

Excel Tutorial Framework

Executive Summary

Anthony Frehner

There are many people that are not familiar with Microsoft Excel and are unaware how much Excel can help them perform their daily tasks. Just the simple act of opening Excel can scare some away, simply because of the strange interface and multitude of buttons and options that are available to the user. While some may be able to get past this first shock of information overload, history and experience tells us that many who are intimidated are unable to get past this experience.

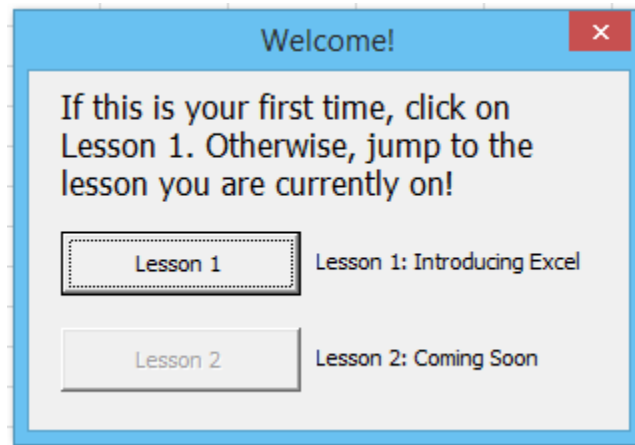
The purpose of my project is to help facilitate the process of learning new items and tasks in Excel, but in a non-traditional manner: I made an easy-to-use and heavily commented framework in VBA that should allow anyone with a basic knowledge of VBA to create and modify their own lessons. By creating this framework, I hope to enable anyone to make a tutorial on any Excel topic simply and easily, even if they have very little VBA knowledge. It can be as simple as writing a paragraph, to as complex as having actions and validations for each step of the tutorial.

To provide an example, I created a simple tutorial that is intended for absolute beginners to Excel, which walks the user through a couple of basic Excel terms, and also contains several actions and a validation.

