

VBA: Tutorial Week 2

- Displaying the number of spelling mistakes
- Changing font properties
- Writing text to a document
- Finding things in a document (text strings, font effects, formatting styles in Word)
- Word collections (Documents, InlineShapes, Shapes, Tables)

Official resource for MS-Office products: <https://support.office.com>

First Tutorial (Monday or Tuesday)

Activities In Tutorial

- TA demos:
 - Used for more complex features (typically multiple steps are required).
 - The tutorial instructor will show on the projector/instructor computer each step for running the feature in Excel.
 - Unless otherwise specified the tutorial material will take the form of a TA demonstrating the use of features in Excel.
 - Slides titled “Lecture Review” are covered for the second time and dealing with less complex material.
 - For this reason they will only be covered briefly in tutorial.
- Student exercises:
 - Used instead of TA demos for simpler features.
 - You will have already been given a summary of how to invoke the feature and the purpose of the exercise is to give you a chance to try it out and get help if needed.

VBA tutorial notes by James Tam

Example: Writing Text To A Document

- The following program will write the number of typographical mistakes at the top of the currently active Word document.
- The written text will have the following font characteristics:
 - Bolded text
 - Dark red in color
 - 24 point

typos = 3

absorb what is useful,
 reject what is useless,
 add what is specifically your own.

--

Bruce Lee (Lay-sui Lung)

Adsfkjasdlkf

Afkasjfk

Askdifkjasdjf

VBA tutorial notes by James Tam

Example 1

- **Example name:** 1TypoCountFormattingFonts

```
Dim numTypos As Long
Dim message As String
```

```
numTypos = ActiveDocument.SpellingErrors.Count
message = "# typos = " & numTypos
MsgBox (message)
Selection.Font.Bold = True
Selection.Font.ColorIndex = wdDarkRed
Selection.Font.Size = 24
```

VBA tutorial notes by James Tam

Student Exercise #1

- **Starting document:** exercise1_starting
- **Document with solution:** exercise1_solution
- Open the starting document and add the following capabilities to the starting program:
 - Selected Text will have the following font effects applied:
 - Bold the text
 - Change the color of the text to violet
 - Change the font to Arial

VBA tutorial notes by James Tam

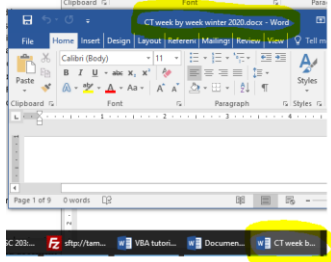
Student Exercise #2

- **Starting document:** exercise2_starting
- **Document with solution:** exercise2_solution
- Open the starting document and add the following capabilities to the starting program:
 - Modify all the text in Word document in the following way:
 - Underline the text with a dotted line.
 - Change the color of the text to bright green.
 - Prompt the user for a font name and the selected text will be changed to this type of font.
 - Prompt the user for a name.
 - Write the name (that the user was prompted to enter in the starting program) after the last part of the text in the document “Ip Man 2”
 - After this is done automatically close and save the document without a user prompt.
 - JT: this exercise may appear fairly daunting for beginners but you can refer to the lecture notes: VBA Part II (focus on the ActiveDocument and the Selection

VBA tutorial notes by James Tam

Finding (Replacing) Text In A Document

- It can be done via the ActiveDocument object
- (This will perform the ‘find’ in the currently active Word document).
- If you have multiple Word documents open this may not always be the document containing your VBA program.
 - Close all Word documents except for your program to avoid confusion!



