Macros and VBA for Beginners

Simon Sez IT®
Your source for software training.

Table of Contents

Section 1 – Course Introduction

Video: Course Introduction

Deborah: Hello everyone, and welcome to this course on Macros and VBA. My name is Deborah Ashby, and I'm an IT trainer. So, it's my job to design, deliver, and facilitate Microsoft training both online and in the classroom.

Now, I've been using Excel for a very long time. But, I have to admit, it did take me a good few years to really explore the world of Macros and VBA. I think that was partly because I felt it was a bit too difficult, and possibly something that wasn't essential to learn. However, once I did start creating my own Macros and writing code, I got to see how beneficial it was to be able to automate repetitive daily tasks, and then I was pretty much sold.

So, this mini course is really designed for people who currently use Excel, but haven't really used Macros, or VBA. And over the balance of this course, we're going to create a few different Macros to automate tasks. I'm going to show you how to run them and also how to save your workbook correctly. And in the last half of the course, we're going to shift our focus to VBA. So, we'll do things like explore the VBA Editor and the interface, we'll edit some Macro code, and eventually work up to creating our own Macro from scratch.

VBA is such a huge topic that can get very complex. But in this course, we just hope to give you a good foundation from which to build further advanced skills. So, with all that said, it's time for you to grab a coffee because we're about to head into our first module.

Video: Examples of Macros in Action

Deborah: So, we're going to start out this course by taking a look at quite a few different examples of Macros in action. And what I want to do here is really just give you an idea or a taster of the types of things that you can do with Macros, so you can start to think about the repetitive tasks that you need to do in Excel, and how you might go about automating those. So, let's start out with our first example.

Now in this lesson, I'm not going to show you how to create a Macro, we're going to do that in the next lesson. This is purely just some examples of common tasks that you might want to create Macros for. So let's start out with this first worksheet. I've got quite a bit of data on here. And you can see that within my data, I have some blank rows. So this is a fairly common issue you might find if you tend to work with large Data Sets. If you import data in from external systems, or maybe you receive data from clients, sometimes the data comes across with blank rows, and we want to quickly remove those.

So, I've created a Macro that's going to do all of that for me. So all I need to do is click somewhere in my data, and I'm going to run my Macro. Now when it comes to running Macros, I have a few different options. So let's run through all of them, I'm going to jump up to the View Ribbon, across into Macros, and select View Macros. These are all the Macros that I've created. And the one that I'm going to use on this spreadsheet is Delete Blank Rows. Now look what happens when I click Run, you should see all of those rows get removed from my data. And like magic, there we go. So we've saved ourselves quite a few steps by having a Macro to do all of the hard work. Let's move on to the next example.

This is another fairly common task. If you've got a Data Set, it might be that you want to apply some number formatting to one or more rows. And you can see here in column D all the way across to column J, I have figures in here which need to have accounting formatting applied to them. Now my preferred formatting is accounting with zero decimal places. And so I've set up my Macro to do just that. Now this time, I'm going to run my Macro in a slightly different way, I'm going to jump up to my Custom Ribbon called My Macros, and I'm going to run my Macro from here. And there we go. So I now have all of my accounting format applied with zero decimal places.

Now the next example I have here is again, something which you might want to do. And that is quickly formatting data as a Table. So I've got this same Data Set, and I'm probably going to want to analyze this maybe using something like a Pivot Table or a Pivot Chart. But prior to creating my Pivot Table, I will always put my data into an Excel table so that it's easy to update. So I've created a Macro, that's going to format my data as a Table, and apply my favorite table style. So this time, I'm going to run my Macro from my Quick Access Toolbar. And it's this one just here. So click and look at that, like magic, my data is now in a table.

And next example is something that I find myself doing all the time, and that is pasting values. So I have a small data table here that has some sales figures in it for the months of the year, January to December, Q1 to Q4. And then I have some totals. And if you take a look at my totals, I have a sum calculation. So there's a formula sitting behind these cells. Now it might be that I want to send this spreadsheet to a client or somebody else, but I don't want them to be able to edit those formulas. So what I want to do is essentially keep the number but remove the formulas from underneath. And that is a scenario where I would normally paste as values. So I've set up a Macro that's going to do that in one click. So let's first deal with this total row at the bottom.

Currently, I have formulas in there. But if I go up to my View Tab, and let's go back into View Our Macros, I have a Macro in here called Paste Values. Click Run. And now you'll see that when I click on any of these cells, if you look in the formula bar, it's just showing me the number. It's thrown away those underlying formulas. I could do the same thing here for this total range. And this time, I'm going to use my Quick Access Toolbar. It's this little icon just here, Paste Values, and we've thrown away those underlying formulas. Hopefully you're starting to see how quick things become when you automate your tasks. Let's take a look at a few more different examples.

Now on this worksheet, I have the same data, but I want to do something slightly different. What I want to do is apply some Conditional Formatting to the figures in my table. And I want to apply Data Bars. So I'm going to select the range of cells where I want to apply those Data Bars, and I'm going to run my Macro. So I'm going to jump up to my Macros ribbon, and you can see that I've created a button here for Data Bars. And when I click it, my Conditional Formatting applies to those selected cells.

Now, something that I find myself doing all the time is removing hyperlinks from website addresses. So sometimes you might receive a spreadsheet. And if you have any URLs in there, Excel automatically underlines them and makes them a clickable hyperlink. But a lot of the time, I don't want them to be URLs. And the annoying thing with Excel is that if you have quite a lot of URLs, you can only remove them one at a time. So, this Macro to bulk remove hyperlinks is probably one of my favorites. So what I can do here, if I want to remove the hyperlinks is just select all of these cells. I'm going to go up to My Macros ribbon, and I have a Remove Link Macro just here. And like magic, all of those hyperlinks have now been removed.

Now you can even create Macros to automate things like creating new worksheets. And this is a very straightforward Macro. If I click on my new sheet button at the top here, you should see I now get a brand new sheet in my workbook, sheet number 10.

Our next example is highlighting cells in a worksheet. So again, this is using Conditional Formatting. So here I have a table which has some values, again for Q1 to Q4 for 2019, and 2020. And I have this setup to show subtotals and grand totals. Now it might be that I want to highlight all of the cells that are greater than a certain value. So, I've set up a Macro to do exactly that. So, I'm going to highlight the cells that I want to apply this Macro. And for me, that's these two blocks just here. And let's go to our Quick Access Toolbar this time. I have a Macro called Highlight Cells. And when I run it, it's highlighting all of the figures that are over $9,000.

And then finally, we have something slightly different here. On this worksheet, I've set up a little button that's going to calculate the percentage. So just below I have some revenue figures, and then I have the profit. And what I want this Macro to do is tell me what percentage of the revenue is profit. So I've created the Macro, and then I've assigned it to this button. So now all I have to do is click the button, and like magic, I get my percentages.

So that's quite a few examples there of different Macros that you can set up to automate your repetitive tasks. And throughout this course, I'm going to show you how you can create Macros similar to these. I'm also going to show you how you can create buttons like this calculate button, and also how you can create your own Macro ribbons and add your Macros to your Quick Access Toolbar. So now we've seen quite a few examples, it's time to create our first Macro. So grab yourself a coffee and let's dive in.

Video: VBA Explained

Deborah: In this course, not only are we going to be exploring how to create Macros to automate repetitive tasks, we're also going to be taking a look at the underlying VBA code. Now, the idea here isn't to turn you into computer programmers overnight. But, it is definitely worth taking the time to at least be able to understand the VBA Editor Window, be able to read through your Macros, and be able to make minor edits to solve any issues that you might come across. And that's pretty much what we're going to do throughout the balance of this course.

But before we get started with the creation of Macros and delving into VBA, let's start out with just a quick explanation of what VBA actually is. And then I'm going to show you the VBA Editor Window, and some things you should probably turn on, which is going to help you when you're working with VBA.

Now VBA stands for Visual Basic for Applications. And it's not just something that you'll find in Excel. In fact, you'll find it in all of the Microsoft applications. And it's basically the programming language that sits behind the application. So it's used for programming, editing, and running application code. And the way that it relates to Macros is that when you record a Macro in Excel, you're essentially creating VBA code in the background. So it's pretty important for you to be able to jump into that code that sits behind your Macro, and be able to make minor changes and customize it. So let's start out just by taking a look at the VBA Window and how you can quickly access it.

Now, makes sure that you have an Excel worksheet open, and then there's a couple of different ways that you can jump into the VBA Editor. Now the first one is by using the Developer Ribbon. Now if we take a look up at my Ribbon tabs, you'll see that I don't have one there called Developer. And if you look at your own Ribbon tabs, you might not have it there either. And that's because the Developer Tab isn't usually turned on by default. So this is the first thing I would suggest you do if you want to work with Macros and VBA. And it's a very simple process, we just need to jump up to the File tab, to jump into the backstage area, all the way down to Options. And we want to select Customize ribbon. Because this is where we can find all of the ribbons that we have turned on.

And if you take a look on the right hand side, you'll see that towards the bottom, I have the Developer Ribbon currently unchecked. So all we need to do is put a check in that box, click on Ok, and now I have my Developer Ribbon. And this is where you'll find a lot of your commands that are related to your Macros and also VBA. So in this first code group, you can see we have a Visual Basic icon, and there's a keyboard shortcut for this. And I will say that I tend to use this keyboard shortcut more than coming to the Developer Tab and clicking on the icon. So to jump straight into the VBA Editor, if you press Alt + F11, it's going to open up the window.

Now I'm just going to move mine over very slightly, because we've got quite a lot going on here. So I really want you just to understand what it is you're looking at. Now the first thing that's most apparent when you come into VBA is that it is kind of old school. If you remember back to earlier versions of Excel before we got the fancy Ribbon System, we were using Menu

Dropdowns. And that is pretty much what you still have when you go into VBA. So I have Menu Dropdown System at the top. And I also have a Toolbar underneath. And this whole window is basically made up of other windows.

So you can see here on the left hand side I have the VBA Project Window open. And this is showing me currently everything I have going on in any workbooks that I have open. So currently where I'm clicked, if you take a look, it says VBA Project Book Two. And if you look at the Excel spreadsheet I have behind that is just Book Two, I haven't renamed this yet, I haven't saved it, I just basically opened a blank document. It showed me that I have sheet one. And if I double click on sheet one, I get a blank window because I haven't added anything into this worksheet as yet. However, if you look a bit further down, I have another workbook open.

So this was the workbook I had open in the first module of this course, where I was running through all the different examples of Macros. And that workbook was called Examples of Macros. And then underneath you can see the sheets and then in brackets, it has the Macro that I'm using on those particular sheets. And this is just what I've called each of my individual Macros. And then right at the bottom here we have some modules. And I'm currently clicked on module one. And this basically shows me all of the code related to these Macros. So you can see here sheet one, I have a Delete Blank Rows Macro in use.

