

From Excel User to Excel Master: Demystifying VBA

About the presenter

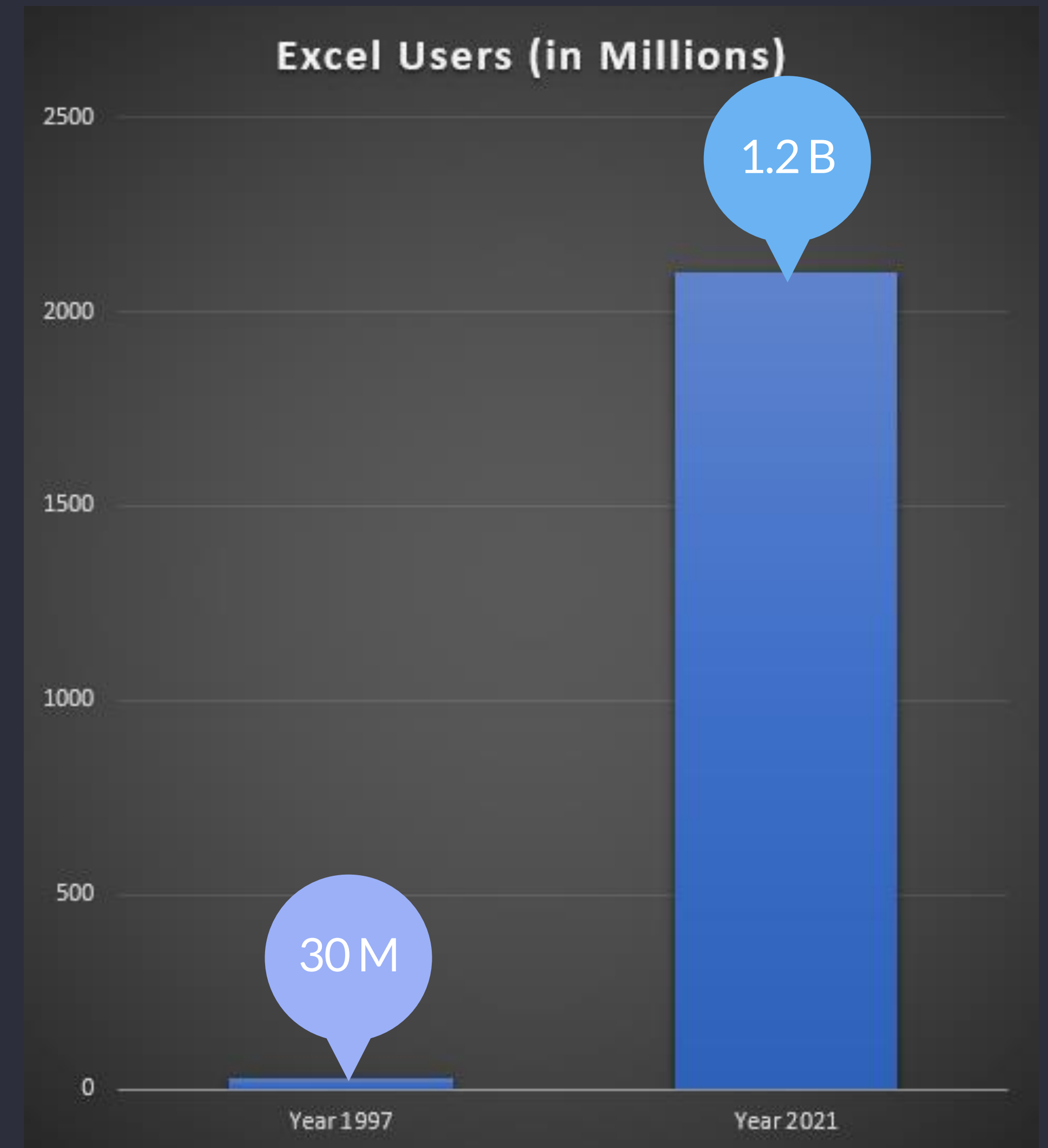
Daniel Strong

- Founder of ExcelVbalsFun.com
- Excel Instructor and VBA programming expert
- Trained over 200K students from 192 countries
- Internationally Bestselling Author & Trainer
- Founding member of the (invite-only) Microsoft Office Creators and Microsoft Tech Community programs



Why Program Excel?

- Most people have Excel
 - In 1996 there were 30 million Excel users
 - Today, there are 1.2 Billion



Why Program Excel?

- Teach Excel to Do Your Work FOR YOU
 - Automate Data Entry and Analysis
 - Make Massive-Level Changes with Precision Targeting
 - Customized Reports, Copying, Manipulating on Autopilot
 - Make Real Programs – Buttons, Automation, Loops, Data Entry Forms...
- Make your life easier through Customization and Automation

