**VBA** language  **= Visual Basic for Applications**

<https://www.excelcampus.com/vba/excel-vba-macro-shortcuts/>

from AD

<https://www.excel-easy.com/vba/examples/debugging.html>

<http://www.vbindent.com/?indent>

VBA is a loosely type language…don’t need to clearly define variables

Intro to programming logic, building code, Microsoft product integrated w excel

Syntax…so many languages…focus on algorithms, etc

IDE (integrated dvpmt envt)

Step by step logical procedures

4 fundamental bld blocks

**1) Variables** / arrays are effectively the items in a procedure. Nouns of the code

They can be physical things (like an ingredient) or abstractions (like a counter).

sets / lists / maps

In VBA, items can be declared as variables by using the **Dim** followed by the type.

Dim = dimensions – save it w a meaningful name

They can then be assigned a value. \*\*\*see below

What are VBA variable types/**data type…**

String

Integer range is -32867 to 32866

Long

Boulean (T or F)

Double (for decimals)

<http://www.informit.com/articles/article.aspx?p=339929&seqNum=2>

**2) conditionals** are simply declared using the keywords If, Then, Elseif (Elif), Else, and End if.

Theypresent a way to control the flow of logic based on certain conditions being met.

WHAT IS DIFFERENCE BT ELSE & ELSEIF?

**3) Iteration** is the concept of using **loops** to perform a group of tasks repeatedly for a number of times. In almost all languages, we’ll be using **for-loops** and **while loops**.

**For** i = 0 to 20

Next i

**4) Functions** are, in essence, a sort of “sub-processes”. They allow us to create pre-made, re-usable blocks of code which can be called on demand.

SUM, AVERAGE,

Functions are methods that do something to or w arguments

**Start w:**

Sub NAMEofCODE ( ):

…

End Sub

.xlsx vs. .xlsm SECOND FORMAT SAVES MACROS

< > is not =

POP-UP MESSAGES

**MsgBox** "I can learn VBA!"

<https://spreadsheeto.com/vba-message-box/>

change **title**

MsgBox "I can learn VBA!", , “New title”

change **buttons**

MsgBox "I can learn VBA!", vbYesNoCancel , “New title”

Sub **MultipleLines**()

MsgBox "If you are 21 or older, click 'yes'." & vbNewLine & "If you are not, click 'no'", vbYesNo, "MultipleLines"

End Sub

Sub msg**Result**() **interactive buttons**

**If** MsgBox("If you are 21 or older, click 'yes' " & vbNewLine & "If you are not, click 'no'",

vbYesNo, "Age?") = vbYes Then

MsgBox "Please proceed.", , "Result"

Else: MsgBox "Sorry you are not old enough.", , "Result"

End If

End Sub

Sub **ChessBoard**()

**Range**("A1").Value = "Rook"

Range("B1").Value = "Knight"

…

**Range**("A2:H2") = "Pawn"

…

**Cells**(8, 1).Value = "Rook"

Cells(8, 8).Value = "Rook"

…

End Sub

Or

Range("A1, H1").Value = "Rook"

\*\*\*Sub **FullNameCreator**():

Dim FirstName As String

Dim LastName As String

Dim Age As Integer

FirstName = Cells(2, 1).Value

LastName = Cells(2, 2).Value

Age = Cells(2, 3).Value

Dim FullName As String

FullName = FirstName + " " + LastName + Str(Age)

Cells(2, 4).Value = FullName

End Sub

Sub **TotalCalc**()

Dim Price As Double

Dim Tax As Double

Dim Quantity As Integer

Dim Total As Double

Price = Cells(2, 2).Value

Tax = Cells(2, 3).Value

Quantity = Cells(2, 4).Value

Total = Cells(2, 5).Value

Total = Price \* (1 + Tax) \* Quantity

Range("E2").Value = Total

MsgBox ("Your total is $" + Str(Total))

End Sub

An **array** is a group of related items:

Sub SimpleArrays():

' Basic Array Example

' ------------------------------------------

' Create the Ingredients Array

Dim Ingredients(6) as String

' Add Ingredients to the Array

Ingredients(0) = "Chocolate Bar"

Ingredients(1) = "Peanut Butter"

Ingredients(2) = "Jelly"

Ingredients(3) = "Macaroni"

Ingredients(4) = "Potato Salad"

Ingredients(5) = "Dragonfruit"

' Retrieve specific elements of the array

MsgBox(Ingredients(4))

MsgBox(Ingredients(0))

End Sub

Sub SimpleArrays():

' String Splitting Example

' ------------------------------------------

dim Words() as String

dim Shakespeare as String

Shakespeare = "To be or not to be. That is the question"

' Break apart the Shakespeare quote into individual words

Words = Split(Shakespeare, " ")

' Print individual word

MsgBox(Words(5))

End Sub

Note: this above did not work…?

Day 1 (week 2) Student activity #11

Sub SentenceBreaker()

' Retrieve the user sentence and store in variable

Dim Sentence As String

Sentence = Cells(1, 2).Value

MsgBox (Sentence)

' Retrieve the user word numbers and store in variables

Dim num1 As Integer

Dim num2 As Integer

Dim num3 As Integer

num1 = Cells(4, 1).Value

num2 = Cells(5, 1).Value

num3 = Cells(6, 1).Value

MsgBox (num1)

MsgBox (num2)

MsgBox (num3)

' Split the user's sentence into words

Dim SentenceArray() As String

SentenceArray = Split(Sentence, " ")

' Use the word numbers to retrieve the specific words in the sentence

' Remember to offset by the 0 index

Cells(4, 2).Value = SentenceArray(num1 - 1)

Cells(5, 2).Value = SentenceArray(num2 - 1)

Cells(6, 2).Value = SentenceArray(num3 - 1)

End Sub

Sub Conditionals():

' Simple Conditional Example

' ------------------------------------------

If Range("A2").Value > Range("B2").Value Then

MsgBox ("Num 1 is greater than Num 2")

End If

' Simple Conditional with If, Else, and Elseif

' ------------------------------------------

If Range("A5").Value > Range("B5").Value Then

MsgBox ("Num 3 is greater than Num 4")

Elseif Range("A5").Value < Range("B5").Value Then

MsgBox("Num 4 is greater than Num 3")

Else

MsgBox("Num 3 and Num 4 are equal")

End If

' Conditional with Operators (And)

' ------------------------------------------

If (Range("A8").Value > Range("C8").Value And Range("B8").Value > Range("C8").Value) Then

MsgBox ("Both Num 5 and Num 6 are greater than Num 7")

End If

' Conditional with Operators (OR)

' ------------------------------------------

If (Range("A8").Value > Range("C8").Value Or Range("B8").Value > Range("C8").Value) Then

MsgBox ("Either Num 5 and/or Num 6 is greater than Num 7")

End If

End Sub

13. Choose your own story

Sub Begin\_Journey()

' Use conditionals to change message box based on user input

If (Range("B1").value = 1) Then

MsgBox("You choose to enter the wooded forest of doom!")

Elseif (Range("B1").value = 2) Then

MsgBox("You choose to enter the fiery volcano of doom!")

Elseif (Range("B1").value = 3) Then

MsgBox("You choose to enter the terrifying jungle of doom!")

Elseif (Range("B1").value = 4) Then

MsgBox("You choose to enter the bathroom")

Else

MsgBox("Try following directions")

End If

End Sub

END OF DAY 1 (Mon. 7/16/18)

' Part I: Calculate the total after fees

' Part II: Create an alert if the item is above budget after fees.

