Excel meets Lambda

Andy Gordon and Simon Peyton Jones
Calc Intelligence, Microsoft Research
February 2021

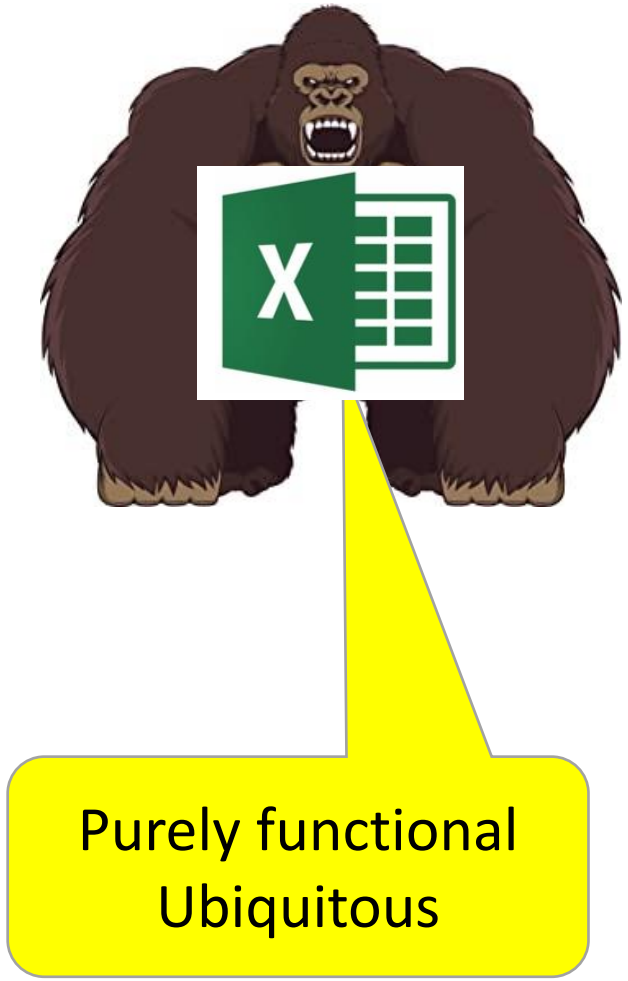
Microsoft Research
Office

Simon’s question (1998)

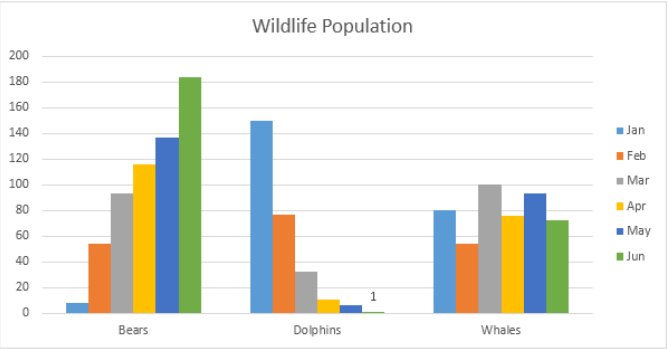
- How can purely functional programming have impact in a company like Microsoft?

Simon’s question (1998)

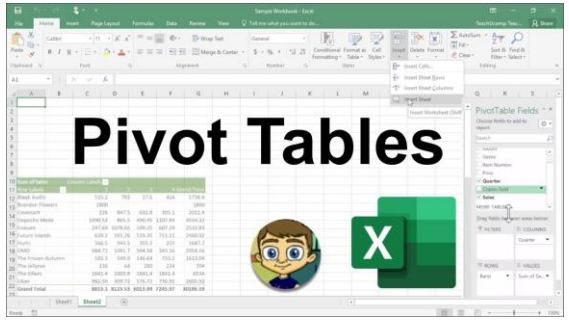
- How can purely functional programming have impact in a company like Microsoft?
- **Realisation**: functional programming is a core business for Microsoft, in the form of Excel
- **Mission**: let’s look at Excel through a functional programming lens, and see what it looks like



Lens 1: Excel as a business modelling tool



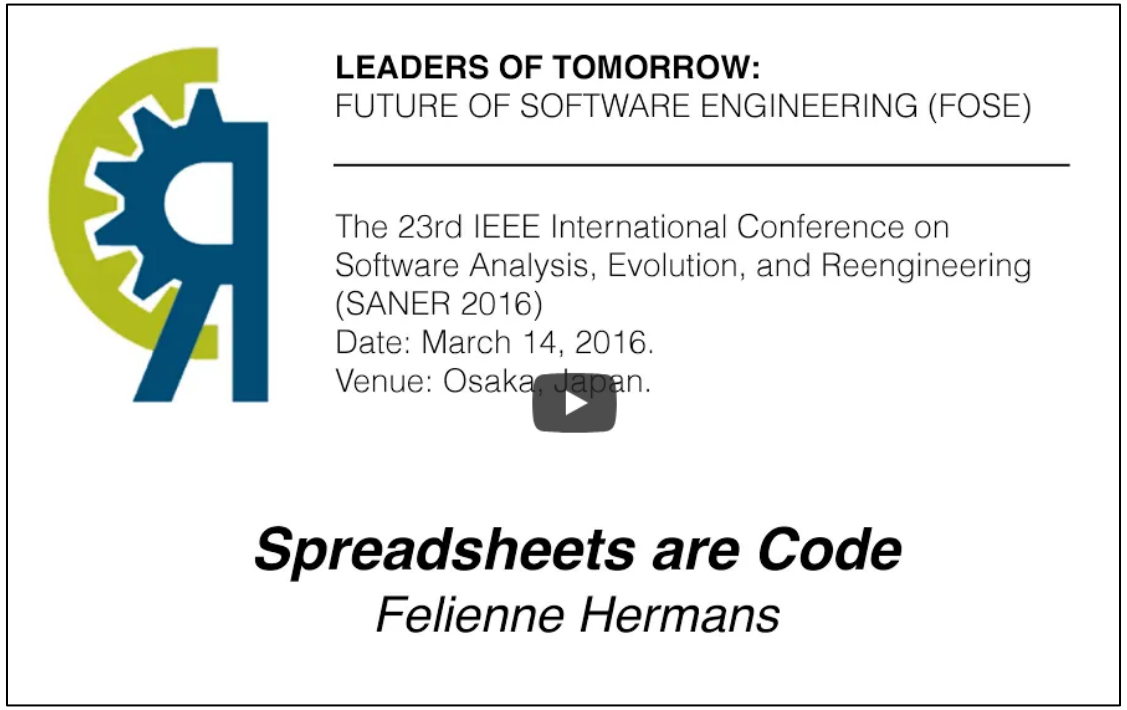
- Tables of numbers, with nice formatting, colours etc
- Charts, visualisations
- Sorting, filtering through user actions
- Pivot tables
- Data import from external sources
- …
- Oh, and formulas too. You can add up columns =SUM(A1:A9). Cool!