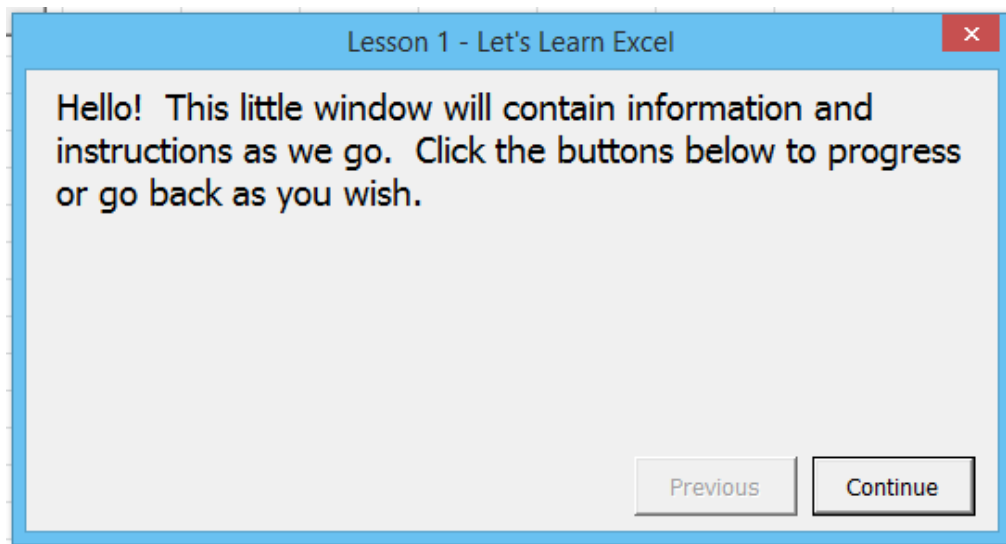
Documentation

Excel Tutorial Framework

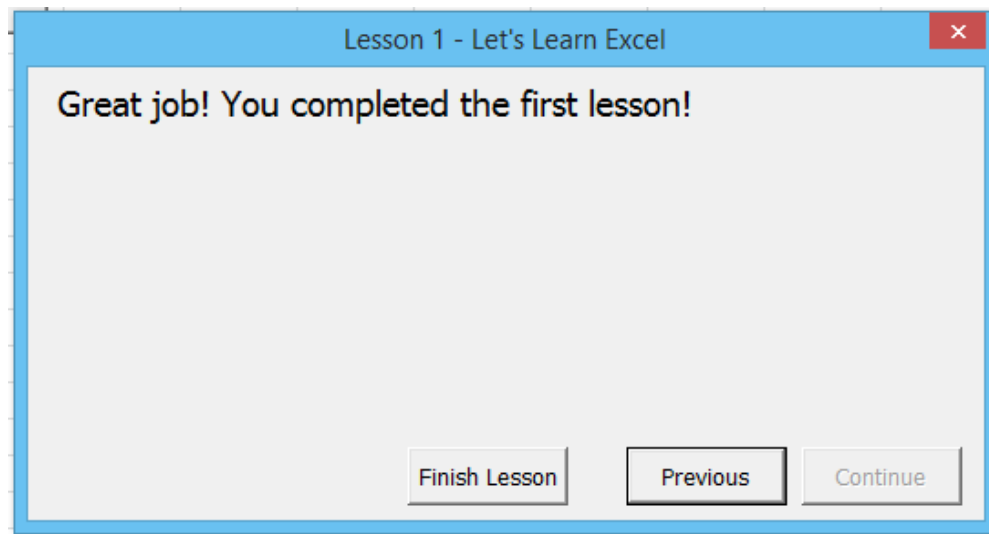
I will outline how the Excel Tutorial Framework displays your tutorial. The first screen shown is what is displayed when Excel is first started, or when the “Open Lesson Window” button is clicked. Notice that you can easily create more than one lesson in a tutorial and users will be able to go to each one simply and easily.



Once the user clicks on “Lesson 1”, then they are presented with the first item in the captions array (which is explained later) with a disabled “Previous” button and a “Continue” button.



The framework automatically takes enables and disables the buttons; all the creator of the tutorial needs to worry about is adding content.



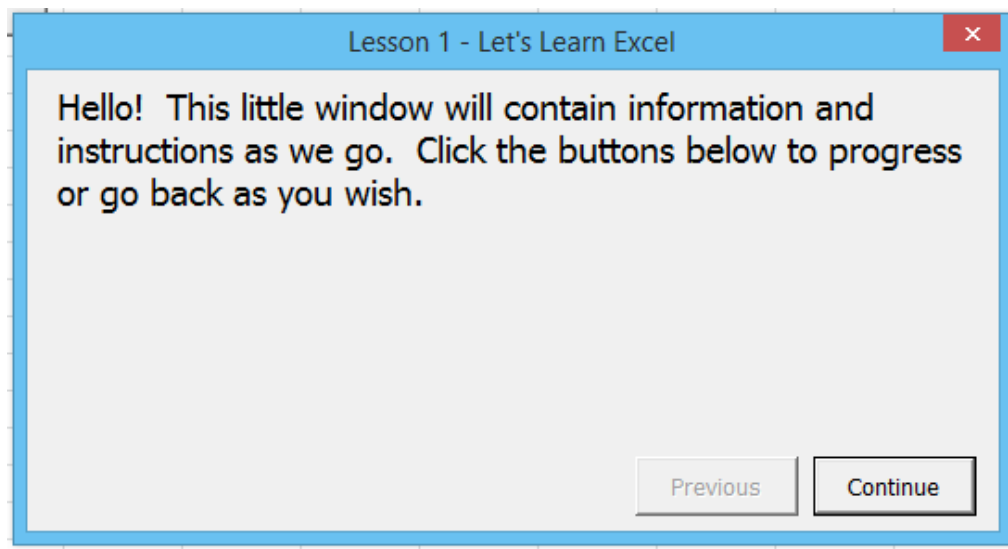
Notice in the example above that the “Finish Lesson” button becomes available when the last step in the tutorial is reached, along with the “Continue” button becoming disabled. Once again, the framework automatically handles this for you. Once the “Finish Lesson” button is clicked, the lesson selection window is displayed again (as seen in the first screenshot).

