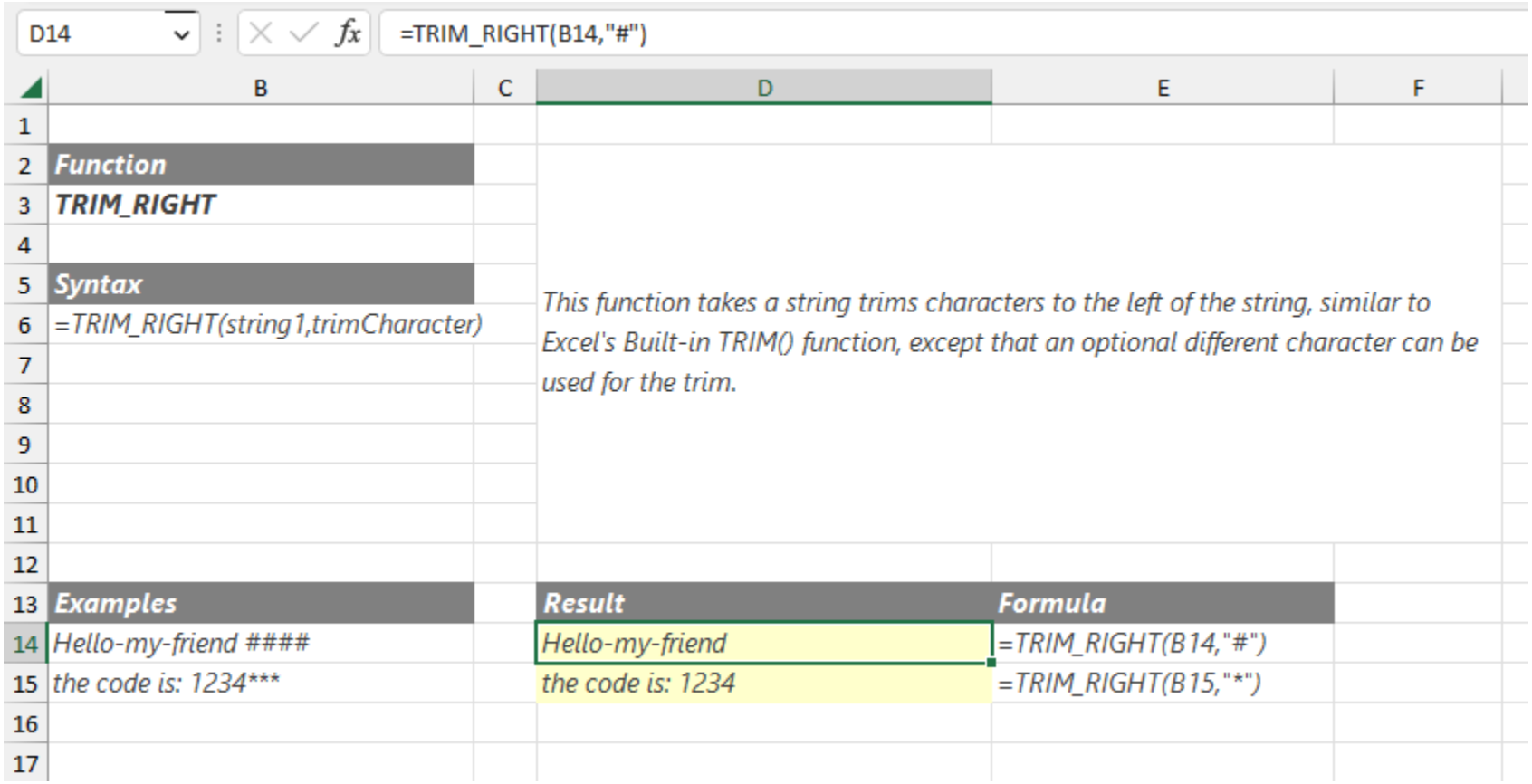
	B	C	D	E	F
1					
2	**Function**				
3	**TRIM_CHAR**				
4					
5	**Syntax**		*This function takes a string trims characters to the left and right of the string, similar to Excel's Built-in TRIM() function, except that an optional different character can be used for the trim.*		
6	=TRIM_CHAR(string1,trimCharacter)				
7					
8					
9					
10					
11					
12					
13	**Examples**		**Result**	**Formula**	
14	#####Hello-my-friend####		Hello-my-friend	=TRIM_CHAR(B14,"#")	
15	***the code is: 1234***		the code is: 1234	=TRIM_CHAR(B15,"*")	
16					
17					