' Part III: If the item is above budget, correct the price such that it would be "under budget"

Sub BudgetChecker()

' Part I

' 1. Retrieve the Price and Fees from the cells

Dim total As Double

' 2. Use these values to calculate the total

total = Range("F3").Value \* (1 + Range("H3").Value)

' 3. Enter the total into the appropriate cell

Range("L3").Value = total

' Part II

' 4. Create a variable to store budget

Dim budget As Double

budget = Range("B3").Value

' 5. Compare using conditionals whether total is greater than or less than the budget

If budget > total Then

MsgBox ("Under budget")

Else

MsgBox ("Over budget")

' Part III

Dim newPrice As Double

newPrice = budget / (1 + Range("H3").Value)

' Change the price

Range("F3").Value = newPrice

' Change the new total

Range("L3").Value = newPrice \* (1 + Range("H3").Value)

End If

End Sub

Basic **For** loop:

Sub forLoop()

' Create a variable to hold the counter

Dim i As Integer

' Loop through from numbers 1 through 20

For i = 1 To 20

' Iterate through the rows placing a value of 1 throughout

Cells(i, 1).Value = 1

' Iterate through the columns placing a value of 5 throughout

Cells(1, i).Value = 5

' Places increasing values based upon the variable "i" in B2 to B21

Cells(i + 1, 2).Value = i + 1

' Call the next iteration

Next i

End Sub

Sub loops\_and\_loops()

' Loop through first 10 rows

For i = 1 To 10

' Set values in column 1 to "I will eat"

Cells(i, 1).Value = "I will eat "

' Set values in column 2 to the sum of the counter + 10

Cells(i, 2).Value = i + 10

' Set values in column 3 to "Chicken Nuggets"

Cells(i, 3).Value = "Chicken Nuggets"

' Call the next iteration

Next i

End Sub

Modulus function…did NOT work… “Even Row” was in all cells…?

Sub conditional\_loops()

' Create a for loop from 1 to 10

For i = 1 To 10

' Use the modulus function to determine if a number is divisible by 2 (even number)

If Cells(i, 1).Value Mod 2 = 0 Then

' Enter "Even Row" the adjacent cell

Cells(i, 2).Value = "Even Row"

' If the number is not divisible by 2 (odd number)

Else

' Enter "Even Row" the adjacent cell

Cells(i, 2).Value = "Odd Row"

' Close the If/Else Statement

End If

Next i

End Sub

Sub FizzBuzz()

' Loop through the values in Column 1

For i = 2 To 100

'Set cell value to variable

num = Cells(i, 1).Value

' Check if the number is divisible by 3 and 5....

If (num Mod 3 = 0 And num Mod 5 = 0) Then

' If so, print Fizzbuzz

Cells(i, 2).Value = "Fizzbuzz"

' Check if the number is divisible by just 3...

ElseIf (num Mod 3 = 0) Then

' If so, print "Fizz"

Cells(i, 2).Value = "Fizz"

' Check if the number is divisible by just 5...

ElseIf (num Mod 5 = 0) Then

' If so, print "Buzz"

Cells(i, 2).Value = "Buzz"

End If

Next i

End Sub

Note: When I tried to make it Biz Buz Bang…it would NOT go past 99…?

Sub lotto\_winner()

' Create variables to hold winners. (Use "Long" because the numbers exceed the limit for integers)

Dim first\_place As Long

Dim second\_place As Long

Dim third\_place As Long

Dim runner1 As Long

Dim runner2 As Long

Dim runner3 As Long

' Establish the winning ticket numbers

first\_place = 3957481

second\_place = 5865187

third\_place = 2817729

' Establish the runner-up numbers

runner1 = 2275339

runner2 = 5868182

runner3 = 1841402

' Loop through each of the lotto tickets

For i = 1 To 1001

' Check if the lotto number matches the first place winner...

If Cells(i, 3).Value = first\_place Then

' If so, create a message box specifying the first place win

MsgBox " Congratulations " + Cells(i, 1).Value

' Retrieve the values associated with the winner and enter them into the winner's box.