And then up here we have the code that is actually running that Macro. Underneath that I have the Macro that applies accounting format to my spreadsheets. And you can see there I have the VBA code that controls that particular Macro. And I could carry on going down, you can see I have a different section for each of those Macros in use in that workbook. Now, I don't want you to worry too much about the code and what the code says or what it means. All I really want you to understand here is that the VBA Project Window is kind of a way to browse around the different parts of any open at workbox.

And of course, in the middle here, this big window is the Code Window where we can view our code for different Macros or different modules that we've written, and we can make changes and edits. Now by default, I like to have a couple of other windows open in my VBA Editor. So to open some additional windows, if you jump up to the View menu, I also like to have turned on my Properties Window. So let's click to open that. And all of these windows, you can actually resize just by dragging up and down. So I'm going to drag that down to make that a little bit smaller. And I also like to have open the Immediate Window.

Now I'm going to explain a bit more what this Immediate Window is a bit later on, but I'm just gonna keep this down the bottom for the time being. So essentially, what we're going to be using these windows for throughout this course is I'm going to show you how you can record different Macros in Excel, we're then going to jump into the VBA code and take a look at the code that sits behind those Macros. And I'm going to show you how you can make some really basic modifications without having to know a great deal about programming or VBA. Because once you understand a few key factors, the code is pretty easy to read through and modify.

And then towards the end of this course, we're actually going to create our own Macros from scratch. So we're going to have a blank window open in the VBA Editor, and we're going to start to write out our own Macro. And hopefully, you're going to see how easy it is. Now once you've finished with the VBA Editor, you can just simply click on the cross in the top corner to close that down. Remember, if you need to bring it back again, you can click on the Visual Basic button on the Developer Ribbon. Or, you can just press Alt + F11 to jump straight back in there.

So now we have more of an idea as to what VBA is, and how we're going to use it, and also how we can jump into that Editor. Let's start out by creating a few different Macros. So I'm going to hop over there now and I look forward to you joining me.

Section 2 – Recording a Macro

Video: Recording a Macro

Deborah: The beauty of Macros is that we don't really have to know anything about coding in order to be able to record a Macro. Because in Excel, we have something called the Macro Recorder, which is going to do it for us. And what the Macro Recorder does is it basically records your actions or your steps and converts it into VBA Code. So then all you need to do is save the Macro, and you're free to rerun it at any time on any of your data. And when you do run a Macro, it just basically runs through those steps, and executes what you set it up to do.

Now, if you're looking for the Macro Recorder, you can find it in a few different places in Excel. So the first place you'll find it is on your Developer Tab at the top here, in the Code Group, you have a Record Macro Button, so you can start recording from here. You'll also find the same Button underneath the View Ribbon. In the group on the end, we have a little Macros Group over here. And if we click we have a Record Macro option in there as well. And the final place where you can start your Macros is down here in the Status Bar.

So if you take a look over in the bottom left hand corner of my Status Bar, you'll see I have this little icon here, which when I hover over it says No Macros are currently recording. Now if you don't see this little icon in your Status Bar, you can turn this on. So if you right click somewhere in the blank space, and make sure you have a check next to Macro Recording, it's going to give you a little visual indicator down here. And you can actually stop and start Macro Recording from this little icon in the Status Bar. So you can use any of those three methods in order to start recording your Macro. So let's take a look at the task we need to complete that we're going to record as a Macro.

Now currently in this worksheet, I don't have anything on here. But maybe I have a task every week where I have to create a weekly overview reports by Department. So maybe Sales Information, or whatever it is my company is interested in. So maybe I have to type in some figures or import some figures. And every week, I have to type the same information into the title and format that title in a certain way. So what I'm going to do is I'm going to create a Macro, which is basically going to format my report heading for me with the click of one Button, therefore saving me quite a bit of time.

Now before we record the Macro, let's take a look at how many steps it would take me to complete this task without a Macro. So I'm going to click in cell A1, and I'm going to type the title of my report, Weekly Overview Report by Department. And maybe I want to go up to the Home Tab and I need to make this a little bit bigger. Maybe I need to change the font to let's say Lato Black, maybe that is the font that's preferred by my team. I'm going to change the background fill of this cell. So let's change it to a light blue color. I'm going to give the cell a little bit more room and we're going to adjust a line that into the middle. And maybe I want to make it Italic and also underlined.

Now whilst that hasn't taken me a great deal of time, there is about seven or eight steps there. So what I could do, if I wanted to cut down the amount of time I'm spending doing this every week is I can just create a Macro that's going to do the whole lot for me. So I'm going to clear what I've done here, and we're just going to clear everything. And we're going to use the Macro Recorder to record the steps. So let's jump up to Developer, go to the Code Group, and let's record our Macro. Now the first thing you're going to be presented with is this Record Macro dialog box. So you need to give your Macro a name.

Now something that's really important when you're naming your Macros is that you can't have two separate words. So this is very similar to naming tables or maybe named ranges in Excel, it has to be all one word. Or if you do want multiple words, you need to separate them with something like an underscore. Now when it comes to naming your Macro, always give it quite a descriptive name so that it's easy for you to identify and kind of indicates what that Macro does. So I'm going to call this one Format_Report_Heading separated with underscores. I now get a choice of assigning a shortcut key. So if I want to run this Macro using a quick shortcut key, I can assign my shortcut key just here.

Now it's worth noting you need to be a little bit careful here because many shortcut keys are already assigned in Excel. So common ones we might know of Ctrl + C to copy, Ctrl + V to paste, Ctrl + X to cut. You want to make sure that you're not assigning one of those shortcut keys to your Macro because it will overwrite the original shortcut key. Now I will say that most of the Ctrl + shortcut keys are actually assigned in Excel already. So what I tend to do is just add in a Shift, so my shortcut key is going to be Ctrl + Shift and then a letter. So I'm going to say I want Ctrl + Shift + F to be assigned to this particular Macro. We now have to choose where we want to store this Macro. And you can see by default, it says This Workbook.

Now we do have a couple of other options down here; we have New Workbook, and we also have Personal Macro Workbook. Now if you choose to store this Macro in This Workbook, it means that the Macro is going to travel with this workbook. So if I was to send this to someone else, the Macro is still going to work because it's essentially stored in This Workbook. But note that it's only stored in This Workbook. So if I was to open up a brand New Workbook, I'm not going to have access to this Macro unless I've saved the Macro into my Personal Macro Workbook. When you save into Personal, it makes all your Macros universally available across all of your workbooks. So just remember that key difference.

Now for the kinds of things we're doing here, I'm always going to save into This Workbook. And then finally, at the bottom, you can type in a Description. So again, if you want to make this super descriptive so that you know exactly what this Macro is going to do, I would recommend you do add a Description. Because, if you were maybe to come back to This Workbook in a year's time, you might not automatically remember what this Macro actually does. So if you've got a Description, that's going to help you out a lot. So let's just add something in here very quickly. So Macro to format the heading of the weekly overview report.

Now once you've completed all of this information, as soon as you click on Ok, you are essentially now recording your Macro. So let's click on Ok. And you can see what I'm getting, it's telling me there is already a Macro assigned to that key. Alright, so I should assign a different keyboard shortcut. So that's good to know. Let's change our shortcut to, let's say Ctrl + Shift + H and see if that one works. There we go. Perfect. So I'm now recording my Macro. How can I tell?

Well, if I look upon the Developer Tab, you can see that in the Code Group, it now says Stop Recording. And also my Status Bar when I'm recording my Macro Recorder is going to show as just a little empty square. Now another thing that I know some people worry about is the time it takes you to record your Macro. So you can see that I've been sitting here for a few seconds not actually doing anything. Well, I would say don't worry, the Macro doesn't record time lapses essentially. So, I could sit here for an hour not doing anything, and the Macro isn't going to record that.

It also doesn't record if you're moving your mouse around the screen like this. So don't worry, it's not going to start going around when you run your Macro. It only records steps that you take so actions that you click on. So with all that said, let's now execute what we want the Macro to do. So in A1, I'm going to type in the title of my report. So I’ve typed in my title, I'm going to select that row, go to Home. I'm going to make this font a bit bigger. I'm going to change the font to, we did Lato Black, we changed the background fill, and I also made it Italics and underlined. And another thing we did was we made this row a little bit taller, and then we center aligned.

Now it's important to note here that wherever you click your mouse once you're finished, that's where your mouse cursor is going to be once the Macro has run. So I'm going to say that after it's applied all this formatting, I want it to finish in cell A3. So those are my steps, I now need to stop my Macro from recording. So back up to the Developer Tab and Stop Recording. And there we go, we have successfully recorded our first Macro.

Now once you've recorded your Macros, they do get stored, and you'll find them underneath this little Macros Button just here. And the keyboard shortcut to go into that is Alt + F8. And you'll see currently I'm viewing all of the Macros in all of my Open Workbooks. And I've got another workbook open behind this one that has quite a lot of Macros in it. So I'm going to switch this drop down to This Workbook, and there is the Macro that we've just created. And as you create more and more Macros, they'll be listed out in this middle window. And you'll see that with any of these Macros, you have some options on the right hand side you can choose to run the Macro from here.

You can step through it, you can edit it, you can also delete Macros from here as well if you don't need them anymore. So what we're going to do is I'm just going to cancel on this, I'm going to create a new worksheet and we're going to test out our Macro.

To rerun a Macro, I could go back to that Macros Button and click on Run. But remember that we assigned a keyboard shortcut to our Macro, which was Ctrl + Shift + H. And there we go, look at that. Our Macro has run, all of those steps have been executed, which has saved me probably about a minute or so in time.

Now once that doesn't seem a lot, this is a very basic Macro that we've created. And you can imagine for a lot longer complex tasks, this is gonna improve your efficiency greatly. So if you managed to follow all of that well done. That is our first little Macro, and we're going to now progress on to taking a look at more complex Macros.

Video: Using Relative Referencing

Deborah: In the previous lesson, I showed you how to create this very simple Macro just to apply some formatting to a report heading. But there's something really important here that you need to watch out for.

So, if I click somewhere else on my worksheet, and maybe I want to apply this same heading with the same formatting somewhere else on this worksheet, so maybe down here in row 14. So if I click in row 14, and then run my Macro, so let's click on our Macros Button, and I'm going to say Run, you can see it looks like nothing happens. The only thing that happens is that my cursor jumps from cell A14, back up to A3 because that is where my Macro ends.

Now the reason why nothing's happened is because it's essentially just applied the same Macro over the top of the heading we already have in this worksheet. And that is because by default, when you record a Macro, it records using Absolute Referencing. Now, if you're an Excel user, you may be used to the terms Absolute versus Relative Cell Referencing. But just in case you're not, let me just explain to you what that actually is. And I'll explain it to you in the form of formulas calculations in Excel. So if I just click somewhere in here, and I'm going to type in some random numbers, let's just do a few of them, and I'm going to do some more just here.