If you are interested in using the framework yourself, the following three items are the only things you will need to work with and understand in order to get a full tutorial working:

- **Captions**: Captions are the sentences or text that will be displayed when a person is going through the lesson. There is an array of captions (and if you do not know what an array is, that’s ok, just follow the outline and you will still be able to get it to work) that is available for you to use and modify. For example, if you add the line:

captions(0) = "Welcome to my tutorial! We've selected B2, please select C3."

then the first thing the person will see is a popup box that says "Welcome to my tutorial! We've selected B2. Please select C3." Simply increment the number in the parenthesis and add more text, and the framework automatically handles the previous and next buttons for you. Also note that you need to change the variable "numOfCaptions" to be the highest number in the list of captions.



In the example above, the text that says "Hello! This little window..." is the caption.

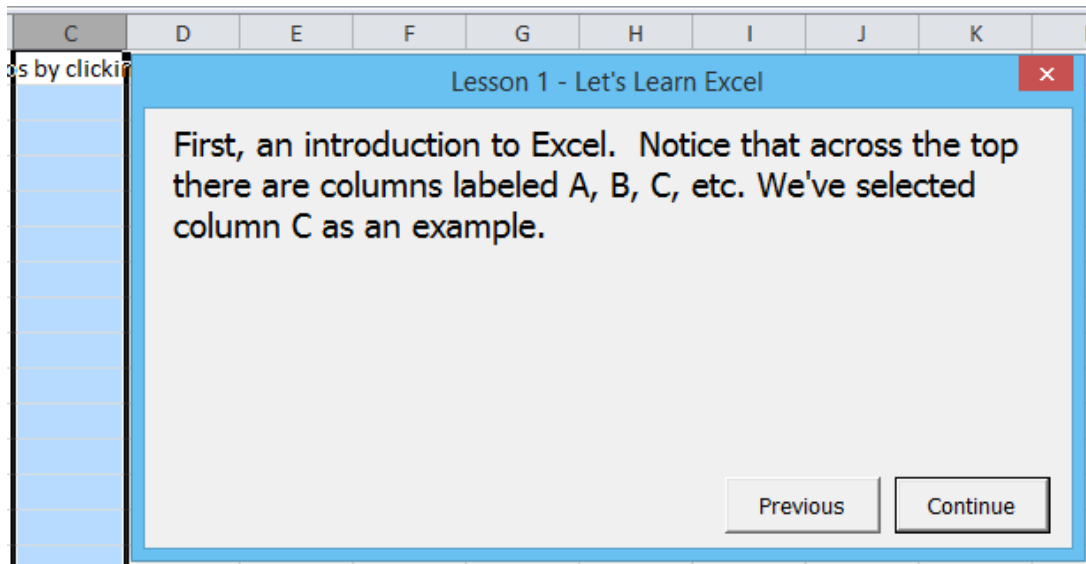
- **Actions**: Actions are code that will be executed when the user gets to that step in the tutorial. The actions numbers are tied to the captions numbers -- but do not worry if a caption does not need an action; the framework will be able to handle that as well. The format for actions is similar to the captions, with the difference being that the text you include in the quotes is the name of a sub procedure. This allows you to write simple or incredibly complex actions that can

be built and tested independently of the main tutorial, and then easily included and automatically run. For example, if you wanted to have Excel select cell B2 when the first caption is shown, then you would write the following code:

```
actions(0) = "subSelectB2"
```

```
Sub subSelectB2()  
    Range("B2").Select  
End Sub
```