Lens 2: Excel as a programming language

- Ambitious spreadsheets have thousands of formulae
- Sometimes developed over years
- By skilled experts

They are complex (purely-functional) programs



But it's an *unusual* programming language

Traditional programming	Excel
Professional programmers	End-user domain experts
Code first (data invisible)	Data first (code invisible)
Edit/compile/run	Spreadsheet always "live"
Programs laid out in time	Programs laid out in space

But it's a rather *weak* programming language

Traditional programming	Excel
Rich data (arrays, records…)	Scalar data only (numbers, strings)
Define new functions all the time	Fixed library of 600-ish functions
GitHub	"I'll email you the spreadsheet"

And yet spreadsheets are so useful/accessible/friendly that people make very ambitious applications

- thousands of formulas (many duplicated)
- dozens of worksheets (or more)

Result: spreadsheets are riddled with errors

$36K spreadsheet error, $1K interest charged

A formula in the spreadsheet picked-up the date, 12/02/98, and interpreted it as a dollar amount

Source: http://www.ed.gov/about/offices/list/oig/auditreports/a07c0009.doc

A £1M council cash error prompts call for the resignation of councilor

"Money" was put in the wrong column, and no one spotted it before the money was handed out, £1M Loss

Source: flintshire-news/2010/02/18/flintshire-county-council-school-cash-blunder-down-to-spreadsheet-error-51352-25856321/

Wrongly grading students

...the spreadsheet had changed all the A grades to A-'s...

Source: http://catless.ncl.ac.uk/Risks/21.94.html

The silent loss of $19,130

Space character in number causes silent Excel miscalculation Error of US$19,130

Source: http://catless.ncl.ac.uk/Risks/20.30.html#subj10

Scientists: Microsoft Excel alters genes, ruins science

...the auto-conversion issue is rampant throughout scientific papers...

Source: https://www.neowin.net/news/scientists-microsoft-excel-alters-genes-ruins-science

MI5 makes 1,061 bugging errors

They bugged 134 wrong phones inadvertently owing to a misunderstanding about the format of phone numbers

Source: http://blogs.mazars.com/the-model-auditor/files/2014/01/12-Modelling-Horror-Stories-and-Spreadsheet-Disasters-Mazars-UK.pdf

But it’s a rather weak programming language

Traditional programming	Excel
Rich data (arrays, records…)	Scalar data only (numbers, strings)
Define new functions all the time	Fixed library of 600-ish functions
GitHub	“I’ll email you the spreadsheet”

Initial focus here

Simple, compositional features that combine, like Lego bricks, to make arbitrarily complex structures

- **User Defined Functions** make Excel express the computations of the domain expert, natively.
  - Any worksheet can become a function: just nominate the input cell(s) and the output cell. Then call it repeatedly.

- **Rich/Compound Data** makes Excel handle the values of the domain expert, natively.
  - Arrays, vectors, records, numbers with units, probability distributions, geo-locations, pictures, media, charts… all can be first class values that
    - Live in a cell
    - Can be the input to, and result of, a function

The product life cycle

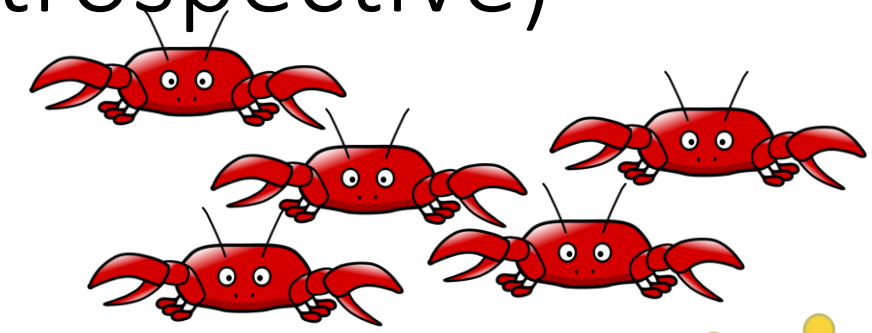
Visit Redmond

Lots of friends
Lots of ideas

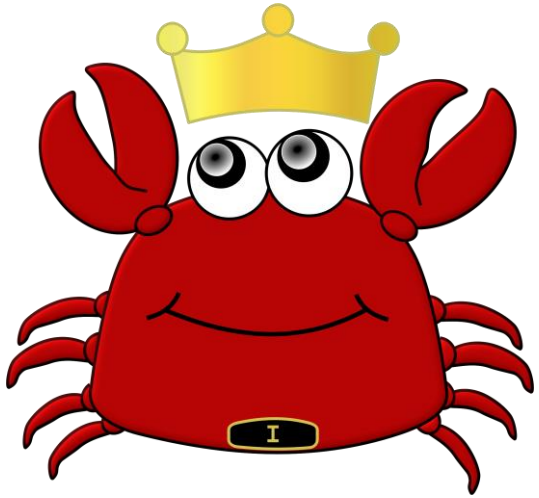
Feature selection for the next release

Alas, we couldn't fit in these cool new ideas

Alternative plan (retrospective)



- Make friends [1998-2006]
- Go away for 7 years
- Wait for one of your friends to be promoted
- Email the Great Boss [Dec 2012]
- Engagement [2013-today]



Also: interestingly, the move to continuous delivery (instead of a Big Release every three years) actually *lengthened* the planning horizon.

