Advanced Excel Essentials

Author(s) Goldmeier, Jordan

Imprint Apress, 2014

ISBN 9781484207345, 9781484207352

Permalink https://books.scholarsportal.info/uri/ebooks/ebooks3/springer/2015-03-04/1/9781484207345

Pages 3 to 9

CHAPTER 1

Introduction to Advanced Excel Essentials

I set out to write a book on the *essentials* of Excel development—that is, a book that concisely presents many of the development principles and practices I've discovered through my work and consulting experience.

But whether on purpose or by accident, this book has become something considerably more than that. Indeed, another name for this book could be *A Contrarian's Guide To Excel Development*. You see, this book will push back against the wisdom of other terrific Excel books, including my favorite book, *Professional Excel Development* (Addison-Wesley 2005). To be sure, the information in those books is terrific, and whatever merits this book might achieve, it will likely never come close to the impact of *Professional Excel Development*.

At the same time, much of the information in these books, I believe, is somewhat dated. For instance, let's take the case of Hungarian Notation. Hungarian Notation is a variable naming convention encouraged by virtually all Excel development books. Even if you've never heard of Hungarian Notation, you've likely seen and used it, if you've ever looked at or learned from example code. It basically says a variable's name should start with a prefix of the variable's type. For instance, `lblCaption`, `intCounter`, and `strTitle` are all examples of Hungarian Notation: the `lbl` in `lblCaption` tells us we're working with a Label object; the `int` in `intCounter` tells us we're working with an integer type, and the `str` in `strTitle` tell us we're working with a string type. If you've done any VBA coding before, this is likely not new information.

You might not know this, however: most modern languages have all but abandoned Hungarian Notation. Microsoft's .NET style guidelines, for instance, even discourage its use. More than a decade has passed since Microsoft last recommended Hungarian Notation. I argue that it's time for a more modern naming style, which I introduce in Chapter 2.

But this book is concerned with more than just naming conventions. I argue that we should change the way we think about development. Previous books have placed significant emphasis on user interface with ActiveX objects and UserForms. This book will eschew these bloated controls; rather, this book will show you how to develop complex interactivity using the spreadsheet as your canvas. You'll see that it's easier and provides for more control and flexibility compared to conventional methods from other books.

In addition, I'll place less emphasis on code and a stronger emphasis on formulas (Chapters 3, 4, and 5). Many books have narrowly defined the principles of advanced Excel in terms of VBA code. But formulas can be powerful. And often they can be used in place of VBA code. You might be surprised by how much interactivity you can create without writing a single line of code. And how much quicker your spreadsheet runs because of it.

This book is divided into two parts. Part I (Chapters 1-5) deals with concepts that are likely already familiar to you. Specifically they concern VBA code and formulas—but I present these concepts in new ways. Part II makes up the last four chapters of the book (Chapters 6-9). These chapters apply concepts from Part I to a real-world example product I built in my consulting experience. Futhermore, in Part II, you'll learn how to input form data without making your spreadsheet bloated. You'll also apply some data analytics used in the field of management science.

However, if you learn anything from my book, it should be that the process of development never stops. The most important skill you'll need is creativity. Just as I saw different ways to approach a problem than my predecessors, so too should you analyze what's being presented to you. Undoubtedly, you'll find even better approaches than I did. I don't expect everyone to agree with my approaches, but what's important is that you understand them, so you can see what works, what doesn't, and why. Because you won't become an advanced Excel developer through rote memorization of the material presented herein; you must learn to think like an advanced developer. This book will teach you the essentials of doing just that.

What to Expect from this Book

This is not a beginner level book. I assume you have intermediate level experience with formulas and Visual Basic for Applications. At the very least, you should be able to understand and write both formulas and code. Complete mastery isn't necessary; because the topics presented in this book are somewhat new, a mastery in these topics might not even help you. All that being said, if you're an experienced Excel user—and you have the aptitude and thirst to learn new things—there's no reason you won't be successful in reading this book! Again, the most important (and cherished) skill that will guarantee your success is creativity.

What's considered "advanced" may mean different things to different people. Here, we're interested in the principles that help us become better spreadsheet users and developers. That said, this book will make use of Excel features such as formulas, tables, conditional formatting, Visual Basic for Applications code, form controls, and charts. For the most part, I will present a brief refresher on what these features do and how they are used. However, you'll find this book moves at a quicker pace than beginner level treatments for these items. Features such as PivotTables, PowerPivot, Power Map, and data tables are not discussed in this book. But you'll find that the principles presented in these pages are extendable to these topics.