The framework automatically runs this action (and thus the sub) when caption(0) is displayed, whether the user presses the previous or continue button. The following screenshot displays the result of an action that selects the column C:



- **Validations:** Validations are code that will be executed only when the user presses the continue button, and will ensure that the user performed a certain

action before allowing him/her to move on to the next step in the tutorial or lesson. Creating validations is very similar to creating actions, in that you add a validation to whichever step you want (and not all steps need validation, which the framework handles) by putting it in the validation array with the same number as the caption, and the text you put in is the name of the sub procedure that contains the code to validate the action. However, the sub procedure for the validation needs you to set the variable “validated” to “TRUE” if the user performed the action correctly, and “FALSE” if the user did not perform the action correctly. For example:

```
validations(0) = “validateSelectC3”
```

```
Sub validateSelectC3()
```

```
    IF ActiveCell.Address = “$C$3” Then
```

```
        validated = TRUE
```

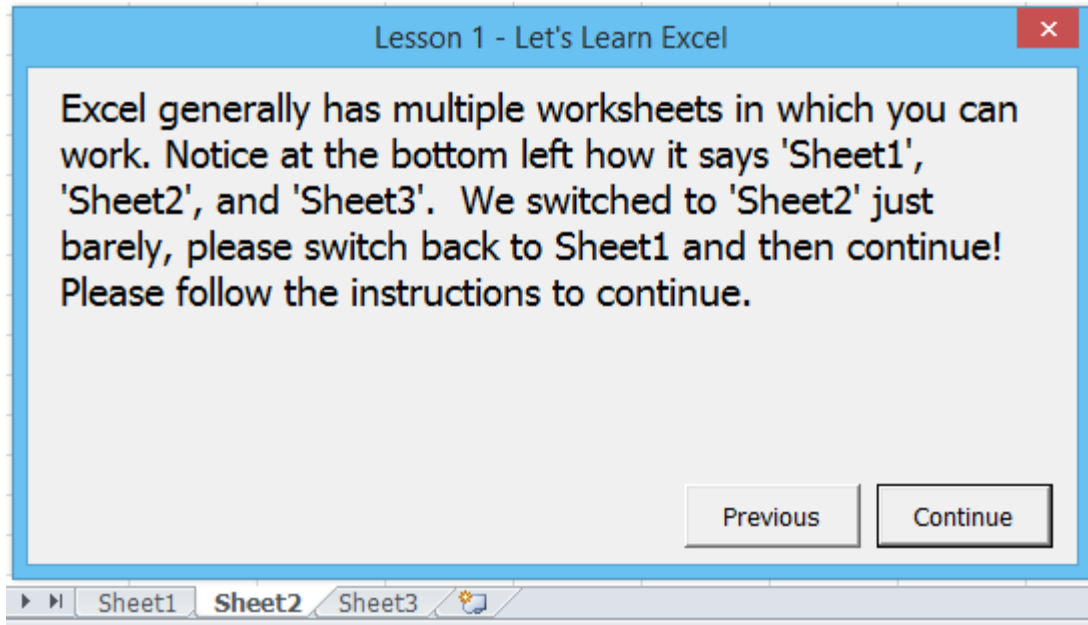
```
    Else
```

```
        validated = FALSE
```

```
    End IF
```

```
End Sub
```

The framework looks at validated to decide if the user passed the validation, and then allows the user to continue to the next step if validated equals true, otherwise a message is added to the caption asking the user to follow the steps outlined and try again.



The screenshot above is an example of what happens when the user fails a validation: the text “Please follow the instructions to continue” is added to the caption, and the user remains on the current step until the validation task is completed.

What I Learned

Excel Tutorial Framework

Throughout the process of building this framework, I learned quite a few things about VBA. Below are the major ideas I learned:

- How to open and call userforms from other userforms
- How to call a sub from a string
- How to handle errors if subs or strings do not exist
- How to hide, display, and disable userform buttons dynamically