Excel as a programming language

Understanding Excel’s formula language

- How hard can it be? A1+7, SUM(A2:B9), etc
- Three things I didn’t know about Excel’s formula language
  - References as first class values
  - Arrays as intermediate values
  - Auto-lifting of functions over arrays

References as first class values

A **reference** to a cell

- =ROW(X) returns the row-number of cell X
- =ROW(C7) returns 7
- =ROW(C7+1) fails
- =ROW(INDEX(A4:A10, 3)) returns 6

INDEX can take a **reference** and return a **reference**

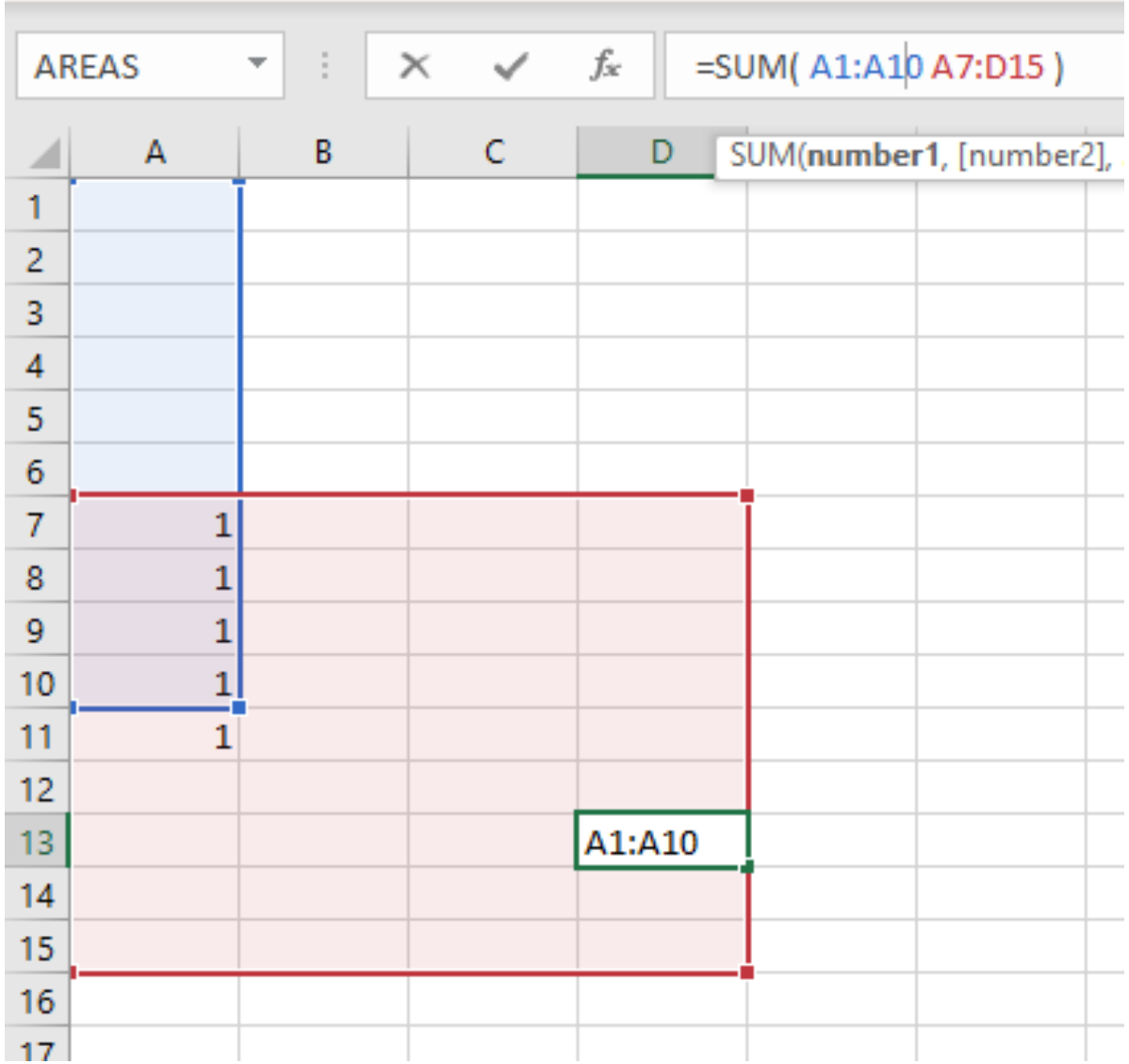
- =SUM(OFFSET(A1:B2, 2, 6)) returns the sum of C7:D8

OFFSET takes a **reference** and returns a **reference** (offset from the input reference)

References as first class values

"Space" is an infix operator on references meaning "intersection"

- =SUM(A1:B10 A7:D20) adds up A7:B10
- "," is range union



Arrays as intermediate values

Range

=SUM(SQRT (A1:A10))

- De-reference the range to get a vector (= 1-D array)
- Apply SQRT to each element of the vector
- Add up the results

Auto-lifting/coercion

What is this "*De-reference the range to get a vector*", and "*apply SQRT to each element of the vector*"?

- SQRT has a "signature": (Number) ->Number

I return a number

I accept (only) a number

- Auto-lifting/coercion rule: when computing F(X), and the value of X is not acceptable to F (based on its signature) then
  - If X is a reference, de-reference it, and try again
  - If X is a string, parse it to a number and try again e.g. SQRT("4.5")
  - If X is an array, apply F to each element of the array

Auto-lifting

- Works for multi-argument functions too

  (*) : (Number, Number) -> Number

SUM(A1:A10 * C2:C11)

- (*) only accepts numbers; but is given ranges
- So de-reference the ranges, and then map down the two vectors

The signature matters

INDEX : ({Matrix,Range}, Number)
-> {Matrix,Range,Number,String,Boolean}

INDEX(A2:A10, {7, 3}) returns the 2-vector {A8,A4}

Range

A literal 2-vector

- First argument of INDEX is acceptable
- Second argument is not
- So we lift only over the second

Auto-lifting and coercion

Excel auto-lifting is done by

- a simple set of rules
- driven by a signature for each function

There are plenty more interesting corners

- Nulls: empty string vs blank cell
- Error values
- Implicit intersection

Story so far

It is quite productive to examine Excel though a programming language lens

So what comes next?