TRIM_LEFT



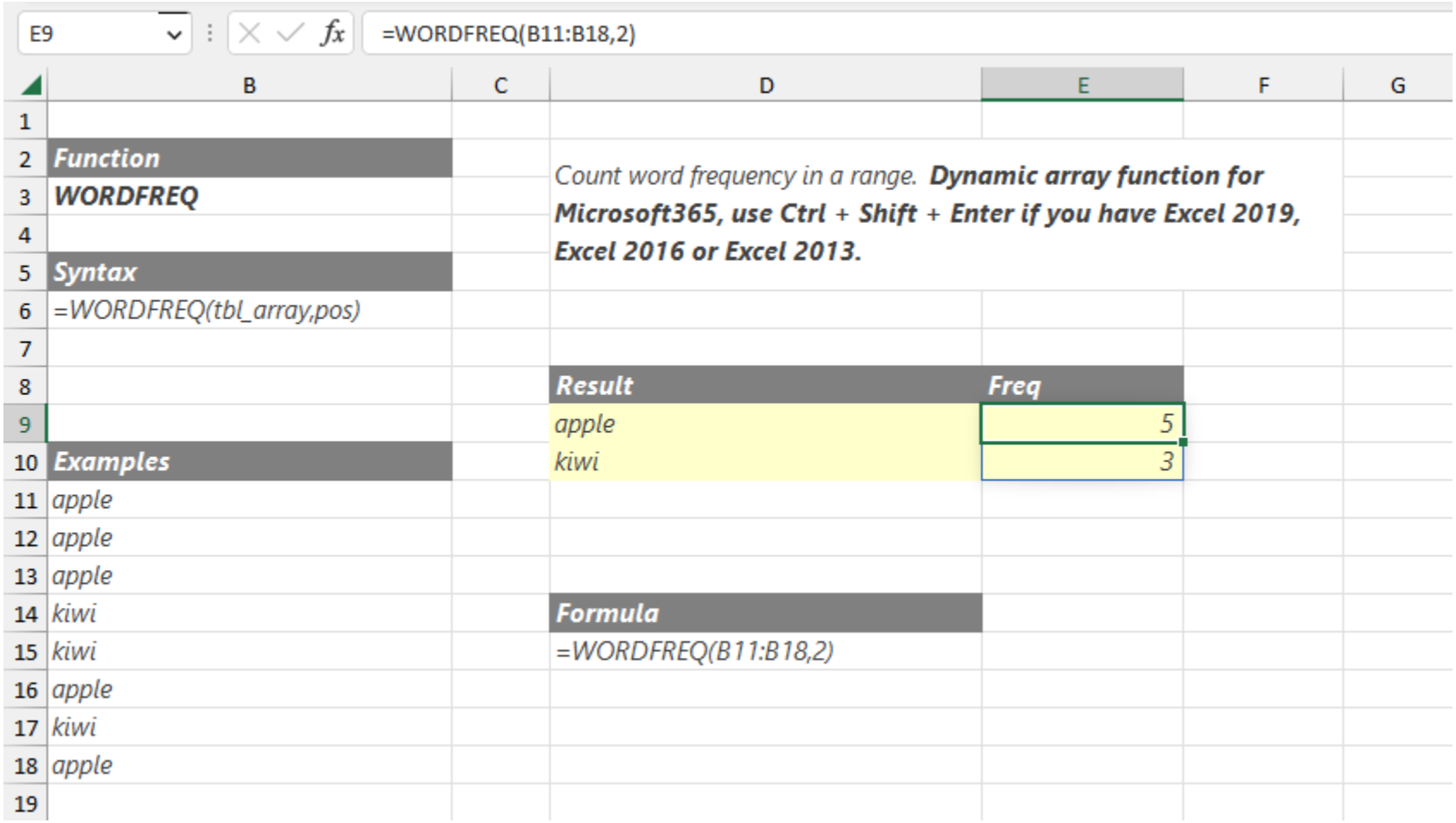
	B	C	D	E	F
1					
2	***Function***				
3	***TRIM_LEFT***				
4					
5	***Syntax***		*This function takes a string trims characters to the left of the string, similar to Excel's Built-in TRIM() function, except that an optional different character can be used for the trim.*		
6	*=TRIM_LEFT(string1,trimCharacter)*				
7					
8					
9					
10					
11					
12					
13	***Examples***		***Result***	***Formula***	
14	*#####Hello-my-friend*		*Hello-my-friend*	*=TRIM_LEFT(B14,"#")*	
15	****the code is: 1234*		*the code is: 1234*	*=TRIM_LEFT(B15,"*")*	
16					