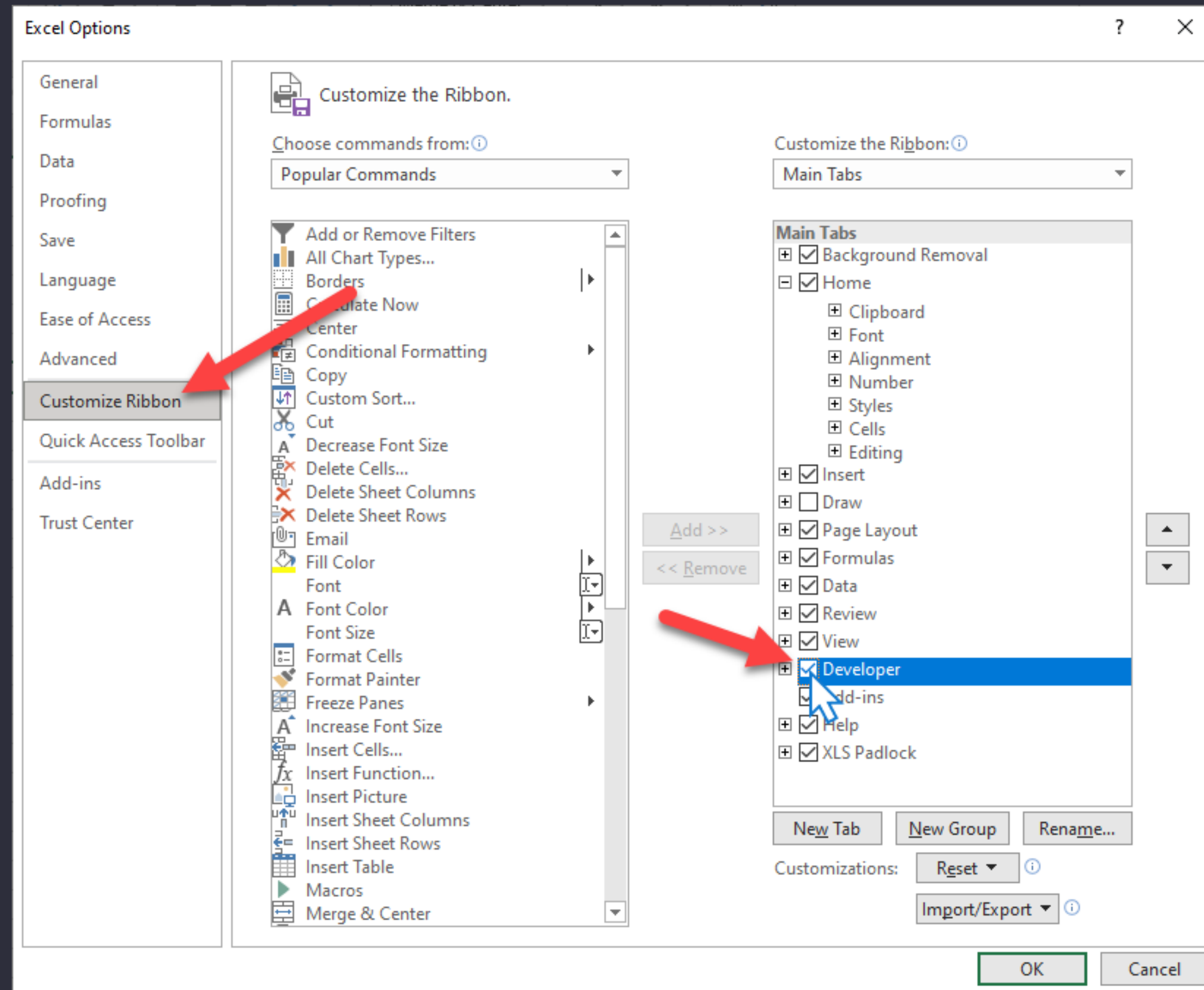
1

The Developer Ribbon

PRESENTATION BY DAN STRONG

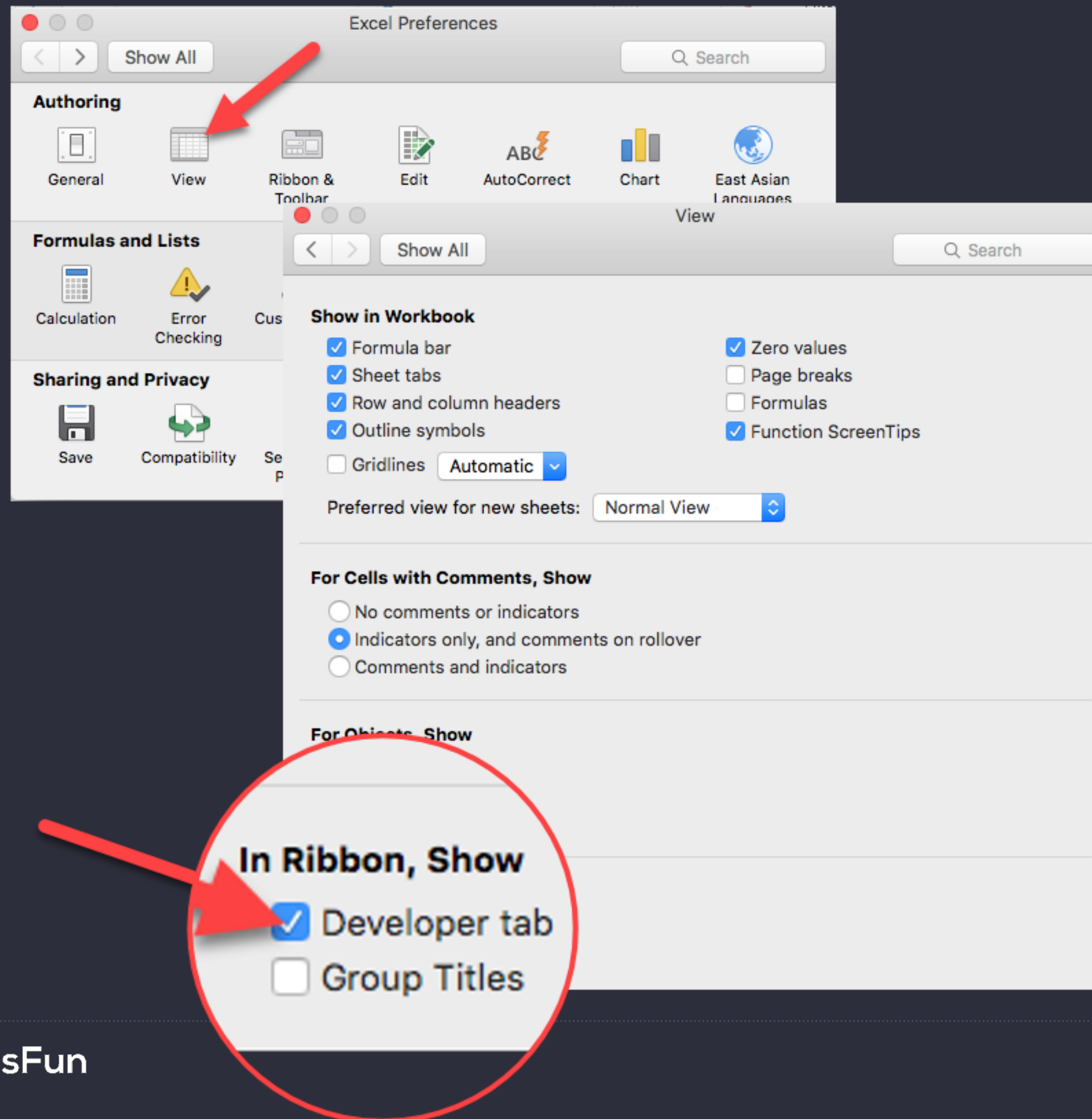
Get the Developer Ribbon (PC):