The Two Big Ideas – How Are They Coming?

- **User Defined Functions** make Excel express the computations of the domain expert, natively.

- **Rich/Compound Data** makes Excel handle the values of the domain expert, natively.

Taking Excel beyond text and numbers

Ignite 2018

SORT, UNIQUE, FILTER

Dynamic Arrays

Sort your data		Clean your data		Filter your data		
Names	Sorted	Names	No Dup	Names	Choice	Veg Only
Mary		Peter		Mary	Meat	
Peter		Jane		Peter	Veg	
John		Jane		John	Veg	
Simon		Alison		Simon	Meat	
Charles		Simon		Charles	Veg	
Jane		Mary		Jane	Meat	
Alison		Simon		Alison	Veg	
		Charles				
		Charles				
		John				


Cambridge
2018

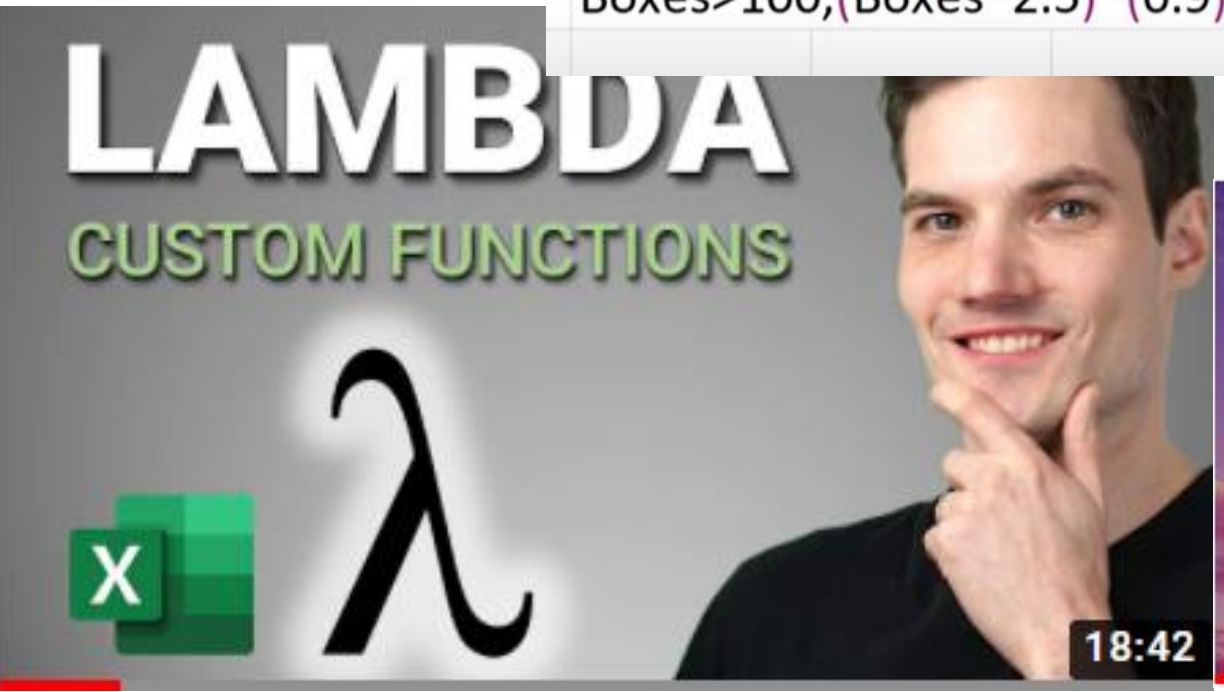
```
=LAMBDA(t,b,a,
if(b="",t, MegaReplace(SUBSTITUTE(t,b,a),offset(b,1,0),offset(a,1,0)
IF(logical_test, [value_if_true], [value_if_false])
```

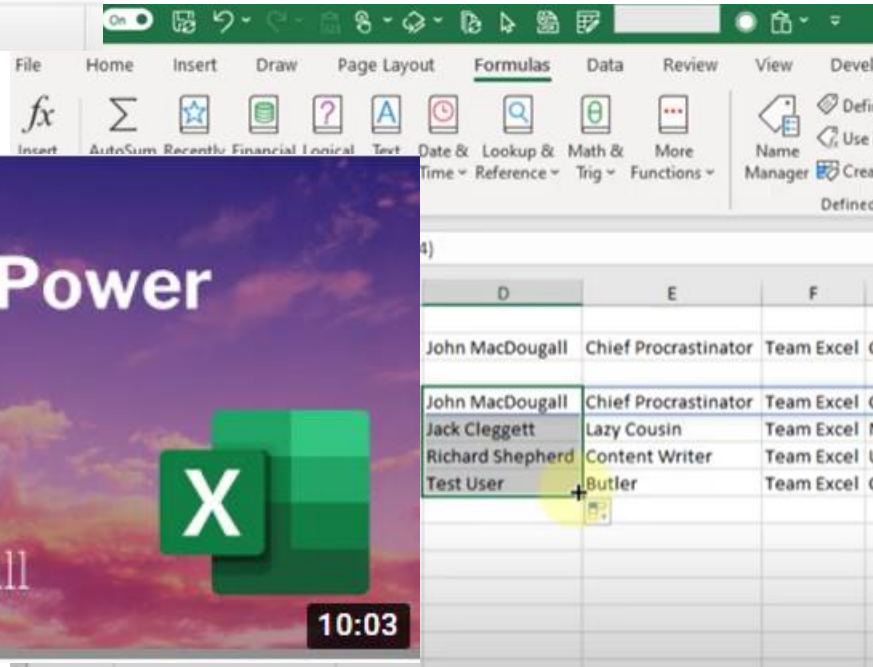
December 2020

```
=LAMBDA(Items, Salary, SORTBY(UNIQUE(Items), SUMIFS(Salary, Items, UNIQUE(Items) ), -1) )
```