Indeed, this book is most concerned with teaching Excel development as first principles. I will explain what they are and how best they are used in practice. Once you learn underlying concepts, extending their use into applications becomes trivial.

Example Files Used in This Book

This book comes with many examples as a complement to the material presented herein. The example files are organized by chapter. Whenever there is a corresponding example file for the material presented, I'll provide you the name of the example file in the text. All example files are freely available to download from the book's Apress web page (www.apress.com/9781484207352). The files are designed to work in Excel 2007 and newer.

The Two Most Important Principles

There are many different ideas and concepts presented in this book. But I'll be daring and attempt to sum them up as two key concepts:

1. When it make sense, do more with less.
2. Break every rule.

■ **Note** The two most important principles are (1) when it makes sense, do more with less, and (2) break every rule.

When It Makes Sense, Do More with Less

You don't need VBA to do everything. Many times, the reason a spreadsheet is slow is because there is too much reliance on code. Similarly, too many formulas—especially volatile functions like `OFFSET` and `INDIRECT`—will almost always slow down a spreadsheet. There are better alternatives to these methods. Often, they require less code and can get more done.

However, we should be wary of brevity for the sake of it. Bill "MrExcel" Jelen and I have a friendly disagreement[1] on whether to use `Option Explicit` in your code. He says he doesn't need it because he always writes perfect code to start with—and that its use needlessly adds more lines of code. I, of course, respectfully disagree. I strongly encourage you to use `Option Explicit`. `Option Explicit` requires that you declare your variables before they're used. That means that you cannot introduce a new variable in your code on the fly. Listing 1-1 shows code without `Option Explicit`; Listing 1-2 shows code with `Option Explicit`.

Listing 1-1. No Option Explicit

```
Public Sub MyResponse()
    ResponseMessage = "Code Executed Successfully!"
    MsgBox ResponseMessage
End Sub
```

Listing 1-2. With Option Explicit

```
Option Explicit

Public Sub MyResponse()
    Dim ResponseMessage as String

    ResponseMessage = "Code Executed Successfully!"
    MsgBox ResponseMessage
End Sub
```

Bill argued using `Option Explicit` required at least one additional line of code for every variable. And it might appear Listing 1-1 is indeed doing more (or at least *the same*) with less code. But, as I show in Chapter 2, not using `Option Explicit` might be more trouble than it is worth. Debugging is much harder without `Option Explicit`, and not using it even encourages sloppy code. From my standpoint, leaving out `Option Explicit` (and the required variable declaration) is simply getting less done with less code. But however you feel on this particular issue, it's worth testing your opinion against that first principle: ask yourself, am I really doing more with less?

Break Every Rule

I truly believe, and stand by, the material presented in this book. But I would have never discovered any of it without departing from conventional wisdom. Again, I'll keep hammering this point until I am blue in the face: the most important takeway from this book is creativity. And you cannot be creative without pushing a few boundaries. Don't be scared to crash a spreadsheet or two in the pursuit of learning.

You'll see in later chapters that some techniques won't always be the best choice for every scenario. For instance, a complex formula that is much faster in practice than a conventional formula might be useless if you must share your spreadsheet and you're the only one who understands it. There will always be an economy between formula readability and utility. I present complex formulas in this book, but I also argue that readability should be a factor in choosing when and where to use them.

[1]Watch Bill and I fight about this on Excel.TV: www.youtube.com/watch?v=yJRLzN3Dzmw.

Most important, you shouldn't be satisfied with Excel's perceived limitations. Over the last several years, I've been blown away by what I've seen others accomplish with Excel. There is a thriving online community dedicated to helping people realize their imaginations with spreadsheets. Whenever I need inspiration, I look to the community.

For your own consideration, I'll provide two examples of my own work that show what can be done with Excel when we think creatively. Figure 1-1 shows a three dimensional maze I created. It might surprise you to learn there is very little code involved. And the "maze" is simply an area chart formatted to look like a three dimensional plane.