Now, if I wanted to add up these numbers, to sum them, effectively, I could use a Sum Formula. I can select both of my numbers, I can close my bracket and hit Enter. And then I can copy that formula down. Now by default, these cells are using Relative Cell Referencing. Which means that Excel knows that if I drag a formula down, it needs to modify those cell references, and move each one down one. So that's Relative Referencing, and that's the default in Excel.

However, there is something else called Absolute Referencing. So if I, let's just type in over here, tax, now we're going to say 15%. So if I now want it to work out what 15% of all of these numbers were, I would do my calculation a slightly different way. I'd still do a sum, I would say, I8, multiplied by the figure, we have 07. But what I need to do here is I need to make 07 Absolute by locking the cell. And we do that by pressing F4, which puts dollar signs in front of the row and the column. And what that means is that when I drag this formula down, Excel isn't going to adjust the cell reference where we have our percentage, because we've effectively locked it to that cell.

So that's the difference between absolute and Relative Referencing. And we have something similar when we're working with Macros. So as I mentioned, by default, Macros use Absolute Referencing. So that means when I rerun this Macro, because I ran it in cell A1, it says in the Code, always start in A1. And so it doesn't matter where I click in this worksheet, it's always going to add this report heading and the formatting into cell A1. And you'll notice when we went to Sheet 2 to test this out, again, it added it into A1, which fortunately, in this case is where I want it to be. But it just means I can't really place it anywhere else in my worksheet.

Now, that's a pretty big limitation when you're working with Macros. But fortunately, we have an option that allows us to use Relative References when we're creating our Macros. And if you take a look up on the Developer Tab in the Code Group, you'll see we have a big old Button here that says Use Relative References. So what I want to do here is record a Macro that's going to use Relative References to populate some Departments and also some Sales Figures. So the first thing we need to ensure that we do is we need to click in the cell where we want to start inputting our data, first of all before we start recording the Macro. And we also want to make sure that we turn on Use Relative References prior to recording the Macro.

Once I've done both of those things, I can click on record Macro, I'm going to give my Macro a name. So let's call this Department_Sales_Figures. I'm going to assign a keyboard shortcut of Ctrl + Shift + S. I'm going to store the Macro in this workbook. And just to save a bit of time, I'm not going to add a Description at the moment. So let's click on Ok to start our Macro. So now I'm going to type in the names of the Sales Teams. So we're just gonna say Sales Team 1, Sales Team 2, 3, 4, and 5. And then we're going to populate the Sales Figures for these teams. So 3000, 4000, 5000, let's say 1000, and 6000. And I'm going to give these some currency formatting. So into my Number Formatting, and I'm going to select Currency. And I'm going to take those decimal places down to zero.

And remember it really important that when you finish your Macro, don't leave your range selected, because that's how your Macro is going to end. So always click on a different cell in your worksheet just to end the Macro. So that's that done. I'm going to jump up to my Developer Tab, and I'm going to stop recording my Macro. And if I click on my Macros Button, I can see my new Macro sitting just there. So let's test this out. I'm going to jump across to Sheet 2, I'm going to click in cell A4, and I'm going to run my Macro. And if you remember, the keyboard shortcut was Ctrl + Shift + S. And there we go like magic.

Now, I used Relative Referencing. And if we take a look at this first worksheet, I actually did start this in cell A4. And I've also decided to populate this information in cell A4 here as well. So just to demonstrate Relative Referencing working, if I wanted to paste this information somewhere else on the worksheet, if I click in a cell over here and run my Macro again, it populates elsewhere in the worksheet as well. If I click down here, Ctrl + Shift + S, it's going to put the information in there as well. So that is the difference when it comes to using Relative versus Absolute Referencing.

Now you might be thinking to yourself, well, isn't this one the most common it? Wouldn't it be more common to use Relative References as opposed to the default which is Absolute Referencing? Well, I would say yes. In my experience, I tend to use Relative References when I'm recording Macros all the time. But there are definitely some instances where Absolute Referencing is the preferred method. And we're going to look at a few of those a bit later on in this course.

But for the time being the main takeaway here is that you grab the difference between Absolute versus Relative, and you can create yourself a Macro that uses Relative References, and then test out on different parts of the worksheet. That's it for now, I will see you in the next lesson.

Video: Useful Macros

Deborah: Hi everyone, and welcome back. In this lesson, we're going to move it up again now, and I'm going to show you how you can create a Macro that's a little bit more complex, and involves numerous different steps in order to reformat this entire table of data.

Now before we get to doing that, I just want to show you something that's super important when you're working with Macro files in Excel. You can see currently at the top, I have this yellow bar, which is basically Excel telling me that I can't save my file, because I need to save it as a different file type. And you'll find this when you start recording Macros and working with Macros in Excel. The next time you come to save your file, it's going to ask you to save it as a Macro-enabled file type.

So as we all know, by default, when you create a workbook in Excel, the file extension is .xlsx. And if you look up in my Title Bar at the top, you can see that's why I currently have this workbook saved as. But I've played around with this file a little bit, I've just tried to resave it. And now it's telling me that I need to save it as a Macro-enabled file type. So that's the first thing I'm going to do, let's click Save As, this is the folder I want to save it into. But I need to change my file type. Because I need to save this as a .xlsm file.

So I'm going to select that file extension, click on Save. My warning message has now disappeared. And if you look in the Title Bar, you can see I now have that new file extension, which basically tells people that this is a Macro-enabled workbook. This also means that if you were to send this workbook to somebody else, when they open it depending on their settings in Excel, a lot of the time people will have their Excel settings to warn them when they're opening a Macro-enabled file type.

Because, there are some Macros that can contain malicious computer Code that can cause damage to your system. So just be aware of that when you're either receiving files that contain Macros, make sure you trust the source prior to opening them.

Now with all that said, let's get on to our task for this lesson. We've already seen how to record a couple of basic Macros. So now I'm going to show you something a little bit more complex. And it works along the same principle, we're going to be using the same method to record our Macros, but we're just going to be doing a lot more steps to tidy up this data. And you can see on the right hand side, I just have a list of instructions as to the different steps that I want to do with this data. And I would recommend you writing out the steps you want to take because what you'll find is if you want to record a particularly complex Macro that has lots of different steps, you don't want to be getting it wrong halfway through recording the Macro or forgetting what step comes next.

Sometimes it's quite helpful to write it out so that you have something to refer to as your recording. So let's take a look at exactly what I want to do here. The first thing I'm going to get the Macro to do is split up the Customer Name into two columns. So where we have Customer Name just here, I basically want to insert another column. And then I want to have this one called First Name and a new column called Last Name. And I want to split up the Customer Name so that we have their First Name in column A and their Last Name in column B.

The next thing I want to do is remove the three zeros from the Order Number. So if you look at all of these Order Numbers, the three zeros are kind of a bit unnecessary because they're the same for all of the Order Numbers. I'm going to make this a bit shorter by removing those and just leaving me with a number that's different. The third thing I'm going to do to this data is I'm going to go in and remove all of the hyperlinks from these Customer Email Addresses. And then I'm going to only show the last four digits of the Credit Card Number.

So here we have the customer's full Credit Card Number, and actually all I want to display in this report is the last four digits. And then I want to have some Asterisks where the rest of the numbers are. And then the final step I want to take here as I then want to just Autofit the entire table. So we've got quite a lot to do. The final thing I would advise you to do before you start running complex Macros is create a backup copy of your worksheet. So I'm going to create a copy of this complex Macro worksheets simply by holding down my Ctrl key, clicking, and dragging. And you can see it says Complex Macro 2, and I'm going to just rename this to, let's just call this Test.

Now the reason why I recommend you take a backup is that Macros are very hard to reverse. So if you run your Macro and it goes a little bit wrong, or you haven't got it quite right, you can't click the undo Button or do Ctrl + Z to back out of it. You're kind of stuck with the result. So I would always create a backup copy, run the Macro, and then if the Macros run correctly, I can delete the backup copy. If not, then I can reinstate the original fairly easily, and I haven't lost anything. So let's now record our Macro.

Now I'm going to click where I want the Macro to start. So I'm going to click somewhere in column A. And in this case, I do want to use Relative References, so I'm going to make sure that I have that Button turned on as well. So let's jump up to Record Macro, I'm going to give my Macro a name. So we're going to call this Reformat_Customer_Table. And I'm going to give this a shortcut key of Ctrl + Shift + T, I'm going to store the Macro in This Workbook only. And at the moment, I'm not going to give it a Description, just to save a little bit of time. So let's click on Ok, I'm now recording my Macro. So let's start running through our different steps.

The first thing I want to do is split up the Customer Name into two columns; First Name and Last Name. So I'm going to insert a column, and then I'm going to use the Text to Functions feature in Excel, which will find on the Data Tab in order to break up this Customer Name. So let's say Text to Columns, it’s a Delimited file type. And what Delimiter do I have separating the First Name from the Last Name? Well, it's a Space which I have selected just here. And you can see just below how that's going to break up that name, which is perfect. Click on Finish. And there we go.

So now I'm just going to change these headings. So the next thing I want to do is remove the three zeros from the Order Number. And to do this, I'm going to use a feature in Excel called Flash Fill. So this is going to involve me inserting yet another column and Flash Fill is great, you basically just tell it what you want to pull out of this string of characters. And for me, that is 1, 2, 3, 4, 5. And then if you jump up to the Data Tab into the Data Tools Group, we have a Flash Fill Button, which is going to fill down the rest. I can then use Ctrl + Shift + Down to copy those, and paste over the top of those Order Numbers, and then delete out that column that I no longer need.

The next thing I want to do is to remove the hyperlinks from these email addresses. So that's a simple case of selecting them all, right click, and Remove Hyperlinks. And then finally, I want to only show the last four digits of the Credit Card Number. And I want the rest of these numbers to show in Asterisks. So in order to do this, I need to use the right Text Function and also a little bit of Custom Formatting. So I'm going to do all of my working out in column G, and then we're going to delete column G afterwards.

So to do this, all I need to do is say =“****-****-***”&RIGHT(F. And the Right Function allows us to specify how many characters from the right we want to pull out of this Credit Card Number. So I want the last four numbers. So I'm going to select the cell, the number of characters is four, close the bracket, hit Enter, and there I have the correct format. So now I can double click to copy this down. I'm going to copy and then I'm just going to paste the values over the top of the data that's already there in column F, and then delete out what I don't need.