	K	L	M	N	O
5	Game	North America	44,675		Utility
6	Game	South America	42,569		Productivity
7	Utility	Europe	43,695		
8	Utility	Australia	34,196		

```
=lambda(Boxes,IF(Boxes>150,(Boxes*2.5)*(0.85),(IF(
Boxes>100,(Boxes*2.5)*(0.9),Boxes*2.5))))(
```



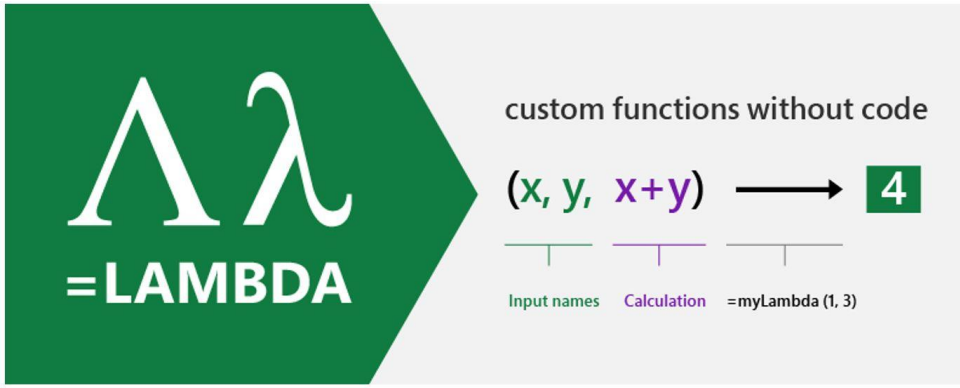


Microsoft Research Blog

LAMBDA: The ultimate Excel worksheet function

Published January 25, 2021

By Andy Gordon, Senior Principal Research Manager; Simon Peyton Jones, Senior Principal Researcher



SD Times SOFTWARE DEVELOPMENT

ITOps Times | Features | Subscribe | Learning Center | Podcasts | Events

API | AGILE | CONTAINERS | DATA | DEVOPS | DEVSECOPS | LOW CODE | MICROSERVICES | MONITOR

Microsoft's LAMBDA language allows new functions to be written in Excel

Latest News | Published: January 26th, 2021 – Jenna Sargent

Microsoft is attempting to make Excel a more powerful application. According to the company, Excel formulas are written by a significantly larger number of people than all C, C++, C#, Java and Python developers combined. It is the world's most powerful programming languag
two main shortcoming

One shortcoming is th
The second is that Exc

custo
(x, y
Input na

Public.

NEWS | POLITICS AND POLICY | BUSINESS AND ECONOMY | ARTS AND CULTURE | LIFESTYLE AND LEISURE

Search Companies, Topics, Organizations, Governments...

MICROSOFT CORPORATION

01/26/2021 | News release | Distributed by Public on 01/26/2021 11:42

LAMBDA: The Ultimate Excel Worksheet Function

Ever since it was released in the 1980s, Microsoft Excel has changed how people organize, analyze, and visualize their data, providing a basis for decision-making for the millions of people who use it each day. It's also the world's most widely used **programming language**. Excel formulas are written by an order of magnitude more users than all the C, C++, C#, Java, and Python programmers in the world combined. Despite its success, considered as a **programming language** Excel has fundamental weaknesses. Over the years, two particular shortcomings have stood out: (1) the Excel formula language really only supported scalar values-numbers, strings, and Booleans-and (2) it didn't let users define new functions.

Until now.

The **Calc Intelligence** project at Microsoft Research Cambridge has a long-standing partnership with the Excel team to transform spreadsheet formulas

Visual Studio MAGAZINE

HOME | NEWS | TIPS & HOW-TO | NEWSLETTERS

VISUAL STUDIO | VISUAL STUDIO CODE | C#/VB | .NET | XAMARIN/MOBILE | TYPESCRIPT/BLAZOR/AS

NEWS

Microsoft's New Programming Language for Excel Now Turing Complete

By David Ramel 01/27/2021

Microsoft, which calls its Excel spreadsheet a *programming language*, reports that an effort called LAMBDA to make it even more of a *language* is paying off, recently being deemed Turing co

Being **Turing complete** is the litmus test of a full-fledge language, marking the ability to imitate a Turing machine **definition**, that means, "A programming language is Turi can implement any possible algorithm with it."

And that's exactly what LAMBDA can now do.

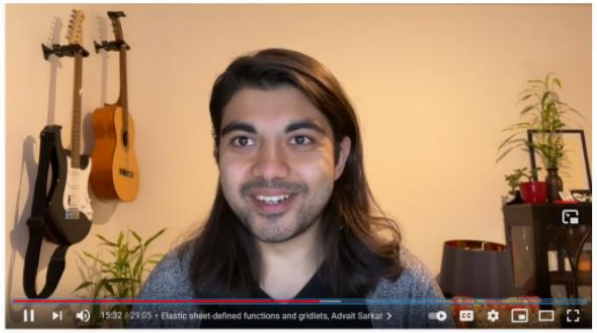
THENEWSTACK | Podcasts | Events | Ebooks | Newsletter | Sponsorship

Architecture | Development | Operations

DATA / DEVELOPMENT

Microsoft Excel Becomes a Proper Programming Language

7 Feb 2021 6:00am, by David Cassel



It may be the oldest piece of software still in widespread use. It was 34 years ago, just three years after Apple introduced its very first Macs, that Microsoft released the first version of its familiar Excel spreadsheet app, initially a rough

COMMUNICATIONS OF THE ACM

Search

HOME | CURRENT ISSUE | NEWS | BLOGS | OPINION | RESEARCH | PRACTICE | CAREERS | ARCHIVE | VIDEOS

Home / News / LAMBDA: The Ultimate Excel Worksheet Function / Full Text

ACM NEWS

LAMBDA: The Ultimate Excel Worksheet Function

By Microsoft Research Blog
January 27, 2021
Comments

VIEW AS: SHARE:

Microsoft's new LAMBDA tool allows you to define your own custom functions in Excel, using Excels formula language.