TRIM_RIGHT

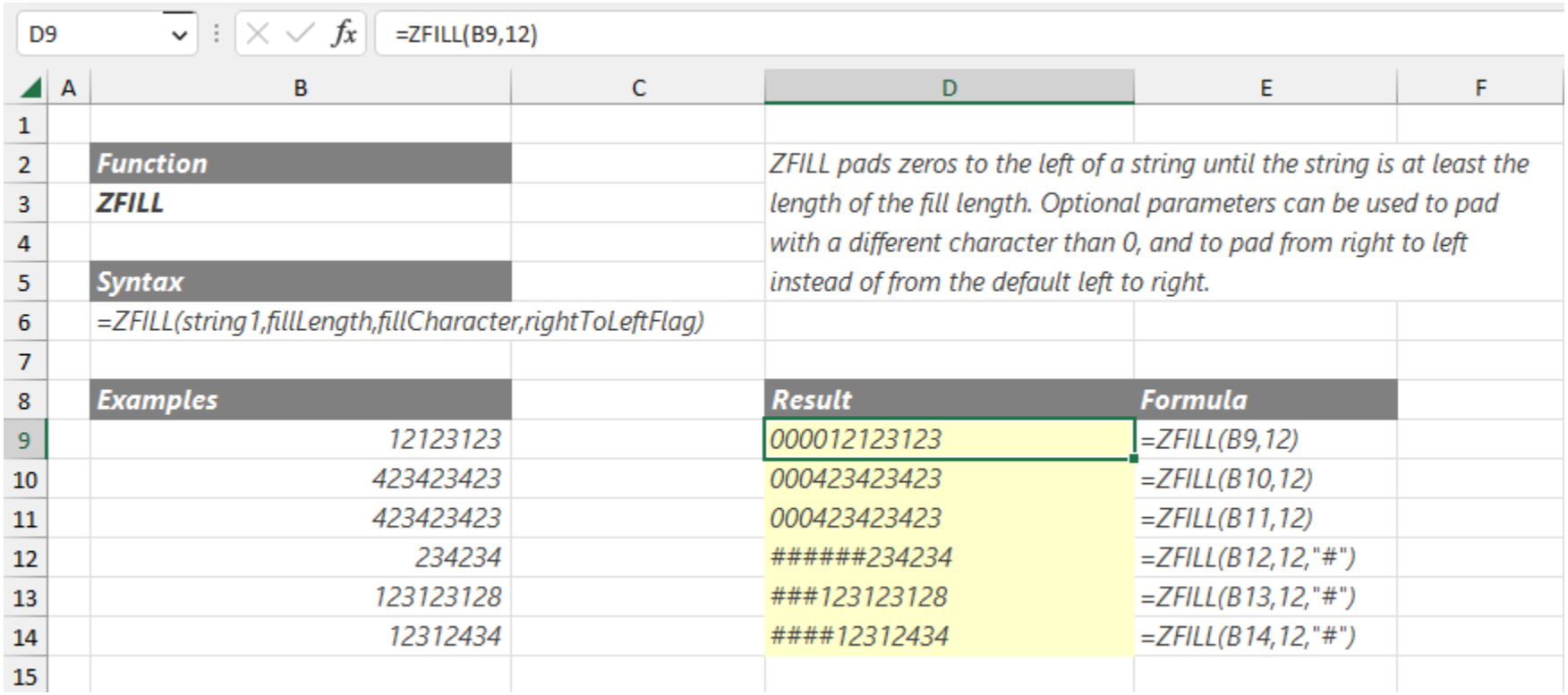


	B	C	D	E	F
1					
2	***Function***				
3	***TRIM_RIGHT***				
4					
5	***Syntax***		*This function takes a string trims characters to the left of the string, similar to Excel's Built-in TRIM() function, except that an optional different character can be used for the trim.*		
6	*=TRIM_RIGHT(string1,trimCharacter)*				
7					
8					
9					
10					
11					
12					
13	***Examples***		***Result***	***Formula***	
14	*Hello-my-friend ####*		*Hello-my-friend*	*=TRIM_RIGHT(B14,"#")*	
15	*the code is: 1234****		*the code is: 1234*	*=TRIM_RIGHT(B15,"*")*	
16					
17					

WORDFREQ


E9
=WORDFREQ(B11:B18,2)
Function
WORDFREQ
Syntax
=WORDFREQ(tbl_array,pos)
Examples
apple
apple
apple
kiwi
kiwi
apple
kiwi
apple
Count word frequency in a range. Dynamic array function for Microsoft365, use Ctrl + Shift + Enter if you have Excel 2019, Excel 2016 or Excel 2013.
Result
Freq
apple
5
kiwi
3
Formula
=WORDFREQ(B11:B18,2)

ZFILL



D9 =ZFILL(B9,12)

	A	B	C	D	E	F
1						
2		**Function**		*ZFILL pads zeros to the left of a string until the string is at least the*		
3		**ZFILL**		*length of the fill length. Optional parameters can be used to pad*		
4				*with a different character than 0, and to pad from right to left*		
5		**Syntax**		*instead of from the default left to right.*		
6		*=ZFILL(string1,fillLength,fillCharacter,rightToLeftFlag)*				
7						
8		**Examples**		**Result**	**Formula**	
9		*12123123*		*000012123123*	*=ZFILL(B9,12)*	
10		*423423423*		*000423423423*	*=ZFILL(B10,12)*	
11		*423423423*		*000423423423*	*=ZFILL(B11,12)*	
12		*234234*		*######234234*	*=ZFILL(B12,12,"#")*	
13		*123123128*		*###123123128*	*=ZFILL(B13,12,"#")*	
14		*12312434*		*####12312434*	*=ZFILL(B14,12,"#")*	
15						