Boot camp notes

Sat Sep 14

Nested function-function within a function

Sub StarCounter()

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

Dim StarCounter As Integer

' 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

' 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

\*\*\*\* This stuff is in workirking with ranges part of the student guide.

Hornets Nest Activity

' Part I: Count the number of Hornets Found

' Part II: Each time you find Hornets replace them with Bugs

' Part III: You have a max amount of Bees and Hornets, utilize no more than what's provided.

' If there are still hornets left, provide the user with a message stating: "Oh no! We still have hornets..."

Sub HornetsNest()

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

' 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

Wells Fargo Macro

' Steps:

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

' Part I:

' 1. Extract words before the phrase "\_Wells\_Fargo" to figure out which State.

' 2. Add the State to the first column of each spreadsheet.

' 3. Convert the headers of each row to simply say the year.

' 4. Convert the cells to currency format

Sub WellsFargo\_PtI()

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

' LOOP THROUGH ALL SHEETS

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

For Each ws In Worksheets

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

' INSERT THE STATE

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

' Created a Variable to Hold File Name, Last Row, Last Column, and Year

Dim WorksheetName As String

' Determine the Last Row

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

' Grabbed the WorksheetName

WorksheetName = ws.Name

' MsgBox WorksheetName

' Split the WorksheetName

State = Split(WorksheetName, "\_")

' MsgBox State(0)

' Add the State to the Column

ws.Range("A1").EntireColumn.Insert

' Add the word State to the First Column Header

ws.Cells(1, 1).Value = "State"

' Add the State to all rows

ws.Range("A2:A" & LastRow) = State(0)

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

' CORRECT THE YEAR NUMBERS

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

' Determine the Last Column Number

LastColumn = ws.Cells(1, Columns.Count).End(xlToLeft).Column

' Rename the Year columns by looping through and renaming each

For i = 3 To LastColumn

YearHeader = ws.Cells(1, i).Value

YearSplit = Split(YearHeader, " ")

' MsgBox YearSplit(0)

ws.Cells(1, i).Value = YearSplit(3)

' MsgBox Cells(1, i)

' MsgBox YearSplit(3)

Next i

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

' CORRECT THE CURRENCY FORMAT

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

' Add the currency

For i = 2 To LastRow

For j = 2 To LastColumn

ws.Cells(i, j).Style = "Currency"

Next j

Next i

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

' FIXES COMPLETE

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

Next ws

MsgBox ("Fixes Complete")

End Sub

Sub stockchallenge():  
  
    ' Set Dimensions  
    Dim total As Double  
  
    ' Start total out at zero  
    total = 0  
  
    ' get the row number of the last row with data  
    RowCount = Cells(Rows.Count, "A").End(xlUp).Row  
  
    ' Set title row  
    Range("I1").Value = "Ticker"  
    Range("J1").Value = "Total Stock Volume"  
  
    For i = 2 To RowCount  
  
        ' If ticker changes then print results  
        If Cells(i + 1, 1).Value <> Cells(i, 1).Value Then  
  
            ' Stores results in variable  
            ' Think about what you should be adding to total  
            ' YOUR CODE HERE  
  
            ' Print ticker symbol in the respective column for the `ticker`  
            ' YOUR CODE HERE  
              
            ' Print total in the respective column for the `total stock volume`  
            ' YOUR CODE HERE  
  
            ' Reset Total  
            total = 0  
  
            ' Move to next row  
            j = j + 1  
  
        ' Else keep adding to the total volume  
        Else  
            total = total + Cells(i, 7).Value  
  
        End If  
  
    Next i  
  
End Sub

Python Notes

﻿# -\*- coding: UTF-8 -\*-

"""PyBank Homework Solutio==n."""

# Dependencies

import csv

import os

# Files to load and output (Remember to change these)

file\_to\_load = os.path.join("Resources", "budget\_data.csv")

file\_to\_output = os.path.join("analysis", "budget\_analysis.txt")

# Track various financial parameters

total\_months = 0

month\_of\_change = []

net\_change\_list = []

greatest\_increase = ["", 0]

greatest\_decrease = ["", 9999999999999999999]

total\_net = 0

# Read the csv and convert it into a list of dictionaries

with open(file\_to\_load) as financial\_data:

reader = csv.reader(financial\_data)

# Read the header row

next(reader)

# Extract first row to avoid appending to net\_change\_list

first\_row = next(reader)

total\_months = total\_months + 1

total\_net = total\_net + int(first\_row[1])

prev\_net = int(first\_row[1])

for row in reader:

# Track the total

total\_months = total\_months + 1

total\_net = total\_net + int(row[1])

# Track the net change

net\_change = int(row[1]) - prev\_net

prev\_net = int(row[1])

net\_change\_list.append(net\_change)

month\_of\_change = month\_of\_change + [row[0]]

#Calculate the greatest increase

if net\_change > greatest\_increase[1]:

greatest\_increase[0] = row[0]

greatest\_increase[1] = net\_change

# Calculate the greatest decrease

if net\_change < greatest\_decrease[1]:

greatest\_decrease[0] = row[0]

greatest\_decrease[1] = net\_change

# Calculate the Average Net Change

net\_monthly\_avg = sum(net\_change\_list) / len(net\_change\_list)

# Generate Output Summary

output = (

f"\nFinancial Analysis\n"

f"----------------------------\n"

f"Total Months: {total\_months}\n"

f"Total: ${total\_net}\n"

f"Average Change: ${net\_monthly\_avg:.2f}\n"

f"Greatest Increase in Profits: {greatest\_increase[0]} (${greatest\_increase[1]})\n"

f"Greatest Decrease in Profits: {greatest\_decrease[0]} (${greatest\_decrease[1]})\n")

# Print the output (to terminal)

print(output)

# Export the results to text file

with open(file\_to\_output, "w") as txt\_file:

txt\_file.write(output)