Now the final thing I want to do here is I want to Autofit the entire table. So let's select everything, up to the Home Tab, into the Cells Group, and I want to say Autofit Column Width. And there we go. I'm ready to stop my Macro. Now where do I want my cursor to end up? Well, I'm just going to put it back in cell A1. Let's jump up to Developer and Stop Recording. So we've recorded a fairly complex Macro there with multiple different steps using all different kinds of techniques in Excel.

So what I'm going to do now is test out this Macro, see if it works. So I've just unhidden the copy of the worksheet that I took, and we're now going to test out our Macro. So let's click somewhere in this table, and I'm going to jump up to my Macros Button. There is my Macro and Fingers crossed, let's see what happens. Let's click Run and would you look at that. It has executed it perfectly, and I've saved myself so much time. Imagine if you had to do that every time you receive some data from a client or maybe a different Department, you've probably saved yourself a good 10 minutes there simply by setting it up as a Macro.

And because we've recorded this using Relative References, it doesn't matter where this table of data is located. I can run it in any part of the worksheet. So the final thing I want to show you here is just how to add a user friendly Button, which you can click on to run the Macro. And this is particularly useful if you're going to send this worksheet to somebody else who maybe isn't that familiar with how to run Macros, it makes it super simple for them to understand, and they can just execute the Macro with one click of a Button.

Now you can assign Macros to shapes. So if you want to go into Insert, and just draw a Shape and assign a Macro to that, you can. But I'm going to do mine from the Developer Ribbon. Because underneath the Insert dropdown, we have some Form Controls. And one of them is a Button. So let's click it. And I'm going to draw a Button at the top here. And you'll see when I let go, immediately, it asked me to assign my Macro to the Button. And it's going to show me any

Macros that I have in this workbook. I only have one at the moment. So let's select it and click on Ok. I'm now going to right click on the Button and edit the text and just give this Button a more meaningful name. So let's call this Reformat Customer Table. So now if I want to execute that Macro, I can simply click on my Button. So let's test it out and see if it works. And there we go. Like magic, everything has been done.

Now this is quite a complex Macro with lots of different steps, including lots of different Excel functionality. So we've probably saved ourselves about 10 or 15 minutes of fiddling around simply by creating a Macro. And if this was something I had to do daily, weekly, or on a fairly regular basis, that's going to save me a lot of time. So I hope you enjoyed that, I will see you in the next lesson.

Video: Creating a Macro to Sort a Filter Data

Deborah: Hi guys, and welcome back to the course. In this lesson, we're going to take a look at creating a couple more Macros, this time to Sort and Filter the data. And what we're going to do then is we're going to create a couple of Buttons. And I'm also going to show you how you can assign Macros to the Quick Access Toolbar to make them super easy for you to access.

Now, the data that I'm using here is just some employee data. So this might be something that you would find in a HR team. And you'll also notice at the bottom, I've already made two backup copies of my data in case this all goes completely wrong and I need to retrieve the original. Now for this example, I'm going to be using Absolute Referencing as opposed to Relative Referencing that we've been using in the last couple of lessons.

Because if you imagine, if you have another table of data, maybe a table that contains different Job Titles, different Departments, or even different Data entirely, if I have a Macro that's applying a filter to these columns, this is very specific to the data that we have in this table. So if I was then to try and run the Macro on a completely different set of data, then that's obviously not going to work. So there's no point in me using Relative Referencing, I'm going to use Absolute Referencing. So the first thing I'm going to do is just click in cell A2, where I have that first column heading. Let's jump up to the Developer Tab, and I'm going to turn off Use Relative References. Let's start recording our first Macro.

Now this first Macro is going to be a Macro that sorts my data. So I'm going to call this Sort_by_Department_Hire_Date. So this is very similar to doing a Custom Sort, we're going to sort by two columns. Now I'm not going to assign a shortcut key here, because I'm going to execute this Macro in a slightly different way. And I’m also not going to bother with a Description just to save a little bit of time. So let's click on Ok. So now I'm into the recording of my Macro.

So the first thing I want to do is I want to sort by Department. So I'm going to click in Department, up to the Data Tab, and into my Custom Sort. So from here, I'm going to sort by Department, A to Z, I'm going to add another level. And we're going to sort by the Hire Date. And I'm going to do, let's do newest to oldest, and click on Ok. Now that's all I want to do for the first Macro. So I'm going to jump back up to my Developer Tab, and stop recording this Macro. So now that I've sorted the data, I might want to apply a filter. And I'm going to apply a filter to the Job Title column. So let's jump back up to record Macro.

And this time, I'm going to call my Macro Filter_by_Job_Title. And once again, I’m gonna leave out the shortcut key and the Description and just click on Ok. And now I'm going to apply my filter. So let's click the dropdown, and I want to filter for the people that work in Finance Team. So that is, Accountants and Credit Controllers. And I think that is it. Yes, click Ok, and now I can stop my Macro.

And then a final Macro that I might want to record here is just a Macro that clears the filter from these columns. So let's record Macro again. And this time, I'm going to call it Clear_Filter. Click on Ok. Let's jump up to the Data Tab, into the Sort and Filter group, and I'm going to click the Clear Button. And then we're going to stop recording our Macro. So very quickly there, I recorded three different Macros. And what I'm going to do now is I'm going to assign a Button to each of them. So let's jump up to the Developer Tab, into that Insert dropdown, and we're going to grab our Button Form Control, and I'm going to put my Buttons at the top here.

So let's just draw a little Button like that. I can now assign a Macro to my Button. And the first one here is going to be Sort-by_Department_Hire_Date and click on Ok. Let's right click and edit the text. And we're gonna insert our second Button. So back up to Insert, grab our Button control, and we're going to draw another little Button in here. And we'll make this one a bit longer. And we're going to attach the Filter_by_Job_Title Macro to that Button, and click on Ok. Let's rename this, Filter by Job Title. And I can probably afford to make that one a bit narrower. And then our final Button in here is going to be our Clear_Filter Button. So let's draw our Button and assign our Macro, click on Ok. And this one is going to say Clear Filter.

Now it's worth noting with these Buttons, you do have some Formatting Options for these. If you right click in the Contextual Menu, you have a Format Control, where you can do things like change the color of the font, the font style, the size, so on and so forth. Now, if you click away from that Button, and then just right click on the Button and go to Format control, you'll see that you actually get a lot more Tabs. So if you're just clicked in the text, then when you right click, you're only going to see this Font Tab. So if you want to make formatting changes to the actual Button, you need to click away and then just right click on the Button to get these other Tabs at the top, where you can make some other modifications as to how that Button looks.

But we're just going to leave ours as they are, and let's test these out. So we're going to do Sort by Department and Hire Date first of all. And it looks like that I've managed to cut off the wording to Hire Date. Now this is a good opportunity for me to show you a little tip, because essentially what I want to do here is grab these two Buttons and move them along. But the problem is if you click on them, they're going to run. So what you need to do to select your Button is right click your mouse, and then you can pick it up, drag it, and move it along.

So right click your mouse and then drag your Button along. And I'm going to right click on this one. And I'm going to drag that out a little bit so we can see Hire Date, and we might as well move that one along as well. So now let's test out our Buttons. I'm going to click on the first one just here. And I should find that my list gets sorted by Department and Hire Date, which it does. So remember Department was A to Z, and Hire Date was newest to oldest. I'm now going to filter by the Job Title, and there we go.

If you remember, I filtered just for the people in the Sales Team. So that's all of the Accountants and the Credit Controller. And then I have a Clear Filter Button, which is going to allow me to clear that filter and apply something different.

So the final thing I'm going to do here is I'm going to show you how you can assign your Macros to your Quick Access Toolbar. And you'll see that running across in mine, I have quite a few Macros already assigned to this Toolbar. And this is a really nice, quick alternative way of executing those Macros. And it's a pretty simple process, all we need to do is click the dropdown to Customize the Quick Access Toolbar. And we want to go all the way down to More Commands, which is going to jump us into Excel options.

Now you'll see it's automatically taken me to the Quick Access Toolbar section. And if you take a look on the right hand side, you can see all of the things that I have added to my Quick Access Toolbar. And the majority of these after that last separator are all Macros. So what I'm going to do is I'm going to add in another separator. So let's select separator from the left-hand side and click Add to put that at the bottom. And now I'm going to add my new Macros that I've just created. So a quick way of finding all of your Macros is to change the dropdown where it says Choose Commands From to Macros.

Now this is going to show you all of the Macros available in Excel, including inbuilt Macros that Excel already has. Now most of the time, you're going to have named yours in a different format to the way that Excel names its Macros, so they're pretty easy to find. So I can see there's one of mine, I'm going to add it. There's another one, I'm going to click Add it and they are all in alphabetical order, which does make it a little bit easier to find them. And there's my final one at the bottom, I'm going to click Add. Now if I want to reorder these, I have my move up and move down arrows. So I'm going to move Sort to the top, then we're going to have a Filter, and then we're going to have Clear Filter.

Now you'll see that they also add with a rather generic looking icon. So if you want something that's more appropriate for the Macro, just click on it and click on Modify. And you can now change the name of the Macro and also assign a little icon. Now if you want to, you can come in here and change the Display Name. In general, it's not really necessary for the Quick Access Toolbar because we can't really see the Display Names anyway. But I'm gonna change this. So let's just say Sort by Dept and Hire Date so that when I hover over the icon, that's what I see, and not the longer title with all of the underscores.

Now I'm just going to select an icon from here and which one shall we use? Let's just use this down arrow and click on Ok. And I'm going to do exactly the same for Filter_by_Job_Title. So let's call this Filter by Job Title, and we'll assign something to that as well. So let's just do this little egg timer and click on Ok. And then finally, I'm going to modify the last one. So let's just say Clear Filter, and we will have the cross in the middle there for this one, and click on Ok. So now when I click my Ok Button, you'll see that I have a separator just there, and those are my three new icons. So let's test out our little Buttons that we just added to the Quick Access Toolbar.

I'm just going to run the filter again as we cleared the filter last time. So let's select our egg timer, Filter by Job Title as I hover over it, and would you look at that. And of course, we can clear that filter as well. So that's just a couple more examples of creating Macros. Hopefully, you're starting to get the idea now.

We've seen once again, how to create different Buttons and also how to assign Macros to the Quick Access Toolbar. That's it for now, I will see you in the next lesson.

Video: Protect Cells Macro

Deborah: Another really useful task that Macros can be used for is to protect cells or worksheets. And this is particularly useful if you have a particular worksheet that maybe has certain cells that you don't want people to change. So on the screen here, I have an invoice. And you can see in column A, I have some items in column B, the Quantity, column C, the Price Per Item, and then I have the Line Total. And then a Subtotal Shipping and a Grand Total at the bottom. And what you'll see for these figures in column D is that if I click on one of the cells, if you take a look up in that Formula Bar, you can see we have a formula sitting behind this cell.