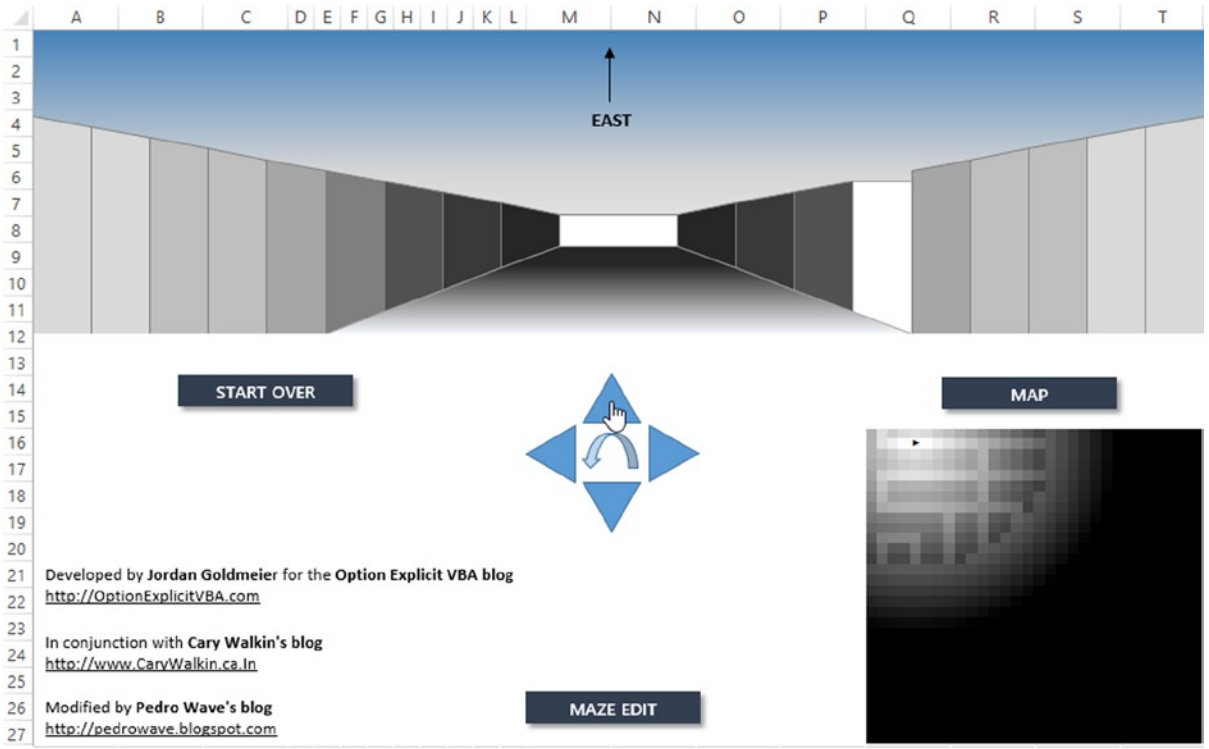

Figure 1-1. *A three dimensional maze, made with Excel*

The second item I would like to present is a periodic table of elements with Excel, shown in Figure 1-2. The periodic table uses a mouseover capability. When the user hovers their mouse over a cell, a macro is executed that updates information about the element. However, the macro uses only a few lines of code, and besides that update, the functionality is largely driven by formula functions. Moreover, that mouseover capability is one I discovered by accident. Before I first wrote about it on my blog, it had been considered impossible.

