Miscellaneous Topics on Looping

Larry Caretto
Mechanical Engineering 209

Computer Programming for Mechanical Engineers

March 7, 2017

California State University Northridge

Outline

- Review last lecture
  - Cells command
  - Nested for loops
- Leaving a function or loop early
- Formatting worksheet output
- Work on programming assignment four on looping

Review The Cells Command

- Way to address cell locations by row number and column number
  - Columns A, B, C, are columns 1, 2, 3, etc.
  - Columns Z, AA, AB: columns 26, 27, 28
  - ZZ and AAA are columns 702 and 703
  - Column XFD is column 16384
- Use: Cells(<row>,<col>).Value
  - Note that row comes before column
  - E.g.: Cell C5 is referred to as Cells(5,3)

Review Nested For Loops

- The simplest example of nested for loops is two loops forming a table

```
For row = firstRow to lastRow
  For col =firstCol to lastCol
    <code for computations for
       each row-col combination>
  Next col
Next row
```



```
Sub simpletwoWayTable()

    Dim velocity As Long, mass As Long
    Dim row As Long, col As Long

    row = 1
    For mass = 10 To 100 Step 10
        col = 1
        For velocity = 5 To 25 Step 5
            Cells(row, col).Value = mass * _
                                    velocity ^ 2 / 2
            col = col + 1
        Next velocity
        row = row + 1
    Next mass

End Sub
```

Outer for loop increments mass

Inner for loop increments velocity

Column adjustment in inner for loop

Row adjustment in outer for loop

Nested for loops to get table of kinetic energy = $mV^2/2$

Result of Two-Way Table Code

	A	B	C	D	E
1	125	500	1125	2000	3125
2	250	1000	2250	4000	6250
3	375	1500	3375	6000	9375
4	500	2000	4500	8000	12500
5	625	2500	5625	10000	15625
6	750	3000	6750	12000	18750
7	875	3500	7875	14000	21875
8	1000	4000	9000	16000	25000
9	1125	4500	10125	18000	28125
10	1250	5000	11250	20000	31250

California State University Northridge Need to provide headers and title

The Format function in VBA

- The Format function defines the appearance of a variable
- Its structure is <string variable> = Format(<variable>, <format string>)
  - <string variable> contains the formatted variable what can be used as output to the worksheet or another function/sub
  - <variable> is the variable being formatted
  - <format string> is described on next slide

California State University Northridge 7

Using the Format String

- <string variable> = Format(<variable>, <format string>)
  - Example: s = Format(x, "#,###.00")
  - x is the variable formatted and the string, "#,###.00" is an example of a format string that formats the variable x in this example
- For numerical variables we can determine number of decimal places, use of an E format or percent, and the use of a comma for numbers > 999

California State University Northridge 8

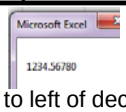
Format Codes

- 0 place a digit in this position, even if it is a lead or trailing zero
- # place a digit in this position only if it is **not** a leading or trailing zero
- . Notes position of the decimal point
- E asks for power of ten notation
- + or – asks for explicit display of sign (used in font of number or after E)
- % multiply by 100; append % sign

California State University Northridge 9

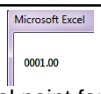
Format Examples

```
Dim x As Double
x = 1234.5678
MsgBox Format(x, "0.00000")
```



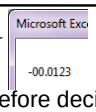
Note trailing zero and all digits to left of decimal

```
Dim i as Integer
i = 1
MsgBox Format(i, "0000.00")
```



Lead and trailing zeroes and decimal point for integer

```
MsgBox Format(-0.012345678, _
              "00.000E00")
```

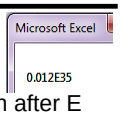


Note automatic minus sign & extra 0 before decimal

California State University Northridge 10

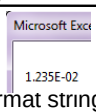
More Format Examples

```
MsgBox Format(0.012345678, _
              "0.000E00")
```



Note incorrect result without minus sign after E

```
MsgBox Format(0.012345678, _
              "0.000E+00")
```



Note correct result even with "E+" in format string

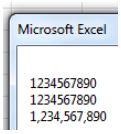
```
Dim x As Double
x = 1234.5678
MsgBox Format(x,"00.000E+0000")
```

Note incorrect result with extra 0 before decimal

California State University Northridge 11

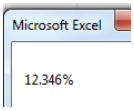
Still More Format Examples

```
Dim x As Double
Dim i As Long
i = 1234567890
MsgBox Format(i, "#") & vbCrLf _
  & Format(i, "0") & vbCrLf  _
  & Format(i, "#,###")
```



Note ability to get repeating commas with limited format

```
Dim x As Double
x = 0.123456
MsgBox Format(x, "0.000%")
```



Note automatic conversion from decimal to percent, rounding, and all leading digits

California State University Northridge 12

Using Format for Worksheet

```
Dim x As Double, y As Double
x = 0.123
y = 0.12345
Range("a1").Value = Format(x, "0.0000")
Range("a2").Value = Format(y, "0.0000")
Range("a3").Value = x
Range("a3").NumberFormat = "0.0000"
```