VBA tutorial notes by James Tam

Finding Things In A Document

- It's done through the 'Find' method and the find is done in the `ActiveDocument` object.
 - i.e. some form of `With ActiveDocument.Content.Find`
 - Some types of things that a VBA program can find:
 - Text
 - Font effects
 - Styles (you have created)

VBA tutorial notes by James Tam

Finding Text: V1

- **Name of the document containing the example:**
`2findReplaceOneCaseSensitive`
 - Features: replaces first instance, case sensitive find, find and replacement strings are constant strings (always the same)

```
Sub findReplaceOneCaseSensitive()  
  With ActiveDocument.Content.Find  
    .Text = "cool"  
    .Replacement.Text = "kewl"  
    .Execute MatchCase:=True, Replace:=wdReplaceOne  
  End With  
End Sub
```

VBA tutorial notes by James Tam

Finding Text: V2

- **Name of the document containing the example:**

3findReplaceAllCaseInsensitive

- Features: replaces all instances, case insensitive find, find and replacement strings are variable (depend upon user input)

```
Sub findReplaceAllCaseInsensitive()
    Dim findWord As String
    Dim replacementWord
    findWord = InputBox("Word to find (try 'cOoL': ")
    replacementWord = InputBox("Replacement word (try 'aBx': ")
    With ActiveDocument.Content.Find
        .Text = findWord
        .Replacement.Text = replacementWord
        .Execute MatchCase:=False, Replace:=wdReplaceAll
    End With
End Sub
```

VBA tutorial notes by James Tam

Writing To A Document

- It is accomplished using the Selection object and the TypeText method
- To move the selection to the start of the document (write text at the start): Selection.HomeKey Unit:=wdStory
- To move the selection to the end of the document (write text at the end): Selection.EndKey Unit:=wdStory
- vbCr
 - It is a Visual basic constant.
 - Stands for 'Visual Basic' carriage return.
 - When concatenated into a string it's equivalent to hitting enter for each vbCr.
 - Example: "hi" & vbCr& "there"
 - This puts 'hi' and 'there' on separate lines

VBA tutorial notes by James Tam

Writing A Text String To A Document

- **Name of the document containing the example:**
4writingToDocument
- **Features:** writes the current location, at the start and end of the document. Spaces and carriage returns are concatenated into the text to be written.

```
Sub writingToDocument()  
    Dim textInsertAtEnd As String  
    textInsertAtEnd = InputBox("Type in text to insert at end: ")  
    Selection.TypeText ("Inserted where the cursor was located")  
    Selection.HomeKey Unit:=wdStory  
    Selection.TypeText ("New text inserted at the very top")  
    Selection.EndKey Unit:=wdStory  
    Selection.TypeText (" " & textInsertAtEnd & vbCr & vbCr)  
End Sub
```

VBA tutorial notes by James Tam

Visually Highlighting Written Text

- To make text stand out font effects (color, bolding, size, font type can be applied to the written text).
- This can be done via the attributes of the **Font** attribute of the **Selection** object.

VBA tutorial notes by James Tam

Highlighting Written Text

- **Name of the document containing the example:**
5modifyingFontText
- **Features:** highlights inserted text by increasing the font size, changing the color and changing the font to the type specified by the user.

```
Sub modifyingFontText()
    Dim insertionText As String
    Dim newFontSize As Long
    Dim newFontName As String
    insertionText = InputBox("Type in text to insert: ")
    newFontSize = InputBox("Size of the font: ")
    newFontName = InputBox("Name of font (e.g. Arial black): ")
```

VBA tutorial notes by James Tam

Highlighting Written Text: 2

```
insertionText = vbCrLf & insertionText & vbCrLf
'Order is crucial! Right after moving the selection to the top is
'when the statements to apply the font formatting effects should
'occur. Inserting other VBA instructions between may change the
'selection.
Selection.HomeKey Unit:=wdStory
Selection.Font.Bold = True
Selection.Font.Size = newFontSize
Selection.Font.Name = newFontName
Selection.Font.ColorIndex = wdBrightGreen
Selection.TypeText (insertionText)
End Sub
```

VBA tutorial notes by James Tam

Second Tutorial (Wednesday or Thursday)

Student Exercise3

- Write a VBA program that will complete the following tasks:
 1. Prompt the user for a string and write this string to the currently active document. Before writing the string it should be formatted as follows:
 - Before the string there should be a carriage return.
 - After the string there should be a colon followed by a space which is followed by three exclamation marks which in turn is followed by two carriage returns.
 2. The string written to the document should be formatted as follows:
 - Font size will be doubled.
 - The text will be italicized.
 - The text will be underlined with a “dot dash” format.
- **Name of the document containing the solution:**
`exercise3_writingTextModifyingFontEffects`