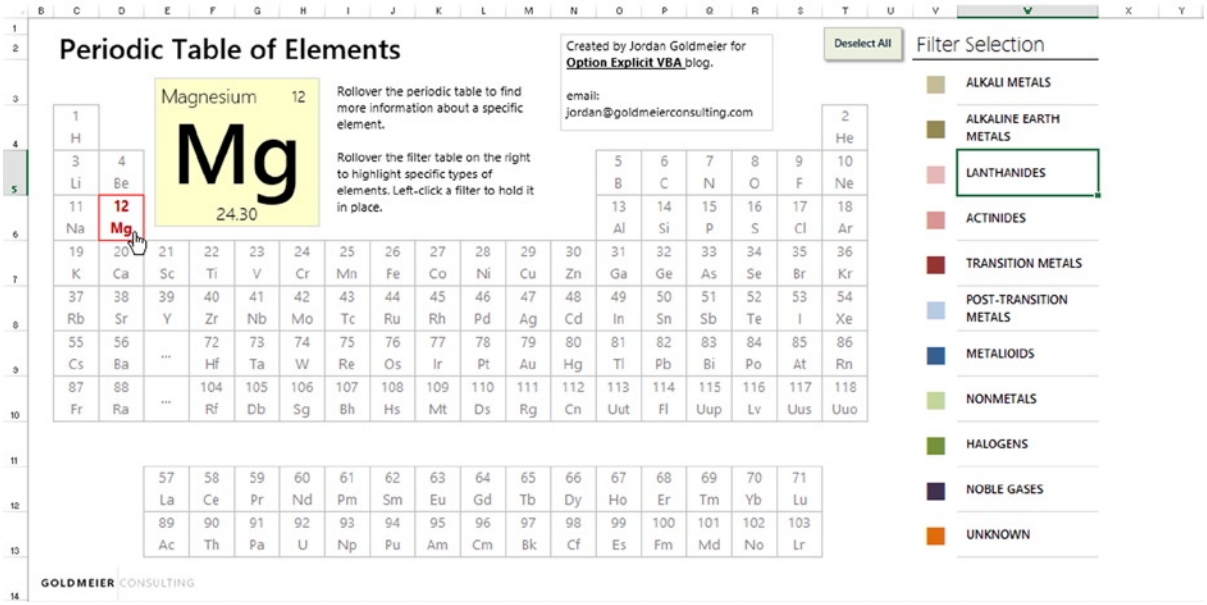

Figure 1-2. *A periodic table of elements with interactivty previously thought impossible with Excel*

Both the three dimensional maze and periodic table are available for you to investigate in the project files included with this book. While it's beyond the scope of this book to explain in detail how these particular spreadsheets were created, they are the direct product of the material I present in the rest of the book. However, if you're interested in reading how these items were developed, see the links in the sidebar.

LINKS ON DEVELOPING A MAZE AND MOUSE OVER MECHANISM

How to Create a Rollover Effect in Excel: Execute a Macro When Your Mouse is Over a Cell

http://optionexplicitvba.blogspot.com/2011/04/rollover-b8-ov1.html

Roll Over Tooltips and Web Actions on a Microsoft Excel Dashboard

www.clearlyandsimply.com/clearly_and_simply/2012/11/roll-over-tooltips-and-web-actions-on-a-microsoft-excel-dashboard.html

Development Principles for Excel Games and Applications

http://optionexplicitvba.com/2013/09/16/development-principles-for-excel-games-and-applications/

Your First Maze

http://optionexplicitvba.com/2013/09/17/your-first-maze-2/

Available Resources

As I said in the previous section, sometimes you need some inspiration to help get you going. Here's a list of resources I use regularly.

Google

Google...Google...Google! Google is your best friend. If you're ever stuck on a problem, simply ask Google the same way you might your friend. Usually, you'll find the results in Excel forums where folks have asked the very same questions.

Chandoo

This site, by Purna "Chandoo" Duggirala, is a phenomenal resource for every Excel developer, from novice to professional. Chandoo covers many topics including dashboards, VBA, data visualization, and formula techniques. His site is also host to a thriving online forum community.

www.chandoo.org

Cleary and Simply

Clearly and Simply is a site by Robert Mundigl. The site is mainly focused on dashboards and data visualization techniques with Excel and Tableau.

www.ClearlyAndSimply.com

Contextures

Debra Dalgleish runs the Contextures web site, which focuses on Excel development and dashboards, particularly with PivotTables. Her approach to dashboards and the use of PivotTables is different from mine, but well worth a read. She is also the author of these Apress Books:

- *Excel Pivot Tables Recipe Book: A Problem-Solution Approach*
- *Beginning PivotTables in Excel 2007: From Novice to Professional*

www.contextures.com

Excel Hero

Excel Hero was created by Daniel Ferry. While his blog is not very active anymore, you will find his older content incredibly useful. Several of his articles have served as the inspiration for the content found in these pages.

www.ExcelHero.com

Peltier Tech

Jon Peltier is a chartmaster. His web site is full of charting tutorials and examples. He provides sage wisdom on data visualization and proper data analysis. His web site covers every conceivable thing you might want to do with a chart in Excel.

www.peltiertech.com

The Last Word

Above all, advanced development is about thinking creatively. You'll see this in practice in the chapters to come. Because some of the material is new, it may appear challenging at first. You may even find yourself frustrated at times. In these moments, it's best to take a break for a moment, find your bearings, and start from the beginning of the section in which you left off. The material is complex, but well within your grasp. I urge you to push through to the end of the book. The material is worth it; but more important, you're worth it. What will you learn in this book will distinguish you. We're only still scratching the surface of what Excel can do. By the time you're finished with this book, you'll be developing work that might even surprise you.