Now, if you're not entirely sure what this formula is doing, it's basically saying, if A6, which is this cell, just here, is equal to blank. So if A6 has nothing in it, then return nothing. If A6 does have something in it, then perform the calculation, C6, multiplied by B6. So basically, when I copy this formula down, what it means is that if there is nothing in this cell, then I'm going to have nothing over here as opposed to some kind of error or a dash. Whereas if there is something in this cell, it's going to perform the calculation, it's going to do the Quantity multiplied by the Price Per Item. And you can see this in action if I was to add another Line Item here.

So let's just say, Webinar, let's say we did three of those this month, and they were $200. Each, you can see that it now populates that Line Total. So essentially, what I'm trying to say here is that I have formulas in this particular column. We also have some more formulas at the bottom, which are working out the Subtotal, which is just a sum calculation of everything above. The Shipping if I had some shipping, and then the Grand Total at the bottom, which is basically a sum of everything in column D.

Now it might be that I want to lock particular cells in this worksheet. So for example, if I'm sending this out, maybe I don't want anybody to be able to modify the formulas, but they can still add Line Items in. So what I've done here to make this a bit clearer is that I've marked all of what I call the input cells in orange, and then all of the calculation cells in this gray color. And the cool thing with Excel is it actually has some inbuilt cell styles, which help you kind of map this out. So if we just quickly look at those on the Home Tab, in the Styles Group, we have a Cell Styles dropdown. And you can see the formatting I've used there, I've used this one for my Input Cells. And this one for my Calculation Cells just to make it super clear.

Now I might want to go a stage further and lock the cells to prevent anyone from editing them. But I want to keep these Input Cells open so that people can add in Line Items. Now there's one thing that you need to understand when it comes to protecting cells and worksheets, because this does tend to confuse people a little bit. Now the one thing you have to remember here is that by default, all cells in Excel are locked. Now, you might be thinking to yourself, well, how can that be because you can click in any cell and start typing. So how are they locked?

Well, they are locked, but the locking doesn't come into effect until you protect the worksheet. So essentially, if I want to lock down all of the cells on this worksheet, except the ones I have highlighted in orange, I would need to unlock these cells, and then protect the worksheet. I hope that makes sense. I know it seems a little bit backwards the first time you come across protection.

Now we could do this with a series of manual steps. But of course, this is a Macros course, we're going to get a Macro to do it for us. So let's jump up to our Developer Tab. And for this one, I am going to use Relative References. Let's click Record Macro, and I'm going to give my Macro a name. So I'm going to call this Protect_Cells.

Let's also give our Macro shortcut. Let's do Ctrl + Shift + C. Let's see if that one's available and click on Ok. It looks like we're good. And we're now into the recording of our Macro. So the first thing I need to do here is I need to select the cells that I want people to be able to edit. So that's going to be all of these cells just here. We're going to right click, and we're going to go down to the Format Cells option. And you'll see here right on the end, we have a Protection Tab and you can see that there is a tick mark next to locked. So this is telling me that currently, all of the cells are locked.

So I'm going to remove the checkmark to unlock those cells and click on Ok. I'm going to click just above and then the final step I need to take here is protecting the worksheet. So let's jump up to the Review Tab. And in the Protect Group, I'm going to select Protect Sheet. Now I could choose to add a password in here. I'm not going to, I'm just going to click on. Ok. And that is pretty much it. So I'm going to go in, and I'm going to stop recording my Macro. So now what you'll see is that I can click in the cells I unlocked and I can type. But if I try and type in one of these lock cells, I'm going to get an error message.

So effectively, I've managed to lock all the cells that contain my formulas, so they can't be edited. But I've still allowed people to input Line Items into the invoice. So I'm going to backtrack out of this and remove the Protection, and then we're going to test our Macro. Okay, so this time, we're going to do something a little bit different. We've seen how to assign Macros to Buttons, we've seen how to add them to the Quick Access Toolbar. This time, I'm going to add an icon and we're going to assign the Macro to that. So let's jump up to the Insert tab and go across to our Icons Library.

Now I'm just going to do a search for something that reminds me of protection. So something like a padlock. Let's select this icon and click on Insert. Now I'm just going to place this next to my table just here. But what you'll notice is that if I right click on this, just over halfway down this contextual menu, I have an option to assign a Macro. So let's click, and I'm going to assign my Protect Cells Macro and click on Ok. And now I'm going to run this Macro to see if it works. So let's click on our padlock. It doesn't look like anything has happened, but let's see. Can I click in these orange cells and type? Yes, I can. Can I click in the gray cells and type? No, I can't.

So my Macro is working correctly. So really what I wanted to show you here was just another type of Macro that you might want to set up. But also the fact that you can actually assign Macros to things other than Buttons and Toolbars and things like that. You can assign them to shapes, pictures, and icons as well. So we've now seen quite a few different examples of Macros to automate common daily tasks. And of course, there are so many that you can create. It really does depend on the kinds of tasks that you do frequently, and what you want to automate.

But hopefully, we've seen this process enough times that you're going to be fairly confident to go in and start having a little experiment yourself with Macros. Just remember: always take a backup copy of your worksheet, so that if you need to go back to the original, you can. So now we've seen how to create and run different kinds of Macros. The final half of this course we're going to focus more on VBA. And that is the Code that kind of sits behind your Macros. So we're going to jump in, we're going to take a look at the VBA editor.

I'm going to show you how you can easily read your Code, make minor adjustments, and then we'll do a few more complicated examples as we get towards the end of the course. But for now, that's it on this first part of the course on creating Macros. I will see you over in the next lesson where we're going to start to explore VBA.

Section 3 – The VBA Editor

Video: The VBA Editor

Deborah: Hi guys, and welcome back. In the first half of this course, we've mainly been focusing on how we can create and run Macros. And now it's time to move on to focusing more on the VBA Code. And over the course of the next few lessons, we're going to fully understand the VBA Editor. I'm going to show you how you can edit Macros, and also how you can start to write your own from scratch. But first things first, let's make sure that we understand the VBA Editor.

Now, we briefly jumped into this a bit earlier on in the course. And so you should remember that you can get to the VBA Editor, either by clicking on the Developer Ribbon, and clicking on Visual Basic, or by pressing that useful keyboard shortcut, Alt + F11, and there we go. There is our VBA Window. And you'll see in here it says VBA Project [protection Macros.xlsm], which is the workbook I have open just behind this window. And I have listed here both of the sheets in that workbook. So Invoice and Sheet 2. Then just under that we have any modules that contain Macros.

So if I click on Module 1, that's basically what we're seeing in the coding window just here. So this is going to show basically all of the Macro Code I have contained in this workbook. And because I just have one Macro, which, which was that Macro we recorded to protect the sheet cells, that's the Code that I'm seeing in the main Code window in the middle. Now just underneath the VBA project window, I have a Properties window. Now one of the things I can do down here is I can rename the module.

So Module 1, Module 2, Module 3 isn't very self-explanatory, not particularly meaningful. So if I wanted to make these easier to find, I might decide to rename this and call this, let's say, Protect_Cells, something like that. And you can see that as soon as I do that, it renames the module in the Project Explorer, but it also names the window at the top here in the middle. So now it says Protect_Cells, as opposed to Module 1. So that's just an admin tip so you can keep on top of the different modules that you're creating.

Now, if we wanted to create some kind of Macro from scratch, so by that, I mean, typing up directly in the VBA Window, as opposed to recording a Macro and then editing, what we would need to do is jump up to the Insert menu. We've got a few different options in here. But if you want to create a brand new module, you would select Module, it opens up a new window.

Now it's called this one Module 1 because I renamed my Module 1 essentially, if I'd left Protect_Cells as Module 1, this new window would be called Module 2. And a bit later on in this course, we are actually going to write our own Code from scratch. So just be aware, that is your starting point, Insert and then Module.

Now I'm going to close that down for the time being, because I want to focus on this Immediate Window at the bottom. If you remember, this is a little window that I opened up earlier on in the course. And if you don't see that, just to remind you, if you jump up to the View menu, and turn on Immediate Window. Alternatively, you can press Ctrl + G, that's going to bring this little window up at the bottom. Now the Immediate Window does many things and can help you in many ways.

Now one of the things it can help you with is you can basically ask it questions, and it will give you an answer. Now if you want to ask a question, you need to start with a question mark. So basically, I'm going to ask the Immediate Window to tell me how many worksheets I have in this workbook. So all I need to do is type in worksheets. and you'll see that Excel's IntelliSense then kicks in and gives me a list of essentially arguments that I can add to this. So if I want to count the worksheets, what do you think I might apply from this little drop down menu?

Well, if you guessed Count, then you would be right. So I can double click to select, hit Enter, and it's telling me that I have two worksheets in my workbook. Let's ask it another question. And for this, I'm actually going to let's make these windows side by side, like so. It makes things a little bit easier. I'm going to click on cell C6 in my worksheet and then I'm going to ask the Immediate Window, ?active cell. So that's basically the cell that I'm currently clicked on, . and then I have my options.

Now this time I want it to tell me what number formatting I have applied to the cell that I'm currently clicked on. So if I start to type Number you can see in the IntelliSense menu, I have an option for Number Format. So let's double click and hit Enter. And it basically tells me the Number Formatting applied. If I was to do this again for a different cell, so let's click on a text cell just here. And once again, we're going to type in ?active cell.number format, and hit Enter. It's telling me that I have General Format applied to that cell. So pretty useful. What other types of questions could I ask?

Well, maybe I want it to tell me what value I have in a specific cell. So let's click on this cell just here C 10. So let's type in ? I'm going to say range, (, you can see I get my arguments underneath. And it's telling me which cell? So let's say I want to find out what is in cell C 10) hit Enter. It's telling me that I have the value 4000 in C10. And if I look at my worksheet, I can see that yes, that is true. So that is one use for the Immediate Window, there's lots of different questions that you can ask it to get an answer returned.

There's something else we can do with the Immediate Window is we can execute lines of Code. So let me look at my worksheet. And I'm going to say that I now want to center the heading label item in the cell A5. Now of course, I could do that from within Excel. But I could also execute it from the Immediate Window. So what I would say here would be selection. So it's basically going to reference the cell that I'm currently clicked on. And I'm gonna say, horizontal alignment, center, across selection. Let's hit Enter.

And if you take a look in my spreadsheet, you can see that it's now aligned, that heading to the center. I could do it for all of the headings. So if I select all of these, and then just rerun my Code, by hitting the Enter key, it's now centered all of those headings.

Now, there are, of course, many other uses for the Immediate Window. But I just wanted you to start to get an idea of some of the kinds of things that you can do from here. And if you find the Immediate Window particularly interesting, then there's a whole heap of information about it online for some further reading. But for now, we're going to shift our focus back to our Code. We're going to take a look at Macros that we've already created.