Cells(2, 6).Value = Cells(i, 1).Value

Cells(2, 7).Value = Cells(i, 2).Value

Cells(2, 8).Value = first\_place

' Check if the lotto number matches the second place winner...

ElseIf Cells(i, 3).Value = second\_place Then

Retrieve the values associated with the winner and enter them into the winner's box.

Cells(3, 6).Value = Cells(i, 1).Value

Cells(3, 7).Value = Cells(i, 2).Value

Cells(3, 8).Value = second\_place

' Check if the lotto number matches the second place winner...

ElseIf Cells(i, 3).Value = third\_place Then

Retrieve the values associated with the winner and enter them into the winner's box.

Cells(4, 6).Value = Cells(i, 1).Value

Cells(4, 7).Value = Cells(i, 2).Value

Cells(4, 8).Value = third\_place

' Ends this series of IF/ELSE conditionals

End If

Next i

' Loop through the lotto tickets a second time to find the first instance of a "runner-up" winner

For i = 1 to 1001

' BONUS: Check for runner ups with an OR operator

If Cells(i, 3).Value = runner1 Or Cells(i, 3).Value = runner2 Or Cells(i, 3).Value = runner3 Then

Retrieve the values associated with the winner and enter them into the winner's box.

runner\_up = Cells(i, 3).Value

Cells(5, 6).Value = Cells(i, 1).Value

Cells(5, 7).Value = Cells(i, 2).Value

Cells(5, 8).Value = runner\_up

' If first match is found, exit the for loop

Exit for

End If

Next i

End Sub

A **Nested For**-loop

Sub NestedLoop():

'Loop through rows

For i = 1 To 3

'Loop through the columns

For j = 1 To 5

'Print the Student Name

MsgBox ("Row: " & i & " Column: " & j & " | " & Cells(i, j).Value)

Next j

Next i

End Sub

Count the # of hornets…MsgBox

Sub HornetsNest()

' PART I

' ------------------------------------------------

' Create a variable to hold the number of hornets

Dim HornetsCount as Integer

' Set the initial value for the HornetsCount to 0

HornetsCount = 0

' Loop through all rows

For i = 1 to 6

' Loop through all columns

For j = 1 to 7

' If the value of a cell is equal to Hornets

If Cells(i, j).Value = "Hornets" Then

' Add to the HornetsCounter

HornetsCount = HornetsCount + 1

End If

Next j

Next i

' Show the number of hornets found

MsgBox(HornetsCount & " Hornets Found")

End Sub

Change the word Hornets to Bugs

Sub HornetsNest2()

' PART II

' ------------------------------------------------

' Create a variable to hold the number of hornets

Dim HornetsCount as String

' Loop through all rows

For i = 1 to 6

' Loop through all columns

For j = 1 to 7

' If the value of a cell is equal to Hornets

If Cells(i, j).Value = "Hornets" Then

' Replace the Hornets with Bugs or Bees

Cells(i, j).Value = "Bugs"

End If

Next j

Next i

End Sub

Modify the script a third time, this time keeping in mind that you have a limited number of Bugs and Bees. Use the full set of Bugs and Bees you have available to replace the Hornets. If you run out of Bugs or Bees provide the user with the message: "Oh no! We still have hornets..." NOTE: NOT SURE THIS WORKED…?

Sub HornetsNest()

' PART III

' ------------------------------------------------

' Create a variable to hold the number of hornets

Dim HornetsCount As Integer

' Create a variable to hold the number of bugs and bees

Dim BugsCount As Integer

Dim BeesCount As Integer

' Set the value of Bugs and Bees counters

BugsCount = Range("L2").Value

BeesCount = Range("R2").Value

' Set the initial value for the HornetsCount to 0

HornetsCount = 0

' Loop through all rows

For i = 1 To 6

' Loop through all columns

For j = 1 To 7

' If the value of a cell is equal to Hornets

If Cells(i, j).Value = "Hornets" Then

' Add to the HornetsCounter

HornetsCount = HornetsCount + 1

' Check if we have bugs available

If (BugsCount > 0) Then

' Replace the Hornets with Bugs

Cells(i, j).Value = "Bugs"

