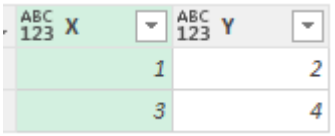


# Power Query (M language) cheat sheet

**Note: M is a case sensitive language!**

Kind	Literal	Comment
null	null	Empty value, void 1 * null = null // be careful!
logical	true / false	
number	0 1 -1 1.5 2.3e-5, 0xff	Whole / decimal number, number in hex
time	#time(9, 15, 0)	#time( hour, minute, second ) #time(24,0,0) = #time(0,0,0) If hour is 24, then minute and second must be 0 0 ≤ hour ≤ 24, 0 ≤ minute ≤ 59, 0 ≤ second ≤ 59
date	#date(2013, 2, 26)	#date( year, month, day)
datetime	#datetime(2013, 2, 26, 9, 15, 0)	#datetime( year, month, day, hour, minute, second )
datetimezone	#datetimezone(2013, 2, 26, 9, 15, 0, 9, 0)	#datetimezone( year, month, day, hour, minute, second, offset-hours, offset-minutes ) 0 ≤ year ≤ 9999, 0 ≤ month ≤ 12, 1 ≤ day ≤ 31 0 ≤ hour ≤ 23, 0 ≤ minute ≤ 59, 0 ≤ second ≤ 59 -14 ≤ offset-hours + offset-minutes / 60 ≤ 14
duration	#duration( 0, 1, 30, 0)	#duration( days, hours, minutes, seconds )
text	"hello"	Just text in quotes Special characters. Carriage return: ="#(cr,lf)" same as ="#(cr)#(lf)", string to check ="a#(cr,lf)b" = "a#(tab)b" // a        b = "a" & "b""c" // ab"c
binary	#binary("AQID")	If you work with binary – you know
list	{ 1, 2, 3 }, { 1 .. 10 }, {"A".. "Z", "a".. "z"}	Comma separated values in curly brackets
record	[ A=1, B=2 ]	Comma separated "Field Name = Value" in square brackets
table	Simple way: #table( { "X", "Y" }, { { 1, 2 }, { 3, 4 } } )  Preferable: with specified column types #table( type table [Digit ID = number, Name = text], { {1,"one"}, {2,"two"}, {3,"three"} } )	result:   #table( list of field names, list of lists with values for <u>rows</u> of future table )  #table( { "Field1 Name", "Field2 Name" }, { { "Field1 Value1", "Field2 Value1" }, { "Field1 Value2", "Field2 Value2" }, { "Field1 Value3", "Field2 Value3" } } )  Empty table: #table( { "A", "B" }, { } )
function	(x) => x + 1	( arguments ) => some operations. (optional num as nullable number) => let step1 = if num = null then 0 else num, step2 = step1 * 2 in step2
type	type{ number } // list type table [A = any, B = text]	Type of "data type"

Operator		x = y	Equal
x > y	Greater than	x <> y	Not equal
x >= y	Greater than or equal	x or y	Conditional logical OR
x < y	Less than	x and y	Conditional logical AND
x <= y	Less than or equal	not x	Logical NOT

<b>Expressions</b> "Hello World" // a text value 123 // a number 1 + 2 // sum of two numbers  {1, 2, 3} // a list of three numbers  [ x = 1, y = 2 + 3 ] // a record containing two fields: x and y  (x, y) => x + y // a function that computes a sum  if 2 > 1 then 2 else 1 // a conditional expression  <b>let</b> x = 1 + 1 <b>in</b> x * 2 // a let expression  error "A" // error with message "A"	<b>Recursion (<a href="#">blog post</a>)</b> Factorial = (n) => if n <= 1 then 1 else n * @Factorial(n - 1),  x = Factorial(5)  // @ is scoping operator
--	---