We're going to work our way through the Code line by line so you can start to understand how to read through your Code and also do things like identify errors and test out your Code. So that's all in the next lesson, I will see you then.

Video: Understanding VBA Code

Deborah: So before we start creating our own Macros and our own Code, it's a good idea to take a look at how Macros actually build VBA Code when you run them. So what we're going to do in this lesson is we're going to run a Macro on this worksheet, and we're going to keep the VBA Window open as we do it so that you can see how the Code is built. And then I'm going to show you how simple it is to make some minor modifications. So let's start out by taking a look at the data that we're using.

So on this worksheet, I just have some UK Software Sale Figures for 2019 and 2020. And I've got some subtotals applied to these figures. So you can see at the bottom of 2019, we have the total just here. And then at the bottom, we have the 2020 total, followed by the grand total, which is basically just a calculation of the 2020 and 2019 figures. Now this might be some kind of report that I need to send out to a client or maybe a different department. But it might be that once I've sent this out, they come back to me and say, “You know what, actually, the layout or the font that you're using in this report isn't what we want, we want you to change the font. And we also want all of the headings and the subtotals to show in bold”.

So I'm going to record a Macro, which basically does that. And I'm going to record it with the VBA Editor window open so you can see what happens. So let's press Alt + F11, to bring up the VBA Editor. And what I'm going to do is just place these windows side by side, so we can see what we're doing. So if we look at the VBA Window for a moment, you can see at the top here in the project window, it says that we're in the change font worksheet, which we are that's the name of my current workbook. And I'm currently clicked on Sheet 1, which is the sales sheet. And one other interesting thing to note here is that I don't have a folder that says modules.

If you remember, once you've recorded any Macros, then you will have a Modules Folder. And you can access your Macro Code from within that. But currently, because there are no Macros in this spreadsheet, I don't have that folder. But you'll see that as soon as I start to record one, that folder is going to appear. So what I'm going to do with this worksheet is we're going to record a Macro that basically changes all of the font to a different style, and also bolds these headings, subtotals, and total.

So let's jump up to the Developer Tab. And you'll see that because I now have my window smaller, it's only taking up half the screen, my groups of commands are now squished up. So if you take a look here, this one at the top is my Macro Recorder. Now for this, I'm going to turn off Relative Referencing, let's click on Record Macro and give our Macro a name. Now I'm going to call this Change Font. And I'm going to change this to a font called Segoe UI. So I'm going to add that to the title so I remember what this one is.

I'm going to assign a shortcut key of Ctrl + Shift, and let's do U this time, and I'm going to store it in this workbook. So let's click on Ok. Now you can see I've got a yellow warning message pop up there straightaway that simply because I don't have this saved as a Macro-enabled workbook as yet. So we'll do that after we've recorded the Macro. So don't worry too much about that yellow warning message just yet.

The other thing to know is that we now have a Modules Folder. And that literally appeared as soon as I started recording. So let's run our Macro. I'm going to select the entire worksheet by clicking on this cell in the top corner between A and 1 to select everything in the worksheet, we're going to go up to Home, into Font, and we're going to change our font from Calibri. And we're going to change this to Segoe UI. So that's going to be towards the end of the list. And it's this one here. And then I'm going to change the headings to bold.

So let's click just here, and select up to Font and Bold. Let's scroll down and change our subtotals to Bold. And then the final two here are also going to be Bold. And let's just widen that column out. And then we're going to stop recording. Now before we take a look at this Macro Code, I'm very quickly going to save this workbook as a Macro-enabled file. And there we go, that warning message has now gone.

So let's go over to our Modules Folder and click the plus to expand it. And you can see we now have Module 1. So if I double click that's going to open the VBA Code for the Macro that we've just recorded. So let's walk through this Code and start to understand what we're looking at.

Now with any Macro, you'll always find that the Code starts with Sub, and then the name of the Macro, and also ends with Sub as well. And sub stands for Sub procedure. Now at this stage, you don't have to worry too much about what exactly that means, just know that these kind of act as bookends to your Code.

So you'll always have Sub and then the name, and then End Sub at the bottom. What you see underneath at the top is some text that's in green. And it also has an apostrophe at the start. Now anything you see in this format, so has an apostrophe, and is also in green. Those are comments. So if you want to add some useful instructions into your Macro, or leave a note or a comment for someone who's looking at this Macro Code, then you can do that simply by putting apostrophe in front of what you want to say. And what it means is that when you run this Macro VBA isn't going to execute that line of Code.

So if I wanted to, I could add something up here, put an apostrophe in the front and just say, “This is for the sales team”, something like that. And you can see, as soon as I click away, that turns to green. So that's for any comments or any notes, we then have the start of our Code. So let's start to get a feel of the format. And this Code basically walks through all the steps that the Macro took. So you can see here it says cells.select. So basically, when I clicked to select all of the cells in the entire worksheet, that's the line of Code that it produced. It then says Selection.Font Name=Segoe UI. And that's the font that I changed my worksheet to.

Then what you'll see is we have quite a few lines of Code related to different font properties. And you can see that the majority of these say false next to them, which means that they're not going to apply. So it says here, .strikethrough = false, .superscript = false, so on and so forth. And this is what you'll find with a lot of Macros that you record using the Macro Recorder. Instead of it just saying, okay, I'm going to change the font to Segoe UI, it basically goes through a whole bunch of other properties, and says, nope, don't want to turn that one on, nope, don't want to turn that one on, so on, and so forth.

So effectively, all of this Code, apart from where it changes the font selection to Segoe UI, isn't really needed. So we could effectively delete this out, and the Macro will still work. And that's exactly what we're going to do in a moment. And then the final part of this Code is related to really the second kind of step that I took in the Macro. So this is where I started to make all of the headings bold. So you can see it says Range [“A6:E6”] .Select bold = true. So it's looking at A6 across to E6, and it's changing it to bold.

We then have active window, scroll down. So that's where I scroll down the page, and then Range (“A19:E19”) .Select. So that's where we're talking about this row just here. We then have the same thing, we're changing it to bold. We’re scrolling down, and then selecting A 32 to E 33. So both of these rows down here, and we're changing it to bold, bold = true. And then finally, if you remember right at the end, I auto fit column E to accommodate that newly expanded grand total. And that's the final line of Code just here, column E, entire column auto fit. So without really knowing a great deal about VBA Code, I fairly easily been able to follow the process that that Macro has taken.

So I'm going to click my mouse right at the end just here. And we're going to create another Macro, I'm going to leave this window open so you can see it building. So maybe now I've sent this to a completely different department. And they've said we prefer a different style of font altogether. So let's create a new Macro. We're going to click on Record, and I'm going to call this change_font_Lato Let's assign a keyboard shortcut of Ctrl + Shift + L and click on Ok. And you can see over in my Code window, it's now put in the Sub, the End Sub, and some comments at the top.

So I'm going to go through the same process. Let's select the entire worksheet, you can see the Code building, cell select, up to Home, into Font. And now I'm going to select Lato font. As soon as I did that, a whole bunch of Code was put into the Code window. I'm now going to go to Developer and click on Stop Macro. So now if we look at our Code, we have basically a very similar thing. The only thing we don't have for this one is basically this section of Code, because we already have bold applied.

So once again, we're selecting our cells, we're changing this to font Lato, and then the rest of this Code isn't really need it. So I'm going to go in, first of all, and I'm just going to comment out the lines of Code that I don't want to run. So I only really want this first part to run up here. So everything from there down, I'm going to put an apostrophe in front of. So now what I want to do is I want to test out my Macro to make sure that it still works now that I've commented out these lines of Code. Now if you want to test or run a Macro, all you need to do is click at the top of the Macro where you have the word Sub, click just before it and then if you look at your Toolbar, you have a little play button just here. And this basically allows you to run sections of your Code. So this is going to run this entire sub. So all of this Code all the way down to End sub.

So let's see if our Macro still works. And there we go, it does. So I know that deleting these lines of Code is not going to be a problem. So let's jump in, and let's delete this out. And then what I can do, just to tidy this up a little bit is I don't really need .name on a second line, I can just cut Ctrl + X and paste this next to selection font. Now, something I also don't need in here now that

I've deleted all of those properties is this statement that says With and then End with. So that's unnecessary, as well. So I'm just going to delete that out. And let's run our Code to make sure that it still works. And it does. So let's run the same process on a Macro to change the font to Lato.

So once again, I don't need most of this in here, I just need where it says selection font name = Lato. So I'm going to select all of this and press Delete. And I'm going to move this and I'm just going to drag and drop it this time to the end, and delete out the End with and also the With. Now of course, you can tidy this up by removing some of these blank rows just by deleting them out, make it look a little bit neater. But let's now run this and see if it still works, which it does.

Let's run our one at the top again, just to double check. And yes, we are all looking good. So there we go. We've now seen how the Macro Recorder records Code. We've read through our Code to understand it. We've learned about Sub and Ends and bookmarks and how we can enter in comments so that certain lines of Code don't run. And we've also seen how we can identify Code that's unnecessary, remove it safely, and then test our Macros by rerunning the Sub procedure. In the next lesson, we're going to do a bit more on editing Macros. So I'm going to jump over there now and I look forward to you joining me.

Video: Editing Macros

Deborah: In the previous module, we saw how we can build a couple of Macros. We've seen them in the VBA Editor, and I've just talked you through some of the Code and how you can make minor modifications, and then test out that Code or rerun the Sub procedure. So we're just going to follow up on that a little bit, just so you can start to get a bit more of a feel about the types of things you can change.

Now, we're still working in the same workbook, and the same Macros. So we have our first Macro at the top that changes the font to Segoe UI, and also changes those headings and subtotals to bold. And then we have our Macro at the bottom that changes the font to Lato. Now, one thing you'll notice is that at the end of each of these Macros, we have this line of Code Range (“A4”) .Select. We have it there, and we also have it down here.

So that is basically the last line of Code that the Macro executes prior to exiting the Macro. And all that says is okay, once you've finished running the Macro, where do we want the cursor to end up to be, which cells do we want selected? Cell A4. And you will have noticed that when I run these Macros, my cursor ends up in cell A4. So when we're thinking about making minor changes, this is something that's fairly simple to change.

So if I decided that instead of cell A4, I wanted my cursor to end up in cell F1, that's a very simple thing that we can change in this Code. So I'm going to click down here, and I'm going to edit out A4, and replace it with F1. And I'm going to do the same thing over here. Let's run our first Macro and make sure that that still works. So we're going to click at the top and click on play. And there we go, my cursor is in cell F1. Let's click somewhere else before we run the second one. Click and press play, my Macro executes and my cursor ends back up in cell F1. So, so far, that is working perfectly. And you can probably see that based on that if we wanted to make different range selections.

