**Unit 2 - VBA-Scripting**

**Objectives**

* Understand the fundamental building blocks of all programming languages: variables, arrays, conditionals, loops, and functions.
* Create simple VBA macros to trigger pop ups and change cell values.
* Gain practice in writing VBA subroutines that utilize variables and conditionals.
* Begin to develop essential coding skills of syntax recollection, pattern recognition, problem decomposition, and debugging.
* Understand the basic syntax of a VBA for loop.
* Understand how to utilize for-loops in conjunction with conditionals to direct logic flow.
* Understand the value of a nested for-loop and gain basic proficiency in their use.
* Refine fundamental coding skills (syntax recollection, pattern recognition, problem decomposition, and debugging).
* Be comfortable formatting spreadsheets using VBA code.
* Understand how to loop through a table using VBA code and check for changes in values.

**Helpful Links**

* [Excel VBA Programming](http://www.homeandlearn.org/excel_vba_practice1.html)

**2.1 Lesson Plan - VBA Vibes**

**Overview**

In today's class, students will get their first exposure to fundamental programming in the form of Visual Basic for Applications (VBA). Today's class will cover basic concepts like variables, arrays, and conditionals -- and serve as the prelude to their future work in Python and JavaScript.

**Class Objectives**

* Students will understand the fundamental building blocks of all programming languages: variables, arrays, conditionals, loops, and functions
* Students will understand how to create simple VBA macros to trigger pop ups and change cell values
* Students will gain practice in writing VBA subroutines that utilize variables and conditionals
* Students will begin to develop their essential coding skills of syntax recollection, pattern recognition, problem decomposition, and debugging

**Summary of Activities in this class**

Review of the inventory spreadsheet and associated VBA code.

Introduction of Modules and Subroutines:

Sub HelloWorld()  
 MsgBox(“Hello World”)  
 End Sub

Adding Buttons to the Worksheet and linking them to a Macro (the Subroutine you wrote above)

Reading and writing values to Cells and Ranges:

Sub ReadAndWriteToCellsAndRanges()  
 Cells(2, 1).Value = “Hello”  
 Dim valueOne As Integer  
 valueOne = Cells(3, 2).Value

Range(“B2”).Value = “World”  
 Dim valueTwo as Integer  
 valueTwo = Range(“E5”).Value  
 End Sub

The Chess Board exercise where you used Cells(row, column) and Range(“RangeReference”) to write the names of the chess pieces. Remember that the Range was tricky if you passed it two arguments:

Range(“A2:A8”).Value = “Fred” ‘Puts Fred in Cell A2, A3, A4, A5, A6, A7, A8

Range(“A2, A8”).Value = “Vinod” ‘Puts Vinod in Cell A2 and A8

Range(“A2”, “A8”).Value = “Angie” ‘Puts Angie in Cell A2, A3, A4, A5, A6, A7, A8 because   
 ‘Range interprets the two arguments as a start and end of the range

Comments start with a single quote: '

Variables are declared with a Dim statement:

Dim *variableName* As *DataType*

For example:

Dim myName as String  
 Dim myAge as Integer  
 Dim mySalary as Double  
 Dim iAmALiar as Boolean

Subroutine names are usually written in TitleCase. That is, the first letter of each word is capitalized.

Variable names are usually written in camelCase. That is, the first letter is lower case, and subsequent words are capitalized. Acronyms are usually capitalized. So a variable named amountOfRAM is appropriate.

String concatenation with the + symbol.

Dim myFullName as String  
 myFullName = myFirstName + “ “ + myLastName

Addition with the + symbol.

Dim myDogs as Integer  
 Dim myCats as Integer  
 Dim myPets as Integer

myDogs = 2  
 myCats = 2

myPets = myDogs + myCats

The TotalCalculator exercise showed how to calculate a total including Tax:

Dim price as Double  
 Dim tax as Double  
 Dim qty as Double  
 Dim total as Double

price = 123.45

tax = 0.075  
 qty = 17  
  
 total = price \* (1 + tax) \* qty

Then we introduced arrays. If variables are a container for one item, then array variables are a container for several items. For example:

Dim myName As String ‘Can contain one string value

Dim myFriends() As String ‘Can contain many string values

Remember the Ingredients list with Peanut Butter, Potato Salad and Dragonfruit.

Arrays start at index 0. So if you declare an array with the number 3, it actually has 4 elements: 0, 1, 2, 3. For example:

Dim myPets(4) As String ‘Looks like it has my dogs and cats in it, right? But look:

myPets(0) = “A 150 Gallon Aquarium full of Cichlids”  
 myPets(1) = “Mabel” ‘dog  
 myPets(2) = “Toby” ‘dog  
 myPets(3) = “Alice” ‘cat  
 myPets(4) = “Kitty” ‘guess…  
  
Then we looked at the Split() function and I did the Shakespeare example:

Dim sentence as String   
 Dim words() as String ‘This is an array

sentence = “To be or not to be”

words = Split(sentence)

Msgbox(words(2)) ‘displays “Or” – remember that words(0) is “To” and words(1) is “be”

Then you did the SentenceSplitter exercise.

Then we looked at the Conditional expressions:

If this > that Then  
 DoSomething()  
 ElseIf a < b Then  
 DoSomethingEvenCooler()  
 Else  
 DoSomethingBoring()  
 End If

Then you did the Choose Your Story exercise.

Then we did the Budget Checker exercise which I did in 3 Subroutines. This exercise was a review of all the code we wrote so far.

Remember that in the third subroutine, we needed to go from the budget amount to the maximum price available, so we had to reduce the budget by the fees:

newPrice = budget / (1 + fees)

Then I showed how to round the result to two decimal places:

newPrice = Round(newPrice, 2)

Then we introduced the VBA homework and class was over.