- File>Options>Customize Ribbon> check “Developer”> Hit OK



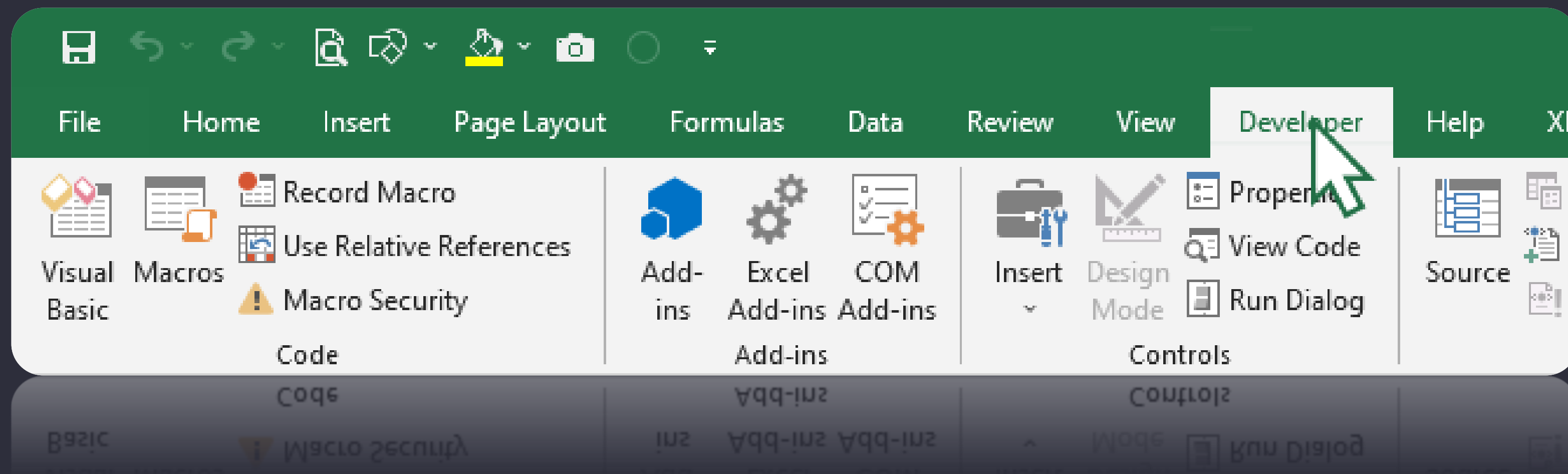
Get the Developer Ribbon (Mac):

- Excel>Preferences>View> check “Developer tab”> Hit OK



Get the Developer Ribbon:

- Enables access to VBA tools, such as Record Macro
 - Record Macro follows your every move/command and translates it into Visual Basic so you can see how Excel processes everyday tasks



2

The Range Object

PRESENTATION BY DAN STRONG

What is a Range?

- A Cell or group of Cells is called a Range
 - Eg: **A1** or **B20:D25**
- (Technical Demonstration)
 - Macro Recorder
 - Debugging = Running code one line at a time
 - Create Procedure (SUB) from scratch

Macro Recorder

- Click Record Macro, press OK
- Select Cell C2, Select cell A1
- Type Hello World! into cell A1, press Enter
- Click Stop Recording
- Alt + F11 to see Visual Basic Editor and view code

```
Sub Macro2 ()  
    '  
    ' Macro2 Macro  
    '  
    '  
    Range("A1").Select  
    ActiveCell.FormulaR1C1 = "Hello World!"  
    Range("A2").Select  
End Sub
```

Debugging (Stepping Through Code)

- You can run a macro instantly or line by line in Visual Basic Editor:
 - Instantly: Press F5 or the 'Run' button
 - Line by line: Press F8 (Cmd+Shift+i on Mac) to run each line, one at a time
 - Yellow line is current line about to be run

```
Sub Macro2 ()  
'  
' Macro2 Macro  
'  
'  
Range("A1").Select  
ActiveCell.FormulaR1C1 = "Hello World!"  
ActiveCell.FormulaR1C1 = "Hello World!"  
End Sub
```


New Macro From Scratch

- Alt + F11 to go into Visual Basic Editor
- Click in new area outside of other Subs
- Type Sub, spacebar, then the name of your macro, press Enter
 - Eg: Sub ShowDate
- Type: Range("A1") = "Date:"
- On next line type: Range("B1") = date
- Run Macro

