

Basics of: Power Query custom functions for Excel formula developers

Consider this function to initialize first names

```
1  ✓ (
2    table as table,
3    column_name as text,
4    new_column_name as text
5  ) as table =>
6
7  ✓ Table.AddColumn(
8    table,
9    new_column_name,
10  ✓  each
11  ✓    let
12      Names = Text.Split( Record.Field(_,column_name) , " " ),
13
14      Initials = List.Transform( Names , each Text.Start(_,1) & "." ),
15
16      FirstMiddle = Text.Combine( List.RemoveLastN( Initials , 1 ) )
17  ✓    in
18      FirstMiddle & " " & List.Last( Names )
19  )
```

The function signature goes at the top of the query

We comma-separate and wrap the parameters in parentheses at the beginning of the query. Parameter types (e.g. "as table") are optional.

```
1  ✓ (  
2    table as table,  
3    column_name as text,  
4    new_column_name as text  
5  ) as table =>  
6
```

The "=>" part separates the function signature from the function body.

"as table" here means the function returns a table. This is also optional.

The body of the query is an *expression* that returns a *value* of the type specified in the function signature

 The function body does **not** have to have a **let..in** statement

```
5    ) as table =>
```

```
6
```

```
7  ✓ Table.AddColumn(  
    |
```

This function's signature tells us it returns a table. If this is omitted, the default is "as any".

The expression in this function is a call to the "Table.AddColumn" function, which adds a column to a table and returns the table with the new column added.

This is the *expression* that returns the *value* for this function.

Custom functions can also appear inside other functions

```
7  Table.AddColumn(  
8      table,  
9      new_column_name,  
10     each  
11     let  
12         Names = Text.Split( Record.Field(_,column_name) , " " ),  
13  
14         Initials = List.Transform( Names , each Text.Start(_,1) & "." ),  
15     )
```

Wherever we see “each”, we are looking at a custom function. In “Table.AddColumn”, “each” is shorthand for “(_ as record) as any”

The underscore here refers to the “each” in “Table.AddColumn”, so this underscore means “the current record in the table”.

The underscore here refers to the “each” in “List.Transform”, so this underscore means “the current list item”.

“let.in” is just a way of defining names that can be re-used (equivalent to =LET)

Split the value in the column “column_name” in the current record using a space as the delimiter. Return a list of words.

Equivalent to =TEXTSPLIT(A1," ")

```
11  ✓ | | | let
12    | | |   Names = Text.Split( Record.Field(_,column_name) , " " ),
13    | | |
14    | | |   Initials = List.Transform( Names , each Text.Start(_,1) & "." ),
15    | | |
```

Apply a function to each name one by one. Concatenate the initial of each name with a period.

Equivalent to

=MAP(A1#,LAMBDA(x,LEFT(x,1)&"."))

Get the first character from the current list item.

Equivalent to =LEFT(A1,1)

Save the whole thing as a new query with the name “InitializeFirstNames”

The screenshot shows the FlexyYourData interface. On the left, a sidebar titled 'Queries [2]' contains two items: 'fx InitializeFirstNames' (highlighted in green) and 'UseInitializeFirstNames'. The main area is titled 'Enter Parameters' and contains three input fields: 'table' (a dropdown menu), 'column_name' (a text field with 'Example: abc' placeholder), and 'new_column_name' (a text field with 'Example: abc' placeholder). Below these fields are two buttons: 'Invoke' and 'Clear'. At the top of the main area, there is a toolbar with icons for 'X', 'checkmark', 'fx', and '=', followed by an open parenthesis '('. At the bottom of the main area, the function signature is displayed: 'function (table as table, column_name as text, new_column_name as text) as table'.

We can also add documentation to the function to show more descriptive text on this screen

Now it can be used to quickly apply those steps with one function call

	ABC 123 Full Name	ABC 123 Initialized
1	Srinivasa Iyengar Ramanujan	S.I. Ramanujan
2	Harish Chandra Verma	H.C. Verma
3	Ravi Dutt Sharma	R.D. Sharma
4	Richard Phillips Feynman	R.P. Feynman
5	Stephen William Hawking	S.W. Hawking
6	Walter Lewin	W. Lewin

Advanced Editor

UseInitializeFirstNames

Display Options ?

```
1 let
2   Source = Excel.CurrentWorkbook(){[Name="Table1"]}[Content],
3   Inits = InitializeFirstNames(Source,"Full Name","Initialized")
4 in
5   Inits
```

✓ No syntax errors have been detected.

Done Cancel