Opening A Document At A Specified Location

- This example requires that the document called "documentExample.docx" is located on the 'C' drive in a subfolder called '203'.
- **Name of the document containing the example:**
6openingAWordDocument

```
Sub openDocument()
    Documents.Open ("C:\203\documentExample.docx")
End Sub
```

VBA tutorial notes by James Tam

Opening Any User Specified Document At A Specified Location

- This example requires a Word document to be located on the 'C' drive in a subfolder called '203'.
- Illustration of the line continuation character for instructions that span multiple lines.
- **Name of the document containing the example:**
7openingUserSpecifiedWordDocument

```
Sub openUserSpecifiedDocument()
    Dim documentName As String
    documentName = InputBox("Type in the name of Word " & _
        "document to open: ")
    Documents.Open ("C:\203\" & documentName & ".docx")
End Sub
```

VBA tutorial notes by James Tam

Inserting And Modifying Images

- This program requires 4 jpg images (pic1, pic2, pic3, pic4) to be located all in one folder.
- It will prompt the user for a path to where these images are located and then insert them into the document.
- The first and the fourth image will then be resized.
- **Name of the document containing the example:**
8insertingModifyingImages

VBA tutorial notes by James Tam

Inserting And Modifying Images (2)

```
Sub insertModifyImages()
    Const GROWTH_SIZE As Long = 1.5
    Const REDUCTION_SIZE As Double = 0.9
    Dim path As String
    Dim newHeight As Long
    path = InputBox("Enter the path to images to insert e.g.
        C:\203\ : ")
    ActiveDocument.InlineShapes.AddPicture (path & "pic1.jpg")
    ActiveDocument.InlineShapes.AddPicture (path & "pic2.jpg")
    ActiveDocument.InlineShapes.AddPicture (path & "pic3.jpg")
    ActiveDocument.InlineShapes.AddPicture (path & "pic4.jpg")
End Sub
```

VBA tutorial notes by James Tam

Inserting And Modifying Images (3)

```

MsgBox ("First image is about to be increased in size by
        GROWTH_SIZE * 100) & "%")
newHeight = ActiveDocument.InlineShapes(1).Height *
        GROWTH_SIZE
ActiveDocument.InlineShapes(1).Height = newHeight
MsgBox ("Fourth image is about to cut in size by
        (REDUCTION_SIZE * 100) & "%")
newHeight = ActiveDocument.InlineShapes(4).Height *
        REDUCTION_SIZE
ActiveDocument.InlineShapes(4).ScaleHeight = newHeight
End Sub

```

VBA tutorial notes by James Tam

Accessing & Modifying Rudimentary Shapes

- **Name of the document containing the example:**
9modifyingSimpleWordShapes

```

Sub modifyShapes()
    MsgBox ("# shapes=" & ActiveDocument.Shapes.Count)
    ActiveDocument.Shapes(1).Height = 25
    ActiveDocument.Shapes(2).Width = 600
    ActiveDocument.Shapes(3).Delete
    MsgBox ("# shapes=" & ActiveDocument.Shapes.Count)
    ActiveDocument.Shapes(2).Fill.ForeColor = wdRed
End Sub

```

VBA tutorial notes by James Tam

Manipulating Tables

- **Name of the document containing the example:**
10manipulatingTables

```
Sub manipulateTables()
    MsgBox ("# tables=" & ActiveDocument.Tables.Count)
    MsgBox ("1st table sorting")
    ActiveDocument.Tables(1).Sort
    MsgBox ("2nd table sorting")
    ActiveDocument.Tables(2).Sort (True)
    ActiveDocument.Tables(3).Shading.Texture =
        wdTextureHorizontal
End Sub
```

VBA tutorial notes by James Tam

Student Exercise4

- Write a VBA program that will prompt the user for an integer that will determine which image will be deleted.
- Error checking to ensure that the integer is not negative or does not exceed the number of images in the currently active document is not necessary.
- **Name of the starting document:**
exercise4_deleting_user_specified_image_starting
- **Name of the document containing the solution:**
exercise4_deleting_user_specified_image_solution

VBA tutorial notes by James Tam