	A	B	
1	Date:	5/4/2021	
2			
3			

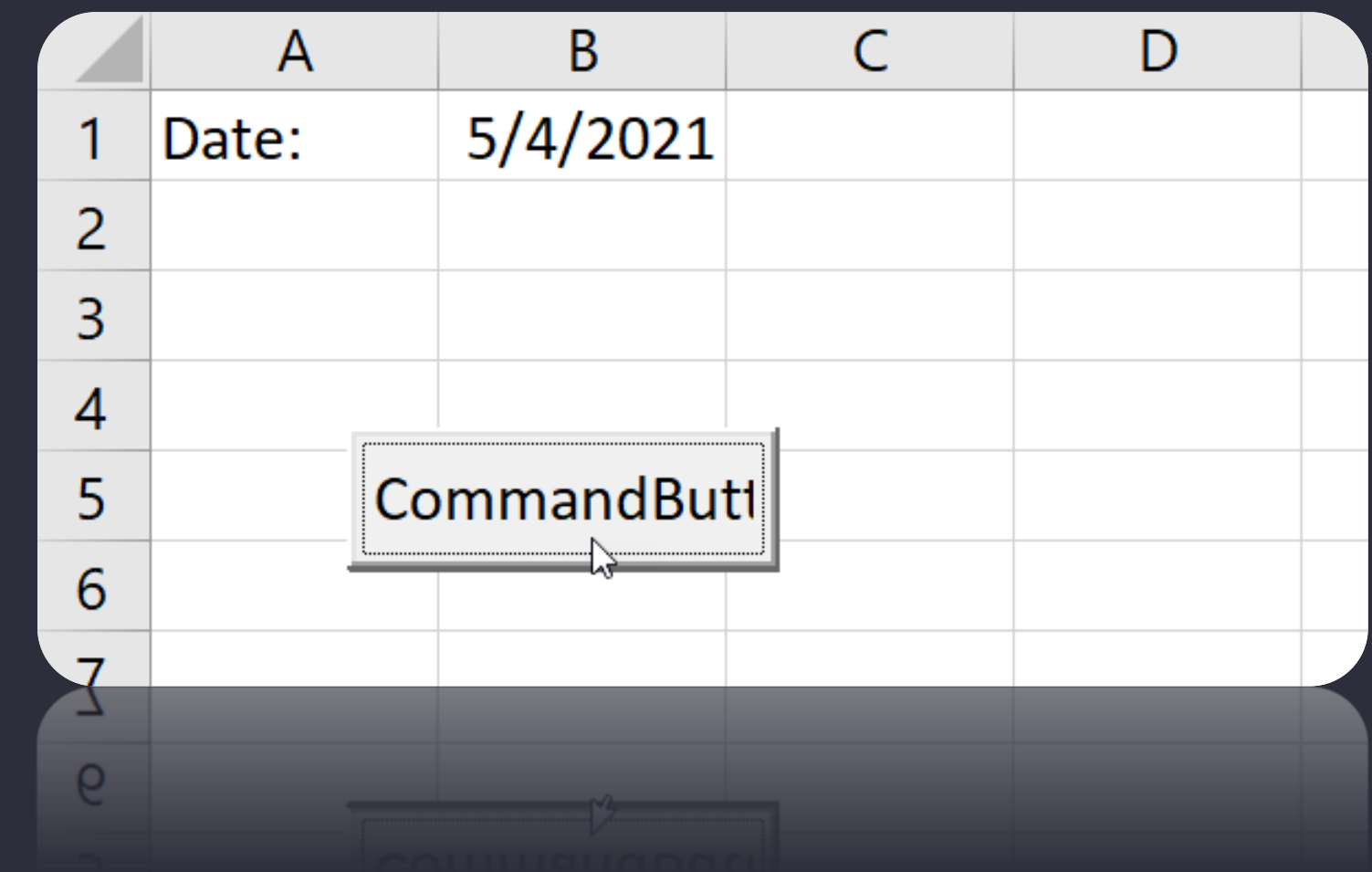
3

ActiveX and Form Controls (Button)

PRESENTATION BY DAN STRONG

Add a Button to Sheet1

- Developer Ribbon>Insert>Activex Command Button
- Click anywhere on the sheet to set button
- Double click on button when in Design mode to edit code for button
- Let's tell it to run **ShowDate** when clicked:
 - Code: Call ShowDate
- Take it off Design mode and click button.
It runs the macro we just designed!!



4

ActiveX and Form Controls (Checkbox and IF/THEN)

PRESENTATION BY DAN STRONG

Add a Checkbox to Sheet1

- Developer Ribbon>Insert>Checkbox (Activex)
 - Checkboxes are either True (checked) or False (unchecked)
 - You can perform If Then statements based on Conditions