So maybe I wanted to change something else to bold, I could just edit the cell references throughout the Code. So reasonably straightforward to understand. So let's take a look at something slightly different. Maybe when I run my second Macro down here, the change font Lato Macro, maybe instead of just changing it to Lato font, I want it to change the headings, subtotals, grand totals to Italic instead. So what I could do to make my life a lot easier is instead of rerunning the Macro, or writing out lots of lines of Code, I can simply copy and paste what I have in the Macro above and then modify it.

Because if you take a look at this Code, it says Selection.Font.Bold = True. So based on that logic, if I wanted this Macro to change it to italics, all I would need to do would be to have the line of Code Selection.Font.Italic = true. So let's see if that works. Now I'm going to copy this Code down to here, I can see that I don't need this line, Range (“F1”), because otherwise it will be repeated. So I'm going to select everything above, up until there, Ctrl + C to copy. I'm going to come down to here. And let's just press enter, and Ctrl + V to paste.

So now what I can do is wherever I have the word Bold, I'm just going to change that to Italic. I'm going to double click, copy it, and we're going to replace bold just here as well, paste. And we have it one more time just here. Paste. So now my Lato Macro says Italic = true. So let's see what happens. I'm going to run the first Macro note, we're already on that. So that's fine. And you can see that this Macro has changed those headings to bold as it's directed to do. So now let's run our second Macro and see what happens. There we go.

It has changed those headings to italic. But it's also kept them in bold. Now it's kept them in bold because I ran the Macro on my spreadsheet that already had bold in the headings. So nothing in this Code tells VBA to turn off bold. It just says make whatever's in this cell Italic. So what about if I did want to turn off bold as well, what would I have to do here? Well, again, it's reasonably straightforward. What I could do is I could take this line of Code, copy it, and change it to say Selection.Font.Bold = False. And that is essentially going to turn off bold. That's what these true or false toggles do.

They say, okay, turn this on. No, turn that off. So what I'm going to do here is I'm going to Ctrl + C and Ctrl + V to paste that line in. And we're going to change Italic to bold, but we're going to say false. So what I should see in the heading here is I should see it still in Italics, but not in Bold. Now I'm going to do the same thing for these other items that have bold. So let's copy this line of Code, and go down to where we next have Selection.Font = Italic, and paste that line in. And we have one more down here and paste that line in. So now I'm going to run the first Macro again, just to get it back to what it was previously. And now let's run our second Macro.

So remember, what we're watching for here is we want the titles to stay in italic, but we don't want bold on them. So let's click on play. And there we go, we have exactly what we're looking for. And our cursor is back in cell F1.

So hopefully now you're kind of getting the idea as to how you can select a cell or different ranges of cells using VBA Code, and then how you can make some adjustments to things like the font style and its properties. So let's finish off this lesson just by adding in a couple of others that work along the same lines. So maybe when I ran the Lato Macro, I also wanted to change the background fill, and the font color of this heading row.

So we have all of the information that we need to do this. So I'm going to go down to the bottom, and I'm going to put this in right at the end just before it does the final selection of cell F1. So we want to start off by typing range. Because we want to select a range first of all. So we're going to open a bracket. And our range selection, which we put in quote marks is again going to be ("A6:E6"). And we want to select that range, so .Select, and when it comes up in the IntelliSense menu, you can just press the Tab key to have it autocomplete.

So we've selected our range. So now what do we want it to do once it's selected that cell range? .well, I want it to change the interior.color index to five, and five equals blue. Now you can easily look this information up online so that you understand which color index relates to actual colors. But for the time being, just know that five equals blue. What do I want to do next? Well, then I also want to change the range A6 to E6. And I want to change the font color again, I can press tab to get that to autofill to VB. Yellow. So now that we've added those two lines of Code, let's rerun our Macro and see if it works. So I'm going to click at the top and press play. And would you look at that, we now have a heading with a blue background and yellow font.

So hopefully I've demonstrated there how simple it is to make some minor changes to your Macro, and also how you can start to type in basic lines of Code. And I will say that Google is a great help when you're learning VBA if there's something that you want to do. In general, if I'm not sure what the Code is, I just type into Google, change background color VBA and you'll find the line of Code that you need. And the more you start practicing typing your own Code, the more methods and actions that you're going to learn. But for the time being, I think that's more than enough for now. I will see you in the next lesson.

Video: Debugging Macros

Deborah: Hi guys, and welcome back. In this lesson, we're going to start to take a look at Debugging. And all Debugging really is is working our way through our Code and identifying errors. So you're more than likely go through the Debugging process for the first time when you run a Macro, and it goes a little bit wrong. It's a great way of identifying where exactly in your Code the error lies, so that you can then jump in there and just fix it up. So let's take a look at an example.

Now currently, I'm clicked on this worksheet at the bottom, that's called Fixed. And this is essentially what I want my worksheet to look like once I've run my Macro. So I have a table of data here, just some employee information, we have a date column that has all of the dates in it formatted with the short date format, and a salary column on the end, we also have a title at the top. And this is basically what I want my worksheet to look like once I finish running the Macro.

Now I've already created a Macro that's going to do all of this for me. This is a reasonably straightforward Macro. So this worksheet here is what I want the worksheet to look like once the Macro has run. But let's take a look at what it was like originally. So you can see I still have my data in the table. But I don't have any formatting applied to column F. And I also don't have my heading up here in cell A1. So what I've done is I've created a Macro that when I run it, it should make my spreadsheet look like this. So what we're going to do is we're going to run that Macro and see what happens.

So let's jump up to the Developer Tab into Macros. There's my Macros sitting there, it's called Format_Col_Currency_Title, let's run it and take a look. So I can see that I have a couple of issues here, because this hasn't worked out quite as I was hoping. So if I take a look at my salary column, I can see that it has applied a currency symbol, but it's applied the wrong one. So clearly, I've accidentally selected euros as opposed to dollars. And you can also see that I've accidentally clicked in a wrong cell when I've typed in my heading. So maybe I didn't realize I was clicked there when I was doing it.

So now let's jump into the Macro Code and start taking a look at our Debugging tools, which are going to help us work out what we need to change here. So I'm going to press ALT + F11, to pull up the VBA Editor. So let's do a couple of admin things here first of all. I'm in my Debugging worksheets, there's my Modules Folder, there's my Module 1, which is showing my Macro. And I'm going to rename this module to make it easy to identify. So let's call it the same thing, Format_Col_Currency_Title, and hit Enter. So let's take a look at our Code.

Now I'm going to work my way through this Code. And this Code isn't particularly long. And I will say that Debugging tools are a lot more useful if you have a lot of Code to work through. But let me show you how you can turn them on. Now there's a few different ways and it depends on what tools you find useful because the Debugging Toolbar has quite a few utilities on it. But I tend to find that I mainly use one or two of them.

So what you could do to turn the Debugging Toolbar on is right click your mouse in a blank space up here on the Toolbar and you have a Debug option. And it brings up a floating Toolbar with all of your Debugging tools, which you could then drag and drop and make part of your Toolbar set up here. Now I find that a bit much for what I want to do, and a lot of these are already on this Toolbar.

So what I prefer to do is right click my mouse and go down to Customize, click on the Commands Tab, and then select Debug from that categories list. That then shows you all of the different commands that fall under that Debug Category. And you can choose which ones you want to drag up to that Toolbar.

Now the two that I use most often is Step Into and Compile Project. But I can see that I do already have Step Into on my Toolbar, so I'm just going to grab Compile Project, drag and drop to place it on the Toolbar. Let's click at the top of the Macro. And I'm going to use my Step Into button, you can see there's a shortcut for that, which is F8. And basically this is just going to go line by line through my Code, highlighting in the worksheet where that Code is being applied. So this really helps you identify exactly where in your Code things start to go wrong, making it a lot easier for you to fix any issues.

So currently, we just clicked on the title of the Macros. So let's click Step Into again to move down. And the first one it goes to is Range (“F5”) .Select. Now, you'll see in the spreadsheet it's currently still selecting cell A9 and A10. That's because you have to click again and then it executes the one before essentially. So now if you look at the spreadsheet, my cursor is clicked in cell F5.

The next line of Code it's gonna execute is going to make a cell selection, and the reason why it says Selection .End(xlDown) is because to select that whole range, I did Ctrl + Shift Down Arrow shortcuts. Once it's made the selection is going to change the number format to euros.

So let's click again, and you can see it execute that. So it's highlighting all of the cells, it’s changed them into euros, and now it selects range A9. So we're over here now in column A, and then it says active cell, and then Global Sales 2020, which is what I typed into that cell. It then select cell A10, and types in Micro World, it then selects both cells A9 and A10. It changes the font to 14. And the reason why it has 12 and 14 listed there in the Code is because I use the increase font icon. So I clicked it twice, it was on 10. Clicked it once to go up to 12, once to go up to 14, hence why we have both of those clicks showing in here.

And then finally, it makes that selection bold. And then it ends the Sub procedure. So if I carry on going back through, I'm now going to correct all of these errors, and then we're going to rerun the Macro. So let's carry on going through. The first error I want to fix is where it changes that salary to euros because I want to change it to currency format. So let's go through. And it's this line of Code that I want to deal with, because this is where it makes a selection and then changes it to that euro format.

Now this is where I might possibly use the Immediate Window before I actually change the Code in the main window. So I know that I'm going to need Selection .NumberFormat = it’s just a question of what needs to go within these quote marks. So I'm going to need the currency formatting, and I want to format it with two decimal places at the end. So this is where you have to kind of know a little bit more about Excel custom formatting in order to be able to write the correct format. So I want a dollar sign in here, and then I want to have the number.

So I need to put in a #, I then want it to be , in the middle there, ##, I want a 0 on the end, and then .00. And put that in quote marks. So what I'm going to do here is I'm going to test this out and see if it works. So let's hit Enter. And I can see that that works perfectly. So I know that I can now safely replace the Code that's currently highlighted with this new piece of Code.

So I'm going to cut it from this Immediate Window, jump up to here, and then we're going to do a little bit of editing. So let's highlight all of this and delete. And we might as well delete the whole thing here because we've got the whole thing copied. So let's do Ctrl + V to paste that in and just delete that blank line in there. Okay, perfect.

Let's now carry on stepping through our Code. So the next thing we need to change is where we're typing in the word Global Sales 2020. And you can see just above it says Range (“A9”) .Select. So it's selected that cell and then it's typed in Global Sales. Now I don't want it in A9, I actually want this in A1. So I can quickly change that to A1. Let's carry on stepping through our Code. It then goes to cell A10 to type in the word Micro World. And I want that to be in A2. So again, I can very quickly just change that to A2.