## Relative dates

```

Today= Date.From(DateTime.FixedLocalNow()),
Yesterday= Date.AddDays(Date.From(DateTime.FixedLocalNow()), -1),
#"End of last month" = Date.EndOfMonth(Date.AddMonths(DateTime.FixedLocalNow(), -1)),
#"Start of Current Year"= Date.StartOfYear( Date.From(DateTime.FixedLocalNow()) ),
#"Start of Previous Year"= Date.AddYears(Date.StartOfYear(DateTime.FixedLocalNow()), -1),

#"ISO Date format"=Date.ToText( Date.From(DateTime.FixedLocalNow()), "yyyy-MM-ddT00:00:00"),

#"Start of Month 12 months ago Excluding cur month"=
    Date.StartOfMonth(Date.AddMonths(DateTime.FixedLocalNow(), -12)),
#"Start of Month 12 months ago Including cur month"=
    Date.StartOfMonth(Date.AddMonths(DateTime.FixedLocalNow(), -11)),

// Generate Calendar – (blog post), solution for Power BI
// List of dates for PrevYear - Today
let
    start = Date.AddYears(Date.StartOfYear(DateTime.FixedLocalNow()), -1), // start of prev year
    end = Date.From(DateTime.FixedLocalNow()), // today
    duration = Duration.Days(end - start) + 1,
    list_of_dates = List.Dates(start, duration, #duration(1,0,0,0)),

    #"Table from List" = Table.FromList(list_of_dates, Splitter.SplitByNothing(), null, null, ExtraValues.Error)
in
    #"Table from List"

Get working days – try solution from Marco Russo
Option 1: Parse table from TimeAndDate.com
Option 2: Use API TimeAndDate.com
Russia: читать в блоге, function on GitHub

```

# Power Query code shortcuts

## IF / THEN / ELSE

Result = if [Column1]>0 then [Column A] else [Column B] // low case if / then / else, M is case sensitive

## TRY / CATCH – error handling

Result = try A/B otherwise 0 // low case “try [some action] otherwise [some action/object]”

## Excel cell value (Named Range consisting of one cell)

Result = Excel.CurrentWorkbook(){[Name="CELLNAME"]}[Content]{0}[Column1]

## Rename Columns according to “Renaming Table”

Renamed\_Columns = Table.RenameColumns(TARGET,  
Table.ToColumns(Table.Transpose(RENAMING\_TABLE)), MissingField.Ignore),

where RENAMING\_TABLE looks like

Old Name	New Name
A	B
C	D

## Rename using List.Zip, when you know order of your columns ([blog post](#))

Renamed\_Columns = Table.RenameColumns(TARGET,  
List.Zip( { Table.ColumnNames( Source ), { "Sales Org", "Territory Key" } } ), MissingField.Ignore),

## Create a table from thin air

For example, when response is null but you want to keep structure of your PowerPivot table

= #table( {"A", "B"}, {} ) – empty table, simple approach

Or with defined column types

= #table( type table [A = text, B = number], {} ) – empty table

= #table( type table [My Column A = text, B = number], { {"one", 1}, {"two", 1} } )

## ISNUMBER() analog

= Value.Is(Value.FromText( VALUE ), type number)

Or:

= "sample" is number // false, = 123 is number // true

## ISTEXT() analog

= Value.Is(Value.FromText( VALUE ), type text)

Or:

= "sample" is text // true, = 123 is text // false

## Convert all columns of table Source to text data type

= Table.TransformColumnTypes(Source,  
List.Transform( Table.ColumnNames(Source), each { \_, type text } ) )

### Expand from nested table all not existing in current table

```
= Table.ExpandTableColumn( buffer, "NewColumn",  
List.Difference( Table.ColumnNames( buffer[NewColumn]{0} ), Table.ColumnNames( buffer ) ),  
List.Difference( Table.ColumnNames( buffer[NewColumn]{0} ), Table.ColumnNames( buffer ) ) )
```

### Expand from nested table only specified in list "fields"

Safe way to expand - it takes only intersection of Difference vs fields

```
// take column Attribute from INPUT_TABLE  
fields = List.Buffer( InputTable[Attribute] ),  
#"Expanded NewColumn" = Table.ExpandTableColumn( buffer, "NewColumn",  
List.Intersect( { List.Difference( Table.ColumnNames( buffer[NewColumn]{0} ),  
Table.ColumnNames( buffer ) ), fields } ),  
List.Intersect( { List.Difference( Table.ColumnNames( buffer[NewColumn]{0} ),  
Table.ColumnNames( buffer ) ), fields } ) ),
```

