Excel VBA Programming

Home & Learn

Excel Dot Notation

Excel VBA uses dot notation to separate the various things you can access and manipulate with the programming language. Dot notation is hierarchical, and usually starts with an object. (In Excel, an object is the thing you're trying to manipulate, such as a worksheet.) After the object, you type a dot. You then specify what you want to do with this object, or what you want to manipulate. The doing is called a method. The manipulating is done via properties or parameters.

If all this is confusing, let's try and clear it up.

Think of a television. This is an object. We can notate it like this:

tv

OK, all very simple so far. But you'll need some more information if you were going to buy a television. One thing you may want to know is how big the tv is. To add a size property, you'd do this:

tv.size

You'd want this to equal something, though, so add an equal sign and a size:

tv.size = "55 inch"

We now have an object (the tv) and a property (the size). We also have a value for the size (55 inch).

If we wanted to buy this tv then we'd be doing something (buying). We can call this "doing" a method. It is a method of the tv:

tv.buy

Methods can come with extra settings, called parameters. A parameter of the buy method could be PaymentType. The PaymentType would then come with its own values (credit card, cash, cheques, etc). We could represent all this as follows:

tv.buy PaymentType:=Cash

We have a space between the method (buy) and the parameter (PaymentType). The value for the parameter comes after a colon and equal sign (:=), with no spaces in between.

We could add more parameters for buy. For example, we could have a Discount parameter, and a DeliveryCharge parameter:

tv.buy PaymentType:=Cash Discount:=No DeliveryCharge:=No

Notice that we've used a space to separate the three parameters and their values.

So, back to Excel. In the VBA programming language, you'll uses these object, methods, properties and parameters a lot. As an example, there's an object called ActiveCell. This is the cell where your cursor currently is. The ActiveCell object can have a font set for it:

ActiveCell.Font

Fonts have a name property all of their own. So after another dot, you type the Name property:

ActiveCell.Font.Name

Because it's a property, you need a value for it. Let's add the name of a font:

ActiveCell.Font.Name = "Times New Roman"

This sets "Times New Roman" to be the font of the ActiveCell.

We can also set a bold value for the Font:

ActiveCell.Font.Bold = True

Again, we have an object called ActiveCell. This is followed by the Font property. The Font property has a bold property of its own. This is set to True.

Methods

An example of an object method is **Quit**:

Application.Quit

To Quit is to do something, which is why it's a method rather than a property. If you're not sure whether something is a method or a property, try putting the word "to" before it. To Quit makes sense, but to Font doesn't. So Font would not be a method.

Another example of a method is **Add**. You can use this to add a new worksheet to your Excel workbook:

Worksheets.Add After:=Worksheets(1)

The object above is **Worksheets**. The **Add** method comes after a dot. Next, we have a space. One parameter of the Add method is called **After**. The value for **After** is the name of a worksheet. (Inside the round brackets you can have either the name of a worksheet in double quotes, or a number. The number 1 means worksheet 1 in the current workbook.)

The Range Property

Quite a lot of the macros you'll write will need to reference the **Range** property. But what is a Range?

A Range just means a range of cells on a worksheet. A Range can be an individual cell, or a group of cells. The Range you want to refer to goes between round brackets. Inside the round brackets you surround your Range of cells with double quotes. Here's a Range object that just refers to the cell A1:

Range("A1")

And here's a Range property that refers to a group of cells:

Range("A1:B7")

Notice the semicolon separating the two cell references, and no spaces between the two. The first cell reference is the top left cell, while the second cell reference is the bottom right cell in your selection - a square, in other words.

Another way to refer to more than one cell is to separate the two cell references with a comma. Each cell is then surrounded with double quotes. Like this:

Range("A1", "B7")

The above Range property refers to the cells A1 to B7.

Once you have a Range you can do something with it. One method you can use with Ranges is the **Select** method. As its name suggest, the method Selects a Range of cells:

Range("A1").Select

Range("A1:B7").Select

The Worksheets Object

We mentioned above that Range is a property. But what is it a property of? Well, Range is a property of the **Worksheets** object. If we wanted to be really specific we should have used the Worksheets object first, then the Range property. Like this:

Worksheets("Sheet1").Range("A1").Select

Here, the Worksheet comes first. In between round brackets and quote marks you type the name of a worksheet, "Sheet1" in this case. You can also type a number between the round brackets:

Worksheets(1).Range("A1").Select

The 1 above refers to the first worksheet in your open spreadsheet.

When you use Range by itself, Excel takes it mean the currently active worksheet, the one you currently have selected and displayed. If you want to refer to a different worksheet, say Sheet 2, you need to refer to it by name:

Worksheets(2).Range("A1").Select

Or

Worksheets("Sheet2").Range("A1").Select

In the next part, we'll explore the VBA Developer toolbar in more depth.

The VBA Developer Toolbar in Excel >

Lots more free online course here on our main Home and Learn site