And then finally at the end here, it does Range (“A9:A10”). So it selects both words, and it changes them to bold. So I need to change these cell references to the new ones, which is going to be A1, and A2. So now I should have been able to very quickly fix those small errors. So now I've been able to fix those errors, we're going to rerun the whole Macro on our original data. So let's jump back to original. And we're going to click at the top of the Macro. And we're going to press our little play button to run the Sub procedure. And there you go.

Would you look at that? Our salary column has changed to the correct format, and we now have those titles in the correct area. So those Debugging tools particularly that Step Into icon, just executes your Code line by line so that if you are having an issue, you can identify exactly where it is and make the necessary changes. That's it for now, I will see you in the next lesson.

Video: Creating A Macro From Scratch

Deborah: Hi everyone, and welcome back. It's now time to create our own module from scratch. And we're going to keep things fairly simple just until you get the hang of this. We're just going to create a Macro or write some Code that's going to insert a row at the top of this worksheet, and then add column headings to this data.

So I want one column that says First Name, one that says Last Name, Department, and then Job Title. And then I want to just format those headings with bold. So let's take a look at how we would do this if we were writing this from scratch in the VBA Editor. So I'm going to press Alt + F11, to jump into the VBA Editor. And the first thing we're going to do is we're going to create a brand new module.

Now, I did show this in one of the previous lessons, but let's just remind ourselves. We need to go up to the Toolbar and click on Insert. And then from the dropdown, we're going to select module. And you can now see here under the Modules Folder, it says Module 2, so let's give this a better name. So we're going to say Format_Cells and hit Enter. So now we have a blank module, we need to first add some structure into it. Because if you remember, all of our Macros started with that Sub and the End sub, our bookends.

So once again, we're going to jump up to Insert, and we're going to say we want to insert a Procedure, we need to give our Procedure a name. So I'm going to call this Format_Cells to keep it consistent, and then we can select the type of Procedure that we want to insert. Now for us, that's going to be Sub, and it's going to be a Public Sub. But there are other types of Procedures that you can add in here. For example, you can create your own functions in Excel. So we're all used to things like the Sum Function, If, Count If, things like that. If there is something that you need to do in Excel, but there isn't a function available to do it, from here, you could select function and then write your own function in VBA.

And obviously, that's much more of an advanced level than we're at now. But it's always good information to bear in mind as your knowledge of VBA progress, it's. So we're going to select Sub and click on Ok. And you can see as soon as I do that, it puts those bookends in for me. So let's press enter a few times to give ourselves a little bit more room. And we're going to start to compile our VBA Code. So the first thing I want to execute here is basically I want to insert a new row at the top of this worksheet. So I'm going to type in Rows. And I now need to specify a row index.

So I now need to insert my row index which needs to go in quote marks. And the first row that I select is row one. And that's the format for typing that out. I then need to assign an action to this selection. So at the moment, I'm selecting row one. And what I want to do is I want to insert a row. So once I've got my row inserted at the top, what do I want to do? Well, I want to select the cell A1 and add the first heading into that new row. So I'm going to type in Range, open my bracket, I'm going to put the cell reference inside quote marks, and our action here is value.

I remember to press the Tab key to select from the IntelliSense menu, and the value we want to enter into that cell equals First_Name, which is what I want my heading to say. So just to recap, we've inserted a row at the top, we've selected the cell A1, and we've typed in the word First_Name. Now I'm basically going to do the same for the rest. So we're going to say Range B1 this time, I'm going to enter into that one Last_Name.

Now of course, to make this a little bit quicker, I could copy and paste a couple of times because we have four columns, and then just modify the cell references and the names. So let's say C1, and I want C1 to say Department. And then D1, and I want this one to say Job_Title. Now once I have those headings in row one, the final thing I want to do here is I want to basically make that entire row bold. So we're going to use rows again. And we're going to type the same thing. We're going to select row one. And then we're going to say we want the font to be bold, and that needs to equal true.

And then just to finish this off, we're going to say Range, we're going to do (“A1”) .Select because that's where I want my cursor to end up. So that's pretty much it. That's pretty much all of the Code. Now of course, you could get a lot more complicated than this, I could add in some other formatting into here. So if I did want to go a little bit more complex, I could add something else into here. So maybe I want to have all of the font in that first row in a different color. So I could essentially kind of repeat what we have above, because we're just selecting that row at the top. But this time, we want to say Font.Color = and then we can enter in the color that we want.

Now, I've just pasted in there a RGB Code that I had saved off, because each color in the spectrum has its own Code. And if you want to find out all of those that is one for Google, you can go in there you can find specific Codes for specific colors, and just copy and paste whichever one you want to use. So that is my Macro, reasonably straightforward. So what we're going to do now is test this Macro out.

So let's click at the top, I'm going to click on our play button to run through this. And let's see if we get any errors. And you can see that we don't, that has worked perfectly. It's inserted that row, it's inserted those column headings, it's made them bold, and it's also changed the font color.

So hopefully you can see there how even if you don't know a great deal about VBA Code, it is quite simple to start out writing your own modules. And practicing this type of thing is going to give you a really good foundation for working your way up to writing more complex Code because there is so much to VBA and we're barely scratching the surface here. But for now, that is it. I will see you in the next lesson.

Section 4 – Creating A Macro Custom Ribbon

Video: Creating A Macro Custom Ribbon

Deborah: Hello everyone, and welcome to the final module of this short course. In this lesson, I'm going to show you how you can create your own Custom Ribbon in Excel to house all of your Macros. So, much like we have the Home, Insert, Draw Ribbons running across the top, you can also make your own Custom Ribbons and assign commands to that ribbon. So we're going to make a Macros ribbon and assign all of the Macros that I have in this workbook to that particular ribbon.

So the first thing we need to do here is jump up to the File tab. And we need to head down into this guy down here, Options. And the category that we want to go to is Customize Ribbon. Now on the right hand side here, this is where you'll see a list of all of the ribbons that you currently have in your worksheet. So we have Home, Insert, Draw, Page Layout, Formulas, so on and so forth. And you can see right at the bottom, I do actually have a ribbon called My Macros. But I've just turned that off for the time being because we're going to create another one from scratch.

So, these ribbons are also known as Tabs. And if you look just under these items, you have a button here that says New Tab. So since I click that, it's entered in a New Tab and then a New Group. Now the difference between these two is that the Tab refers to the Tab Name. So Home, Insert, Draw, Page Layout. Whereas the Group is the Group Name essentially. So I'm currently clicked on the Draw Tab. If you look at Excel behind this window, this first Group is the Input Mode Group.

So, what you can essentially do is create a Tab, name it whatever you like, and then create various different Groups from which to house your commands. So I'm going to click where it says New Tab. And I don't particularly like the position where this New Tab has inserted. So at the moment, it's going to kind of be in the middle of my current ribbons. And really, because this is a custom tab, it has Macros that I don't use every single day, I want to move this tab down to the end of the list.

So, I'm going to select it and use my arrows on the right hand side just to move that all the way down to the bottom. And I'm now going to rename this Tab and I'm going to call it My Initials, and Macros whoops, spell that correctly. And click on Ok. So now I need to enter a name for my Group. And I will say that you do have to use group names. Otherwise, Excel won't let you add any commands to the ribbon.

So I'm going to rename this group, I'm going to call it Formatting. I'm going to choose a little icon for that. Let's choose this one, just here and click on Ok. And I'm going to add another group, and I'm going to rename that one. And we're going to call this group Clean Data. And I'm just going to select one of these icons like that.

So now I have two groups, I can now add commands to those Groups. Now because what I want to add a Macros if I look at the menu on the left hand side, I need to change this dropdown at the top here to Macros in order to see all of them that I've got in this workbook. Now one thing to note here is that Excel does have its own Macros, and that's the majority of these that you can see in here. Fortunately, the Macros that you create, you've probably given them a meaningful name, which makes them stand out from the others and makes them a little bit easier to see.

So, what I'm going to do is I'm going to go through and I'm going to start adding my Macros to my groups. So the one at the top here, Accounting Format, I'm going to click on Add, Calc_Percentage is going to come under Formatting as well, so is Data Bars. But the next one, Delete Blank Rows, I want to put that in the Clean Data Group. So let's make sure we've got that selected. And then we click on Add. Format as Table is also going to go in the Clean Data Group. Highlight Cells, I'm going to put in Formatting, New Worksheet is going to go in Formatting, as is Paste Values, and also Remove Hyperlink.

So now, I have all of my Macros organized into different groups on my new Customize Ribbon. What you'll notice here is that they currently all have the same icon, and they also are taking on the name of the Macro. Now that is absolutely fine. But most of the time, what I tend to do is go in and just rename them and just tidy them up a little bit. So remove things like the underscore so they're a bit more user friendly. So I'll show you how I do the first two and then you can go through and do the rest yourself. So let's click on Accounting Format and click on rename. I'm going to assign it a new icon. So let's say this one just here. And all I'm going to do is remove the underscore from the display name. So it looks like that and click on Ok.

Let's do the next one, rename. I'm going to select the calculator icon. And then I'm going to rename this. Let's call it Calc, and then we'll just do the percentage sign and click on Ok. One more Data Bars, let's choose this icon. And for this one, I'm just going to remove that underscore and click on Ok. And then what I'm going to do is go through and do all of the rest of these. And there we go. All of those now have an icon and they've been renamed. So let's click on Ok, and take a look at the result.

Well, there we go. I can see that I have my new Ribbon Tab at the top. And if I click on it, there's all of my Macros organized into those two groups. So the way that I use this in general is if I have Macros that I use daily, weekly, monthly, I like to have them all together on their own ribbon so that they're easy for me to execute. But as far as adding them, that is the process and I'll leave you to have a play around with those.

So that is pretty much the end of this course. I'm going to say my goodbyes and just close things off in the last module. So I'm going to head over there now and I look forward to you joining me.

Section 5 – Course Close

Video: Course Close

Deborah: Congratulations guys for making it to the end of this short course on Macros and VBA in Excel. We've covered some pretty intense topics. So, if you've been able to follow along and understand, then you should give yourself a really big pat on the back.

Now in this course, we've seen how to run Macros, create Macros, and save our workbooks as Macro-enabled workbooks, and I've shown you how to add Macros to the Quick Access Toolbar. Then in the second part of the course, we moved into exploring VBA. We looked at the VBA interface and the different windows and the Editor, we made basic edits to Macros, and we also looked at how to use Debugging Tools to investigate any issues. And finally, we created our own Macro from scratch.

Now VBA is such a huge topic and we have barely scratched the surface in this mini course. But if you came away from this course with motivation to think about the work that you do on a daily basis, and maybe think about ways that you can automate those processes using Macros, then my work here is done.

I really hope you've enjoyed participating in this course as much as I've enjoyed putting it together for you. And we very much hope to see you again very soon. My name is Deborah Ashby, good bye for now.