**If you have any questions or concerns, please slack them to the 01-class-activities channel in slack. Feel free to answer each other’s questions at any time. The TA’s and I will be on slack from time to time to answer questions as well. Remember that you are all in this together and are each other’s first resource for help. Also, please take advantage of the Trilogy tutors that are included in your class fee. Contact Katy if you have questions about the tutor program.**

Next class is Tuesday:

We will be looking at:

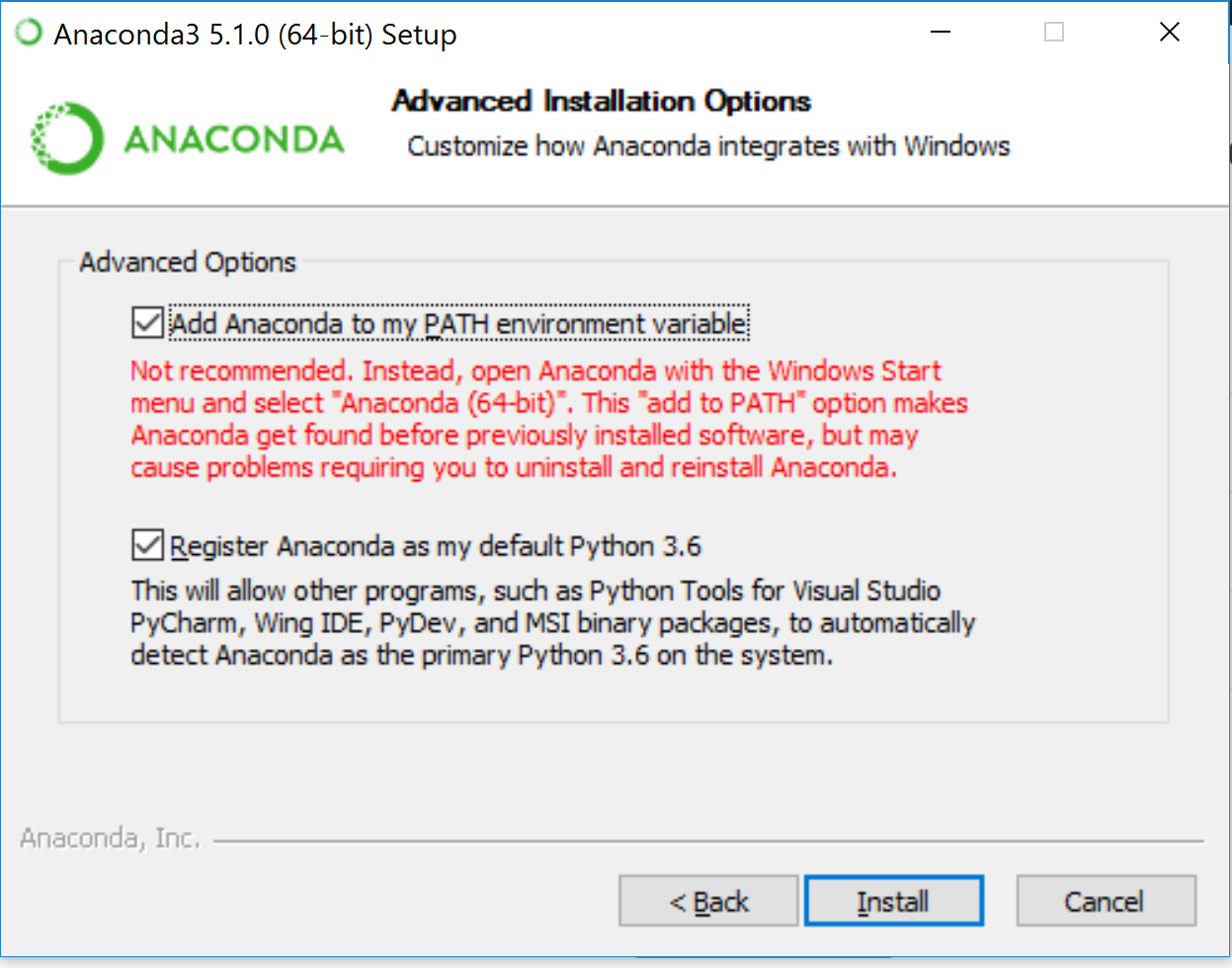
* Loops (For…Next)
* More conditionals (If…Then…ElseIf…Then…Else…End If
* Loops and conditionals working together
* Nested Loops
* Using Counters in Loops
* Formatting cells with VBA code (Range(“A1”).Interior.Color = vbRed)

By Thursday, we will be finishing VBA and starting Python

**See this page for some help on installing Anaconda. We will be using Python/Anaconda on Thursday and Saturday. If you have any trouble, please come to office hours so we can sort it out before class. Thanks.**

# Installing Anaconda

1. Download the [Anaconda Installer](https://www.anaconda.com/download/#windows).
2. Launch the installer.
3. Click next.
4. Read the licensing terms and click "I Agree"
5. Select an install for "Just Me".
6. Select a destination folder to install Anaconda and click the Next button. **NOTE** The default destination should be C:\Users\<your\_user\_name>\anaconda\Anaconda3 and will most likely not need to be changed at all.
7. Add anaconda to your PATH variables and register your default Python to 3.7.



1. Click the Install button.
2. Click the Next Button.
3. **Optional Step** if you have not already installed VSCode you can do so now by clicking the "Install Microsoft VSCode". After the install complete click the Next button.
4. After a completion the there will be a "Thanks for installing Anaconda" dialog box.
5. Finally check that installation has worked by opening git bash and running conda --version.