Credit: Microsoft Research Blog

Ever since it was released in the 1980s, Microsoft Excel has changed how people organize, analyze, and visualize their data, providing a basis for decision-making for the millions of people who use it each day. It's also the world's most widely used *programming language*. Excel formulas are written by an order of magnitude more users than all the C, C++, C#, Java, and Python programmers in the world combined. Despite its success, considered as a *programming language* Excel has fundamental weaknesses. Over the years, two particular shortcomings have stood out: (1) the Excel formula language really only supported scalar values—numbers, strings, and Booleans—and (2) it didn't let users define new functions.

Until now.

The Calc Intelligence project at Microsoft Research Cambridge has a long-standing partnership with the Excel team to transform spreadsheet formulas into a full-fledged programming language. The fruits of that partnership are starting to appear in the product itself. At the 2019 ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2019), we announced two significant developments: data types take Excel beyond text and numbers and allow cells to contain first-class records, including entities linked to external data, and dynamic arrays allow ordinary formulas to compute whole arrays that spill into adjacent cells. These changes are a substantial start on our first challenge: rich, fully-first-class structured data in Excel.

SIGN IN for Full Access
User Name
Password
» Forgot Password?
» Create an ACM Web Account
SIGN IN

MORE NEWS & OPINIONS

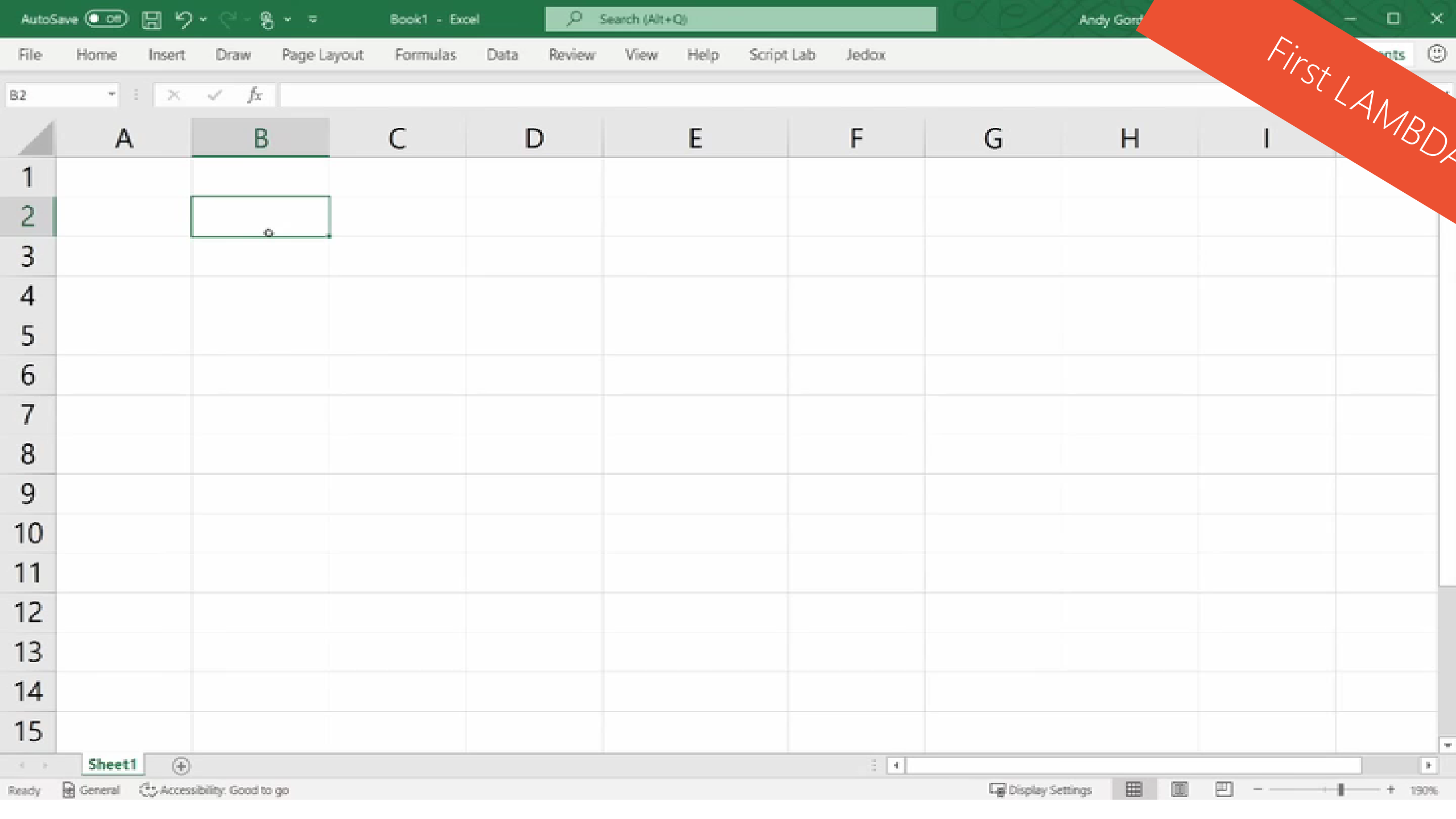
Tweaking AI Software to Function Like a Human Brain Improves Computer's Learning Ability
Georgetown University Medical Center

A Holistic View of Future Risks
Peter G. Neumann

The Cross-Sectorial Collaborative Shared Value Strategy
Orit Hazzan, Ronit Lis-Hacohen, Bella Abrahams, and Mariana Waksman

ACM RESOURCES

Security Awareness (Second Edition) (Part 1): Protecting


AutoSave
Off
Book1 - Excel
Search (Alt+Q)
Andy Gort
File
Home
Insert
Draw
Page Layout
Formulas
Data
Review
View
Help
Script Lab
Jedox
B2
A
B
C
D
E
F
G
H
I
Sheet1
Ready
General
Accessibility: Good to go
Display Settings
190%
First LAMBDA


Bonnie Nardi

End-user programming

"Programming to achieve the result of a program primarily **for personal, rather than public** use. [Nardi 1993]

End-user programmers might be secretaries, accountants, children, teachers, interaction designers, scientists, or anyone who finds themselves writing programs to support their work or hobbies." [Ko et al, 2011]

These people are **domain experts**.