' Subtract from the Bugs Count

BugsCount = BugsCount - 1

' Check if we have bees available

ElseIf (BeesCount > 0) Then

' Replace the Hornets with Bees

Cells(i, j).Value = "Bees"

' Subtract from the Bees Count

BeesCount = BeesCount - 1

End If

End If

Next j

Next i

' Show the number of hornets found

MsgBox (HornetsCount & " Hornets Found")

' Create the final message if we still have hornets

If (Range("L2").Value + Range("R2").Value < HornetsCount) Then

MsgBox ("Oh no! We still have hornets... ")

End If

End Sub

Sub StarCounter()

' Create a variable to hold the StarCounter. We will repeatedly use this.

dim StarCounter as Integer

' Loop through each row

for i = 2 to 51

' Initially set the StarCounter to be 0 for each row

StarCounter = 0

' While in each row, loop through each star column

for j = 4 to 8

' If a column contains the word "Full-Star"...

if (Cells(i, j).value = "Full-Star") then

' Add 1 to the StarCounter

StarCounter = StarCounter + 1

end if

Next j

' Once we've completed all rows, print the value in the total column

Cells(i, 9).value = StarCounter

Next i

End Sub

' BONUS: counts the number of rows

lastrow = Cells(Rows.Count, 1).End(xlUp).Row

' Loop through each row

' BONUS: use lastrow variable instead of 51

For i = 2 To lastrow

See this website for **color** guides: <http://dmcritchie.mvps.org/excel/colors.htm>

<https://msdn.microsoft.com/en-us/vba/excel-vba/articles/colorindex-property>

Sub formatter()

' Set the Font color to Red

Range("A1").Font.ColorIndex = 3

' Set the Cell Colors to Red

Range("A2:A5").Interior.ColorIndex = 3

' Set the Font Color to Green

Range("B1").Font.ColorIndex = 4

' Set the Cell Colors to Green

Range("B2:B5").Interior.ColorIndex = 4

' Set the Color Index to Blue

Range("C1").Font.ColorIndex = 5

' Set the Cell Colors to Blue

Range("C2:C5").Interior.ColorIndex = 5

' Set the Color Index to Magenta

Range("D1").Font.ColorIndex = 7

' Set the Cell Colors to Magenta

Range("D2:D5").Interior.ColorIndex = 7

End Sub

HOW DID THE BUTTONS GET IN THERE?

Sub GradeBook()

' Check if the student's grade is greater than or equal to 90...

If Cells(2, 2).Value >= 90 Then

' Establish that the grade is Passing

Cells(2, 3).Value = "Pass"

' Color the Passing grade green

Cells(2, 3).Interior.ColorIndex = 4

' Set the letter grade to "A"

Cells(2, 4).Value = "A"

' Check if the student's grade is greater than or equal to 80...

ElseIf Cells(2, 2).Value >= 80 Then

' Establish that the grade is Passing

Cells(2, 3).Value = "Pass"

' Color the Passing grade green

Cells(2, 3).Interior.ColorIndex = 4

' Set the letter grade to "B"

Cells(2, 4).Value = "B"

' Check if the student's grade is greater than or equal to 70...

ElseIf Cells(2, 2).Value >= 70 Then

' Establish that the grade is Passing

Cells(2, 3).Value = "Warning"

' Color the Passing grade green

Cells(2, 3).Interior.ColorIndex = 6

' Set the letter grade to "C"

Cells(2, 4).Value = "C"

' Check if the students' grade is failing

Else

' Establish that the grade is Failing

Cells(2, 3).Value = "Fail"

' Color the Failing grade red

Cells(2, 3).Interior.ColorIndex = 3

' Set the letter grade to "F"

Cells(2, 4).Value = "F"

End If

End Sub

Sub Reset\_Button():