	A
1	0.123
2	0.1235
3	0.1230

- Want to print all numbers with four decimal places
  - The “0.0000” format for x in cell “A1”cannot over-come the Excel’s dropping of all trailing zeroes
  - Rounding is used if necessary as in y
  - Use two steps, as shown above, first set value then set NumberFormat using same format codes

Formats with Labels

- Can use format command to get row and column labels obtained with custom formats in first program assignment

```
mass = 5
For col = 3 to 10
   cells(2,col).Value = “m = ” _
      & format(mass, “#”) & “ kg”
   mass = mass + 5
Next col
```


Result of Format Code

	A	B	C	D	E	F	G	H	I	J
1										
2			m = 5 kg	m = 10 kg	m = 15 kg	m = 20 kg	m = 25 kg	m = 30 kg	m = 35 kg	m = 40 kg
3										

```
mass = 5
For col = 3 to 10
   cells(2,col).Value = “m = ” _
      & format(mass, “#”) & “ kg”
   mass = mass + 5
Next col
```

Exercise: Write code that produces velocity labels in cells B3:B7 as shown on next slide

Exercise

- Modify the code for producing labeled mass values in rows to produce labeled velocity values in columns as shown below

	A	B	C	D	E	F
1						
2			m = 5 kg	m = 10 kg	m = 15 kg	m = 20 kg
3		V = 2 m/s				
4		V = 4 m/s				
5		V = 6 m/s				
6		V = 8 m/s				
7		V = 10 m/s				
8						

Summing Infinite Series

- $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$
- Consider series a sum of terms, T_n, where $T_n = \frac{x^n}{n!}$ so that $e^x = \sum_{n=0}^{\infty} T_n$
- Ratio of successive terms

$$\frac{T_n}{T_{n-1}} = \frac{\frac{x^n}{n!}}{\frac{x^{n-1}}{(n-1)!}} = \frac{x^n}{x^{n-1}} \frac{(n-1)!}{n!} = x \frac{(n-1)!}{n(n-1)!} = \frac{x}{n}$$

- How can we use this in a code?

Verify Recursive e^x Result

- Derived result below with $T_0 = 1$

$$\frac{T_n}{T_{n-1}} = \frac{x}{n} \Rightarrow T_n = \frac{x}{n} T_{n-1}$$

- $T_1 = (x/1)T_0 = x(1) = x$
- $T_2 = (x/2)T_1 = (x/2)(x) = x^2/2 = x^2/2!$
- $T_3 = (x/3)T_2 = (x/3)(x^2/2!) = x^3/3!$
- $T_4 = (x/4)T_3 = (x/4)(x^3/3!) = x^4/4!$
- And on it goes giving the original series: $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$

Another Infinite Series

- $\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$
- We can write this series as $\cos(x) = \sum_{n=0}^{\infty} T_n$ where $T_n = (-1)^n x^{2n}/(2n)!$ ($T_0=1$)
- What is the ratio of successive T_n terms?

$$\frac{T_n}{T_{n-1}} = \frac{\frac{(-1)^n x^{2n}}{(2n)!}}{\frac{(-1)^{n-1} x^{2n-2}}{(2n-2)!}} = \frac{(-1)x^{2n}}{x^{2n-2}} \frac{(2n-2)!}{(2n)!} = \frac{-x^2(2n-2)!}{2n(2n-1)(2n-2)!} = \frac{-x^2}{2n(2n-1)}$$

California State University Northridge

Verify Recursive cos(x) Result

- Derived result below with $T_0 = 1$

$$\frac{T_n}{T_{n-1}} = \frac{-x^2}{2n(2n-1)} \Rightarrow T_n = \frac{-x^2 T_{n-1}}{2n(2n-1)}$$

- $T_1 = (-x^2/[2\bullet1(2\bullet1-1)]T_0 = (-x^2/2)(1) = -x^2/2$
- $T_2 = (-x^2/[2\bullet2(2\bullet2-1)]T_1 = [-x^2/(4\bullet3)](-x^2/2) = x^4/4!$
- $T_3 = (-x^2/[2\bullet3(2\bullet3-1)]T_2 = [-x^2/(6\bullet5)](x^4/4!) = -x^6/6!$
- $T_4 = (-x^2/[2\bullet4(2\bullet4-1)]T_3 = [-x^2/(8\bullet7)](-x^6/6!) = x^8/8!$
- And on it goes giving the original series:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} + \cdots$$

California State University Northridge

Summing Cosine Infinite Series

```
term = 1
sum = term
For n = 1 To maxIterations
    term = -term * x^2 / (2 * n * (2*n-1))
    sum = sum + term
    if Abs(term) <= maxRelErr * _
                            Abs(sum) Then
        myCos = sum
        Exit Function
    End If
Next n
myCos = "ERROR: Max iterations exceeded"
```

California State University Northridge