Computers useful but **not intrinsically interesting**.

LAMBDA can help end-users re-use formulas

```
CONVERSIONRATE =
   LAMBDA(from, to, today,
      LET(string, from & "/" & to,
         array, STOCKHISTORY(string,today,today),
         result, INDEX(array,2,2),
         IFERROR(result, "no conversion rate available")
      ));
```

General-purpose financial history function built into Excel

	usd	
	gbp	
01/01/2021	£	0.73
02/01/2021	no conve	
03/01/2021	no conve	
04/01/2021	£	0.74
05/01/2021	£	0.73
06/01/2021	£	0.73
07/01/2021	£	0.74
08/01/2021	£	0.74
09/01/2021	no conve	
10/01/2021	no conve	
11/01/2021	£	0.74

ARRAYMAKE: build arrays with LAMBDA

ARRAYMAKE(m,n,f) = { f(1,1), …, f(1,n);
…;
f(m,1), …, f(m,n) }

Higher order argument

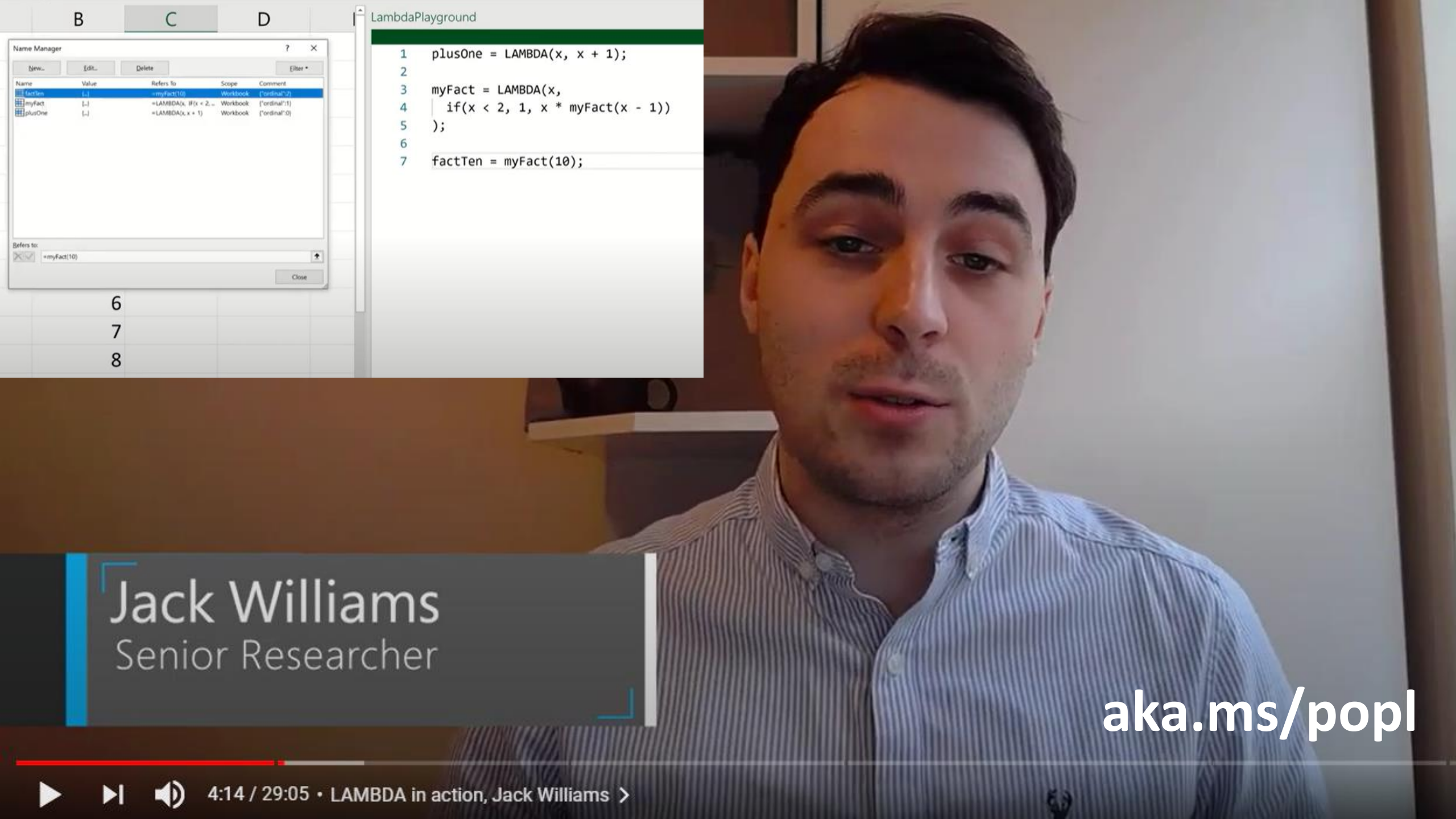
=LET(r,3,c,4,arraymake(r,c,LAMBDA(i,j, j+(c*(i-1)))))

1	2	3	4
5	6	7	8
9	10	11	12

```
VSTACK =
 LAMBDA(array_1,array_2,
  LET(rows_1, ROWS(array_1),
     rows_2, ROWS(array_2),
     cols_1, COLUMNS(array_1),
     cols_2, COLUMNS(array_2),
     blank, " ",
     ARRAYMAKE(rows_1+rows_2, MAX(cols_1,cols_2), LAMBDA(i,j,
      IF(i<=rows_1,
        IF(j<=cols_1,INDEX(array_1,i,j),blank),
        IF(j<=cols_2,INDEX(array_2,i-rows_1,j),blank)) ))
  ));
```