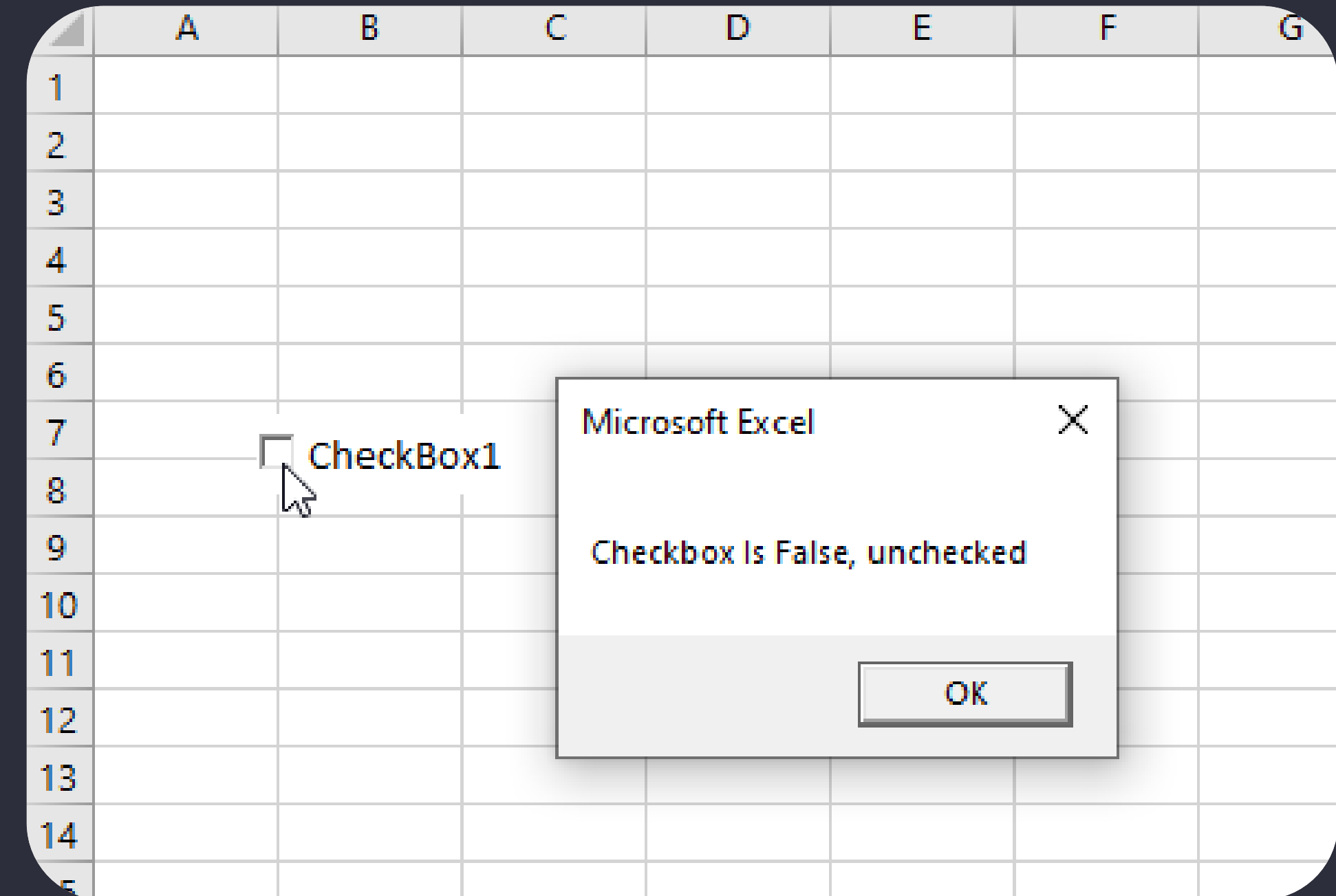
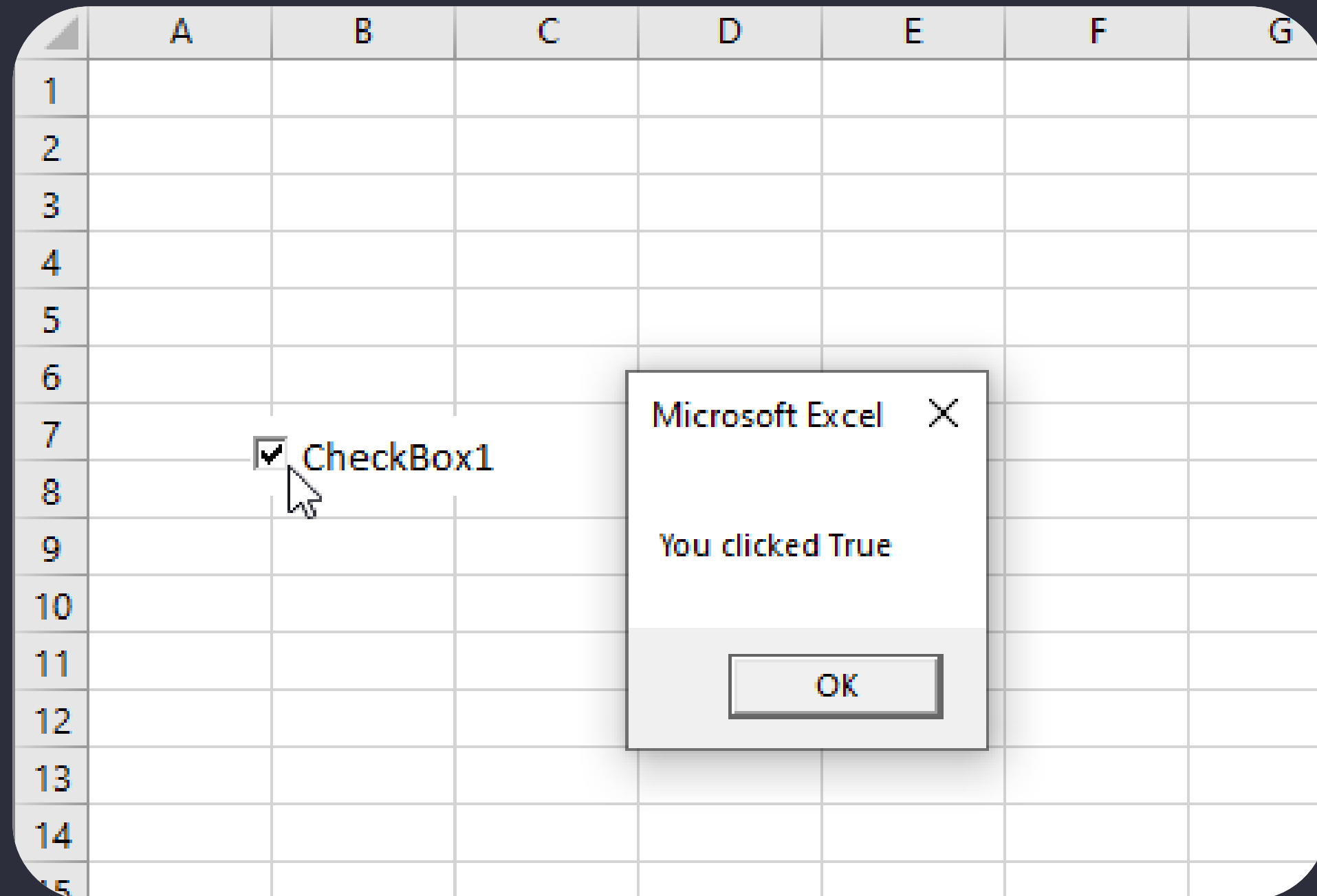
If Then Statements with Checkbox

- Double click on checkbox when in Design mode to edit code
- Let's tell it to give one of 2 messages when clicked!
 - If checked, it will pop up a Message Box saying, "You clicked True"
 - If unchecked, it will pop up a Message Box saying, "Checkbox is false, unchecked"
- Technical Demonstration

```
Private Sub CheckBox1_Click()  
If CheckBox1 = True Then  
    MsgBox "You clicked True"  
Else  
    MsgBox "Checkbox is false, unchecked"  
End If  
End Sub
```

If Then Statements with Checkbox (Cont'd)

- Take it off Design mode and click Checkbox a couple times.