' Pass the previous grade to the Last Grade row

Cells(12, 2) = Cells(2, 2).Value

Cells(12, 3) = Cells(2, 3).Value

Cells(12, 4) = Cells(2, 4).Value

' Empty out the current grade and remember to set the color back to default

Cells(2, 2).Value = ""

Cells(2, 3).Value = ""

Cells(2, 3).Interior.ColorIndex = 0

Cells(2, 4).Value = ""

End Sub

Sub CheckerBoard()

' Setup a counter to track cell number

Dim cellnumber as Integer

cellnumber = 1

' Loop through each row of the board

For i = 1 to 8

' Loop through each column of the board

For j = 1 to 8

' If we are on a cell that is divisible by 2 then color it black

If (cellnumber mod 2 = 0) then

Cells(i, j).Interior.ColorIndex = 1

' Otherwise color it red

Else

Cells(i, j).Interior.ColorIndex = 3

End if

' Add one to our cell number each time

cellnumber = cellnumber + 1

Next j

' Whenever we start on a new row, we also add one to the cell number (to create the alternation)

cellnumber = cellnumber + 1

Next i

End Sub

\*\*\*FOR TICKER # IN HW

Sub NextCells()

' Set a variable for specifying the column of interest

Dim column As Integer

column = 1

' Loop through rows in the column

For i = 2 To 6

' Searches for when the value of the next cell is different than that of the current cell

If Cells(i + 1, column).Value <> Cells(i, column).Value Then

' Message Box the value of the current cell and value of the next cell

MsgBox (Cells(i, column).Value & " and then " & Cells(i + 1, column).Value)

End If

Next i

End Sub

\*\*\*ANOTHER EX FOR TICKER # IN HW

Sub credit\_card()

' Set an initial variable for holding the brand name

Dim Brand\_Name As String

' Loop through all credit card purchases

For i = 2 To 101

' Check if we are still within the same credit card brand, if we are not...

If Cells(i + 1, 1).Value <> Cells(i, 1).Value Then

' Message Box the unique Bank Name

MsgBox(Cells(i, 1).Value)

End If

Next i

End Sub

Ticker + totaling the charges…

Sub credit\_card()

' Set an initial variable for holding the brand name

Dim Brand\_Name As String

' Set an initial variable for holding the total per credit card brand

Dim Brand\_Total As Double

Brand\_Total = 0

' Keep track of the location for each credit card brand in the summary table

Dim Summary\_Table\_Row As Integer

Summary\_Table\_Row = 2

' Loop through all credit card purchases

For i = 2 To 101

' Check if we are still within the same credit card brand, if it is not...

If Cells(i + 1, 1).Value <> Cells(i, 1).Value Then

' Set the Brand name

Brand\_Name = Cells(i, 1).Value

??? ' Add to the Brand Total

Brand\_Total = Brand\_Total + Cells(i, 3).Value

' Print the Credit Card Brand in the Summary Table

Range("G" & Summary\_Table\_Row).Value = Brand\_Name

' Print the Brand Amount to the Summary Table

Range("H" & Summary\_Table\_Row).Value = Brand\_Total

' Add one to the summary table row

Summary\_Table\_Row = Summary\_Table\_Row + 1

' Reset the Brand Total

Brand\_Total = 0

' If the cell immediately following a row is the same brand...

Else

' Add to the Brand Total

Brand\_Total = Brand\_Total + Cells(i, 3).Value

End If

Next i

End Sub

HOW DOES THE SUMMARY BOX GET YELLOW (not in code…?),

AND WHAT IS THE “SUMMARIZE CARDS” BUTTON

<https://support.microsoft.com/en-us/help/142126/macro-to-loop-through-all-worksheets-in-a-workbook>

<https://stackoverflow.com/questions/39581487/loop-through-all-worksheets-in-workbook>

Made it to day 3, exercise 6…got the above link from “07-Stu\_WellsFargo\_Pt1”

But did not examine this.