=VSTACK(SEQUENCE(3,3), SEQUENCE(4,2))

1	2	3
4	5	6
7	8	9
1	2	
3	4	
5	6	
7	8	


B
C
D
Name Manager
New...
Edit...
Delete
Filter
Name
Value
Refers to
Scope
Comment
factTen
myFact
plusOne
Workbook
Refers to:
=myFact(10)
Close
6
7
8
LambdaPlayground
plusOne = LAMBDA(x, x + 1);
myFact = LAMBDA(x,
if(x < 2, 1, x * myFact(x - 1))
);
factTen = myFact(10);
Jack Williams
Senior Researcher
aka.ms/popl
4:14 / 29:05 • LAMBDA in action, Jack Williams

Playground (version 1) - Repaired

Search (Alt+Q)

Andy Gordon (RF

ayout Formulas Data Review View Help Script Lab Jedox

TEXTSPLIT

fx Amy,Felienne,Margaret

F	G	H	I	J	K	L	M	N	O	P	Q	R
		Amy,Felienne,Margaret										

LambdaPlayground

```
38
39
40  explode = LAMBDA(string,
41                 arraymake(1,LEN(string),
42                   lambda(i,j, MID(string,j,1))));
43
44
45
46
47
48
49
50
51
```

TAGVIEW: dynamic pivoting on tags

Country	Team
Nigeria	Menen
Ethiopia	
Sudan	
Kenya	

=tagview(A8:B12)

Nigeria

Playground (version 1) - Repaired

Search (Alt+Q)

Andy Gordon

TAGVIEW

Formulas Data Review View Help Script Lab Jedox

fx Amy,Menen

Country	Team	Amy	Bonny	Felienne	Margaret	Menen
Nigeria	Felienne,Margaret			Nigeria	Nigeria	
Ethopia	Amy,Menen	Ethopia				Ethopia
Sudan	Bonny,Amy	Sudan	Sudan			
Kenya	Felienne,Menen,Amy	Kenya		Kenya		Kenya

LambdaPlayground

```
tagview = lambda(table,
  let(table_body, vtail(table),
      step_1, index(table_body,,2),
      step_2, textjoin(",",TRUE,step_1),
      step_3, textsplit(step_2,","),
      step_4, transpose(sort(unique(transpose( step_3 )))),
      primary, index(table_body,,1),
      step_5, arraymake(rows(table_body),columns(step_4),lambda(i,j,
                let(h,index(step_4,1,j),
                    csv,index(step_1,i,1),
                    if(csv_member(h,csv),index(primary,i,1)," "))))),
      step_6,vstack(step_4,step_5),
      step_1));
```

LAMBDA MVP: some limitations to address

- Release the Lambda Playground
- Tail recursion optimization to avoid exceeding stack depth
- Native ARRAYMAKE to support nesting
- Efficient combinators for MAP/REDUCE
- RECORD function to construct new entities
- Can’t use grid positions like A1 or X1 as bound identifiers
- Convenient syntax for array literals

How do end-users share libraries of LAMBDAs?

- “End-user programmers might be secretaries, accountants, children, teachers, interaction designers, scientists, …”
- We think of **LET, LAMBDA, arrays, entities as forming a platform** to empower some domain experts to build domain-specific libraries that embody their expertise.

Share as online workbook?

Share textual code on GitHub?

Share LAMBDAs as tweets?

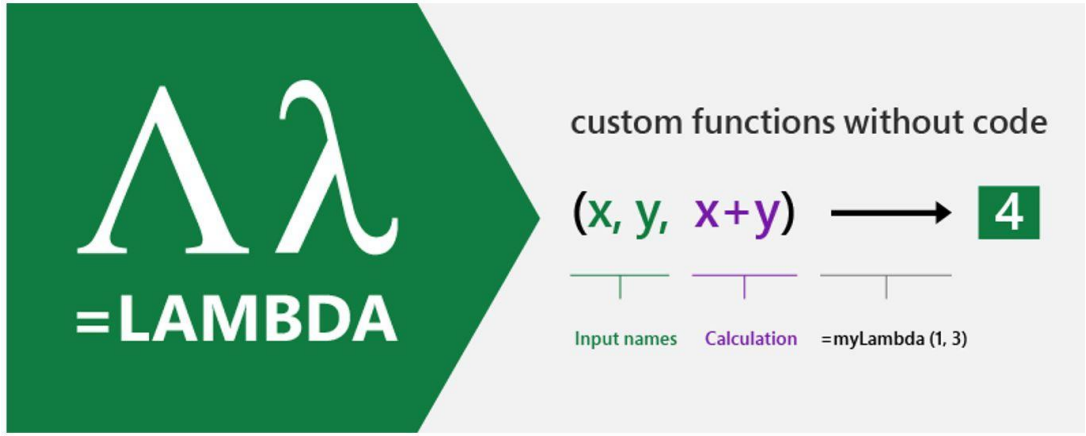
Excel meets LAMBDA

New beachhead for FP!

Grid different from REPL

How to share libraries?

Come join us!



Calc Intelligence

Bring intelligence to end-user programming
aka.ms/calcintel

Subscribe to the Excel Blog to follow the latest product updates
aka.ms/xlblog

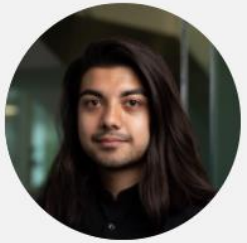
Join the Excel Community to ask questions, get answers about Excel
aka.ms/xlcommunity

Dany Fabian
Principal Research Software Development Engineer

Andy Gordon
Senior Principal Research Manager

Carina Negreanu
Researcher

Simon Peyton Jones
Senior Principal Researcher



Advait Sarkar
Senior Researcher

Sruti Srinivasa Ragavan
Postdoctoral researcher

Neil Toronto
Senior RSDE

Jack Williams
Postdoctoral Researcher

Nicholas Wilson
RSDE II