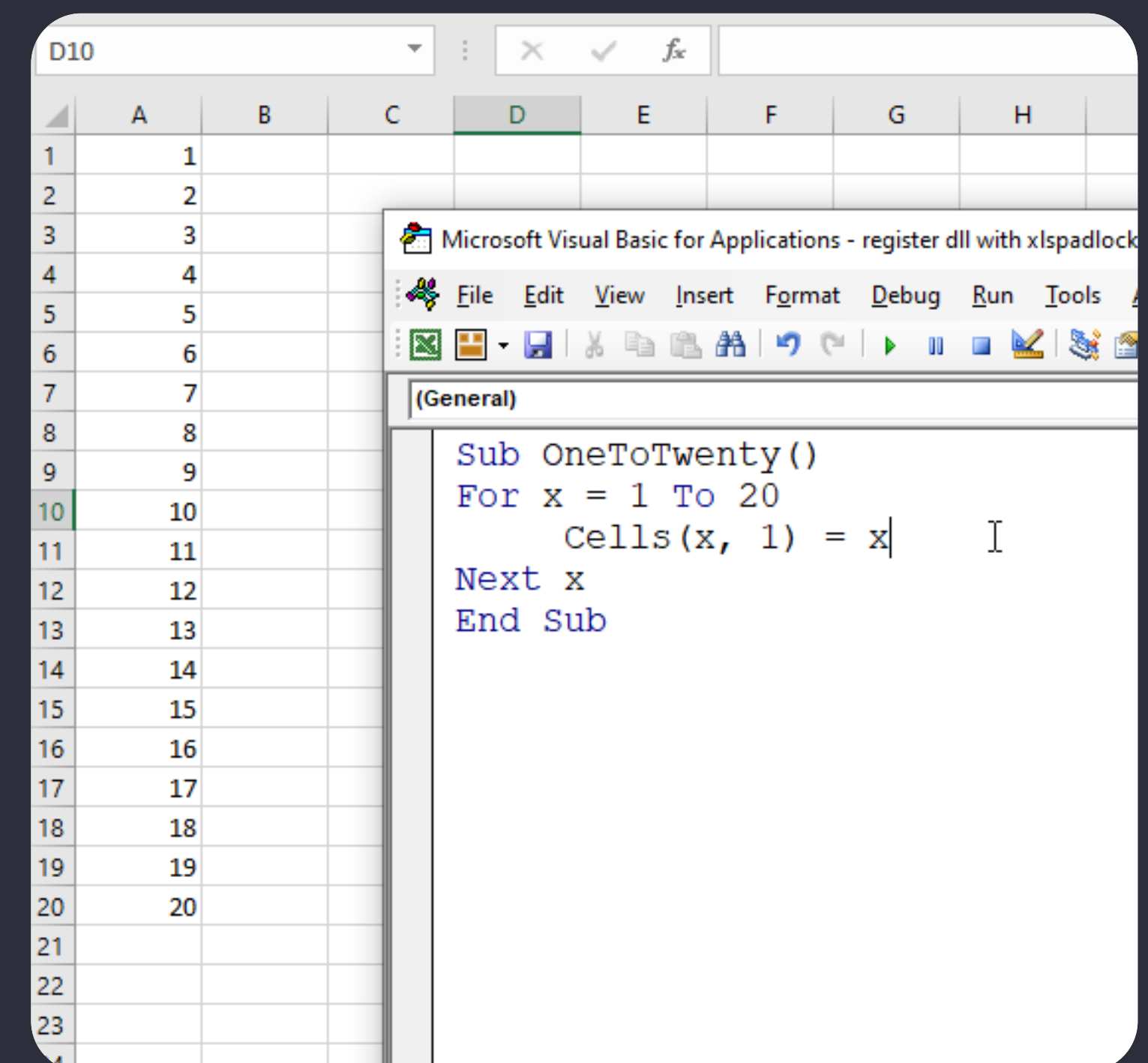
5

Cells Object + Loop

PRESENTATION BY DAN STRONG

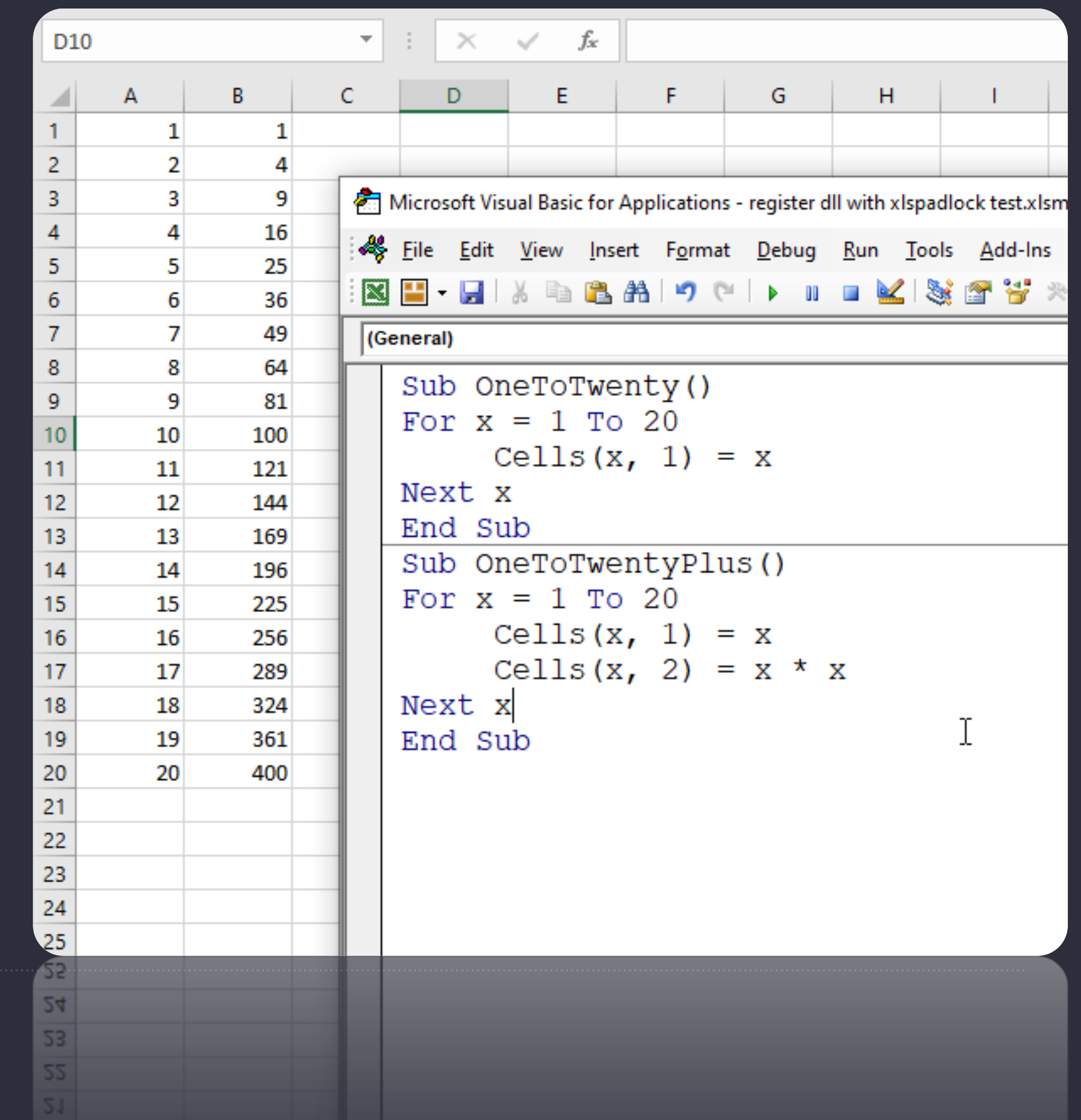
The Cells Object

- Can be used to loop using row,col as numeric coordinates.
- We'll make a new macro called OneToTwenty:
 - Technical Demonstration
 - This places the numbers 1 through 20 on cells A1 through A20 instantly!