### Expand from nested table specified in special list "fields" + rename + add prefix

```
fields = List.Buffer( InputTable[Attribute] ),  
#"Expanded NewColumn" = Table.ExpandTableColumn( buffer, "NewColumn",  
List.Intersect( { Table.ColumnNames( buffer[NewColumn]{0} ), fields } ),  
// add prefix to each field  
List.Transform( // rename according to RENAME_TABLE by replacing items in list  
List.ReplaceMatchingItems( List.Intersect( { Table.ColumnNames( buffer[NewColumn]{0} ),  
fields } ),  
Table.ToColumns( Table.Transpose(RENAME_TABLE)) ),  
each "Parent " & _ )  
) ,
```

### Query Folding for SQL ([blog post](#))

```
// Use filter as one of the first actions in Power Query after Sql.Database
```

```
// replicate "IN" clause using List.Contains
```

```
Table.SelectRows( Source, each [OrganizationKey]=11 and  
List.Contains( {6,7}, [DepartmentGroupKey] ) )
```

### Libraries with custom Power Query functions

<https://github.com/Hugoberry/PowerQueryExtensions> + [Hugoberry's Gist](#)

<https://github.com/tycho01/pquery>

<https://github.com/tncclark8012/Power-BI-Desktop-Query-Extensions>

<https://github.com/lmkeF/RM>

<https://github.com/hohlick/PowerQueryModules>

<https://github.com/acaprojects/m-tools>

# Operations with date and time in Power Query

## Time

**#time( hour, minute, second )**

Operator	Left Operand	Right Operand	Meaning
x + y	time	duration	Date offset by duration
x + y	duration	time	Date offset by duration
x - y	time	duration	Date offset by negated duration
x - y	time	time	Duration between dates
x & y	date	time	Merged datetime

## Date

**#date( year, month, day)**

Operator	Left Operand	Right Operand	Meaning
x + y	date	duration	Date offset by duration
x + y	duration	date	Date offset by duration
x - y	date	duration	Date offset by negated duration
x - y	date	date	Duration between dates
x & y	date	time	Merged datetime

## DateTime

**#datetime( year, month, day, hour, minute, second )**

Operator	Left Operand	Right Operand	Meaning
x + y	datetime	duration	Datetime offset by duration
x + y	duration	datetime	Datetime offset by duration
x - y	datetime	duration	Datetime offset by negated duration
x - y	datetime	datetime	Duration between datetimes

## Duration

**#duration( days, hours, minutes, seconds )**

#duration(0, 0, 0, 5.5) // 5.5 seconds

#duration(0, 0, 0, -5.5) // -5.5 seconds

#duration(0, 0, 5, 30) // 5.5 minutes

#duration(0, 0, 5, -30) // 4.5 minutes

#duration(0, 24, 0, 0) // 1 day

#duration(1, 0, 0, 0) // 1 day

Operator	Left Operand	Right Operand	Meaning
x + y	datetime	duration	Datetime offset by duration
x + y	duration	datetime	Datetime offset by duration
x + y	duration	duration	Sum of durations
x - y	datetime	duration	Datetime offset by negated duration
x - y	datetime	datetime	Duration between datetimes
x - y	duration	duration	Difference of durations
x * y	duration	number	N times a duration
x * y	number	duration	N times a duration
x / y	duration	number	Fraction of a duration

Main source is M language specification: <https://msdn.microsoft.com/en-us/library/mt211003.aspx>

Actual version of cheat sheet: <https://github.com/IvanBond/Power-Query-Cheat-Sheet>

## Recommended blogs

<https://bondarenkoivan.wordpress.com/> - Ivan Bondarenko (@ Ivan Bond)

<https://blog.crossjoin.co.uk/> - Chris Webb (@Technitrain)

<http://datachant.com/> - Gil Raviv (@gilra)

<https://www.excelguru.ca/blog> - Ken Puls (@kpuls)

<https://querypower.com/> - Igor Cotruta (@igocrete)

<http://exceleratorbi.com.au/> - Matt Allington (@ExceleratorBI)

<http://excel-inside.pro/> - Maxim Zelensky (@Hohlick)

<http://www.thebiccountant.com/> - Imke Feldman (@TheBiccountant)

<https://powerpivotpro.com/> - Rob Collie, Avi Singh and others (@powerpivotpro)

In Russian:

<https://www.facebook.com/groups/Excelforever/>

<http://www.excel-vba.ru/?s=power+query>

<http://needfordata.ru/blog/>

## Author – Ivan Bondarenko



<https://bondarenkoivan.wordpress.com/>

Twitter: [@ Ivan Bond](#)

[Ivan Bondarenko](#) - 19-Oct-2017