The Cells Object (Cont'd)

- You can also manipulate column 2 or others...
- We'll make a new macro called OneToTwentyplus:
 - Technical Demonstration
 - This places the numbers 1 through 20 on cells A1 through A20 instantly and multiplies column 1 times itself into B!



6

Worksheet Events

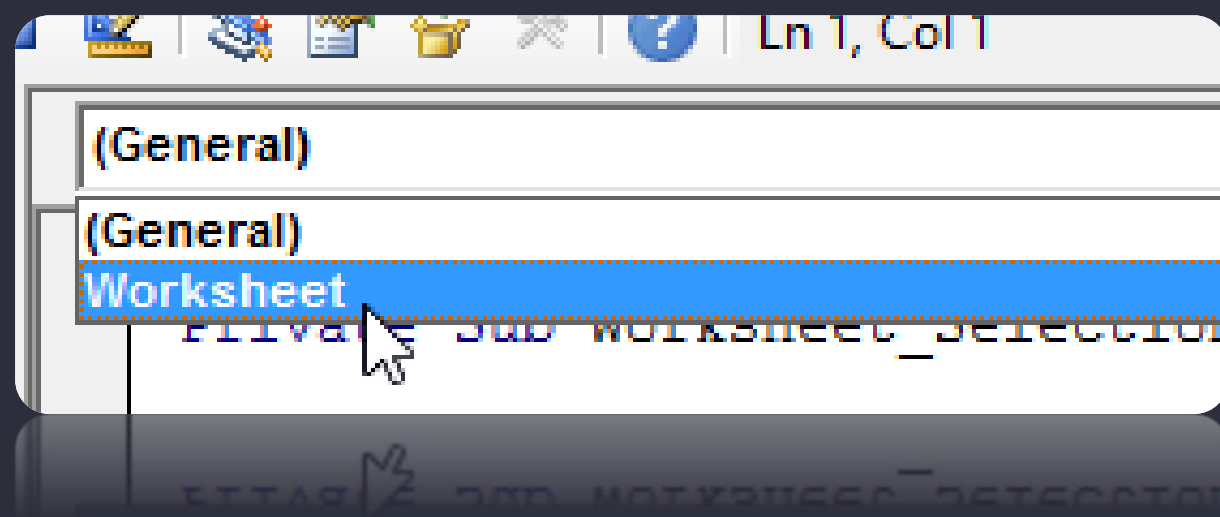
PRESENTATION BY DAN STRONG

What is an Event?

- Events are ordinary User-Actions that can Trigger Macros
- Prevent or safeguard actions
- Auto-hide sheet when deselected, run macro when certain sheet selected, workbook open events, autofit columns on sheet calculations...
- Even change right-click context menu on certain cells with custom items...

Worksheet Events

- Triggered by clicking, selecting, saving, activating cells, worksheets or things on a worksheet.
- Can not be manually triggered like normal macros
- Must change type of macro from (General) to Worksheet type at top:



- **Note:** on a Mac, you must copy/paste the Sub/End Sub portion for Events...

- Technical Demonstration: Cell Change and Selection Change

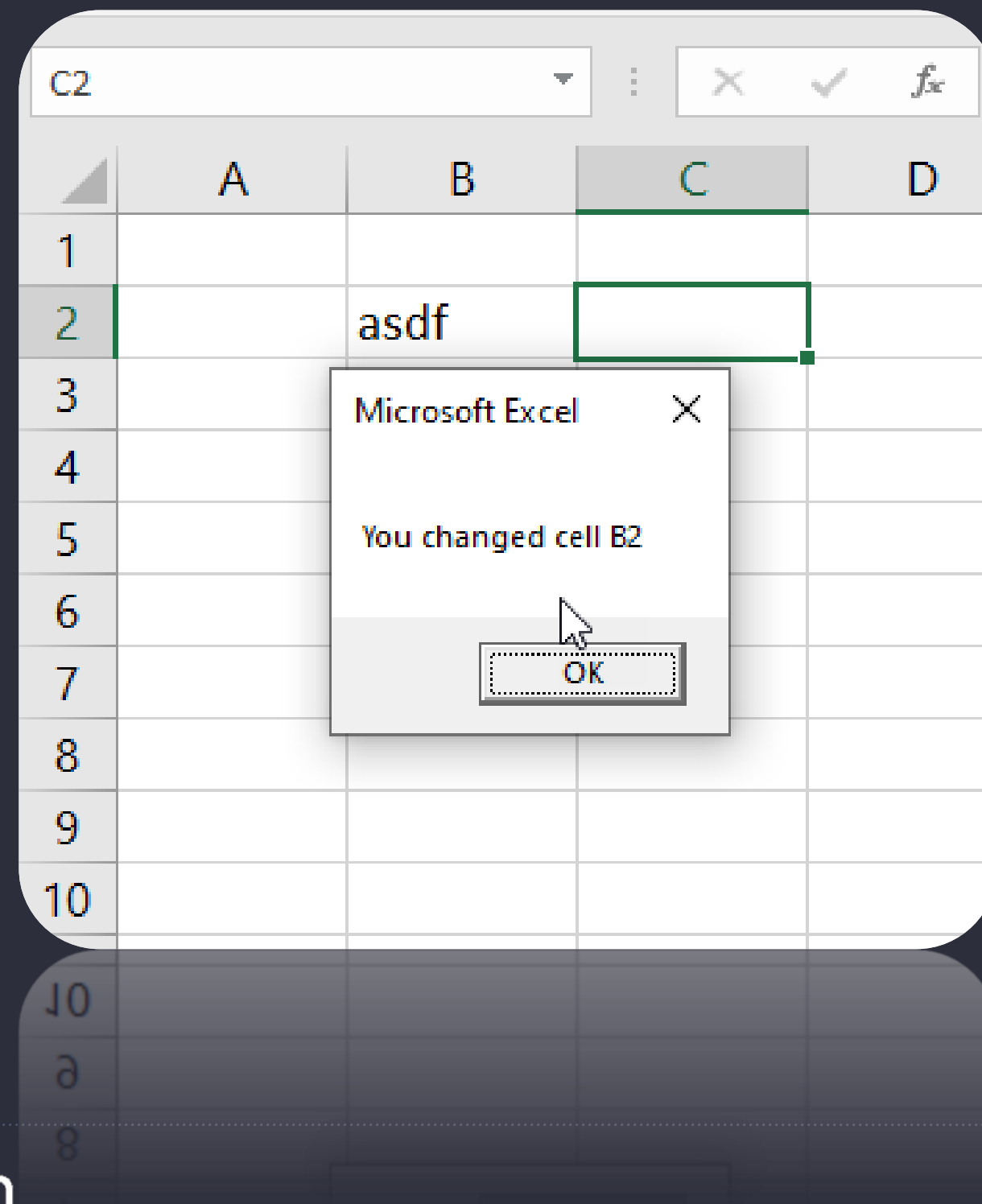
7

Worksheet Events (Cell Change)

PRESENTATION BY DAN STRONG

Worksheet Events: Cell Change

- Triggered whenever a cell's value changes.
- Uses Target to represent cell in question.



Worksheet Events: Cell Change (Cont'd)

- Code:

```
Private Sub Worksheet_Change(ByVal Target As Range)
    MsgBox "You changed cell " & Target.Address(0, 0)
End Sub
```

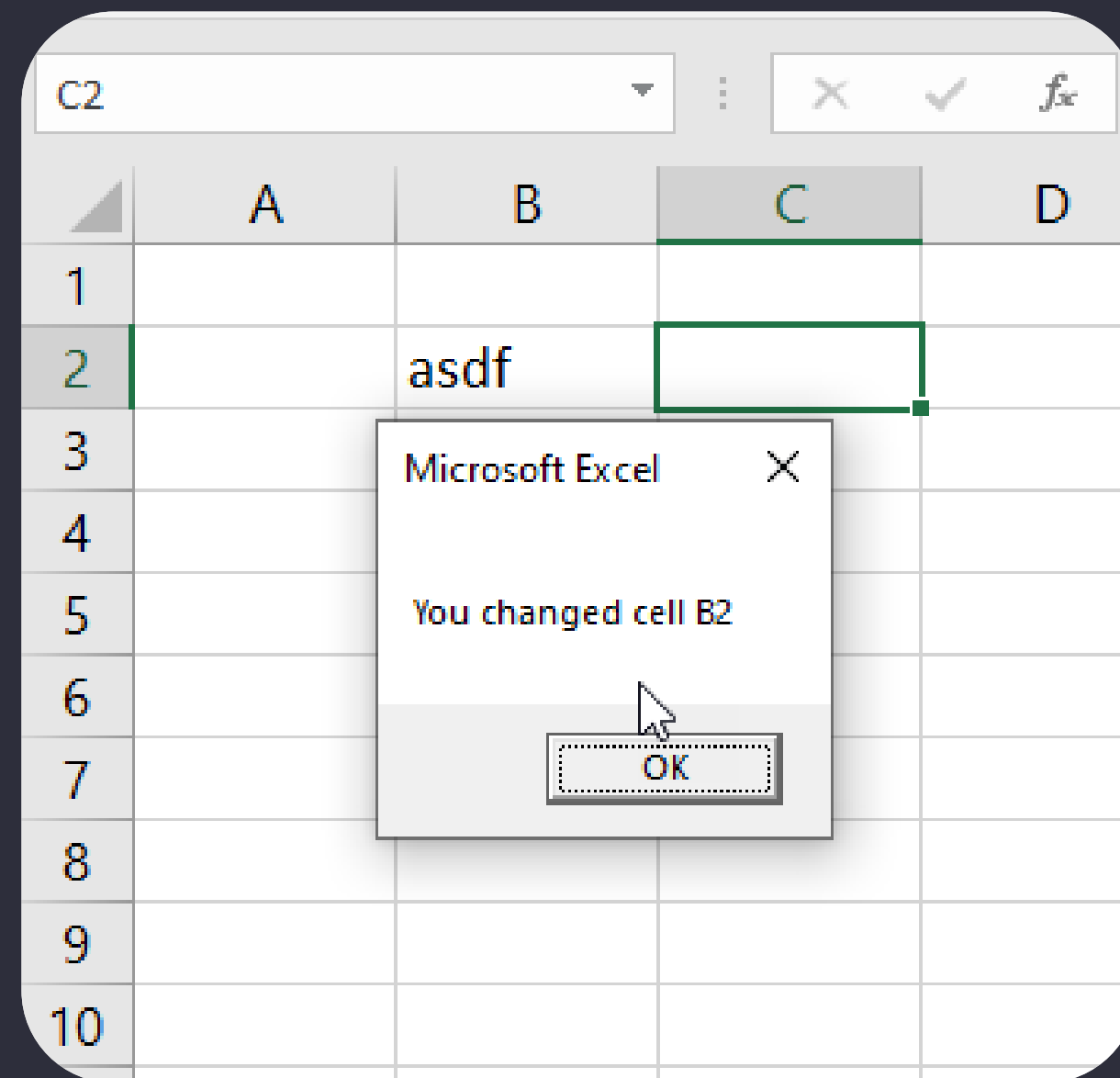
8

Worksheet Events (Cell SELECTION Change)

PRESENTATION BY DAN STRONG

Worksheet Events: Cell Selection Change

- Triggered whenever a cell's selection changes.
- Uses **Target** to represent cell in question.



Worksheet Events: Cell Selection Change (Cont'd)

- Code:

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
MsgBox "You moved cell selection to cell " & Target.Address(0, 0)
End Sub
```