COMP1022Q
Introduction to Computing with Excel VBA

More on For Loops

Gibson Lam and David Rossiter

Outcomes

- After completing this presentation, you are expected to be able to:
  1. Use for loops with a step value
  2. Write nested loops in VBA

For…Next

- We already know how to use a for loop

```
For counter = start To end
        . . .statement(s). . .
Next counter
```

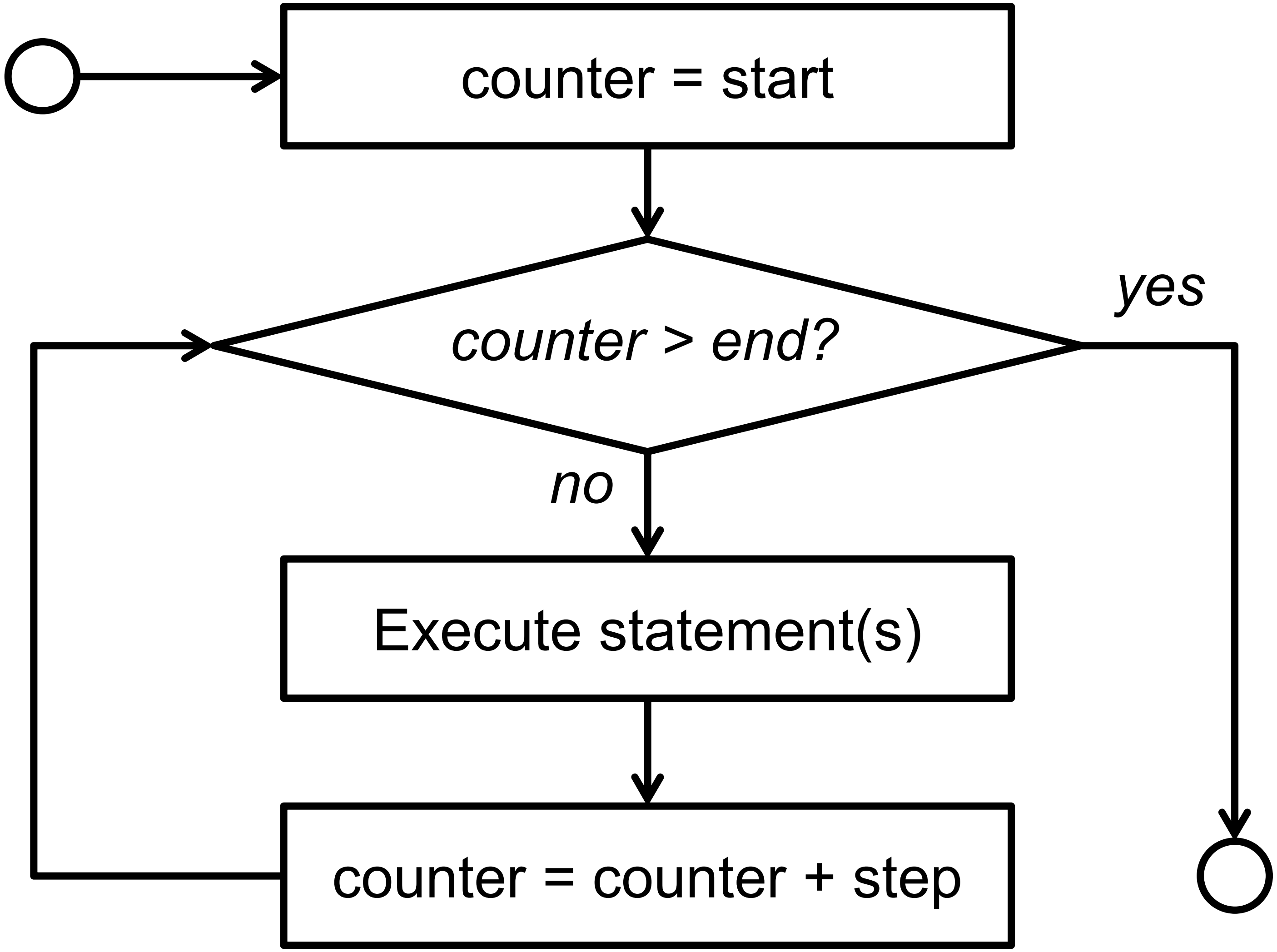
- Until now, the *counter* has always been increased by exactly one after the loop content is executed
- The loop executes until the *counter* is equal to the value of *end*

For…Next with Step

```
For counter = start To end Step step
        . . .statement(s). . .
Next counter
```

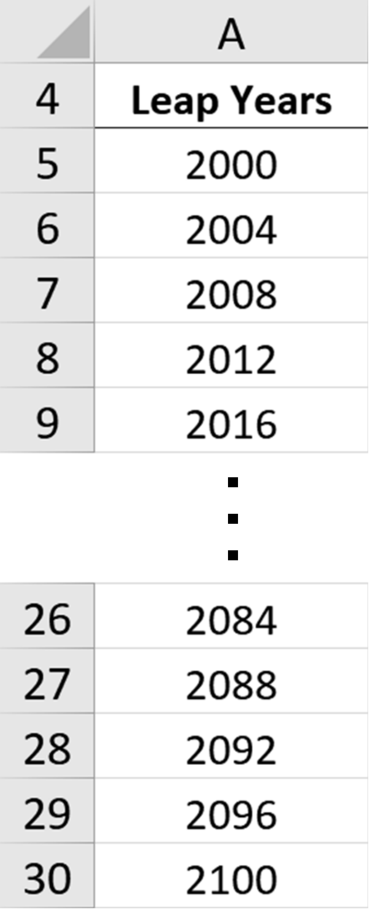
- Now let's look at using a for loop with a ***step*** value
- Using a *step* value, the *counter* can be increased or decreased by a fixed amount (the step) each time
- The *counter* is equal to *start* in the first iteration but it **may not be equal to** *end* in the last iteration

The Flow of For…Next with Step



An Example of For…Next with Step 1/3

- The following example puts the leap years from year 2000 up to 2100 in the cells in column A
- In the example, we assume that leap years occur every 4 years and year 2000 is a leap year (it is a bit more complicated than that in reality)



	A
4	**Leap Years**
5	2000
6	2004
7	2008
8	2012
9	2016
	⋮
26	2084
27	2088
28	2092
29	2096
30	2100

An Example of For…Next with Step 2/3

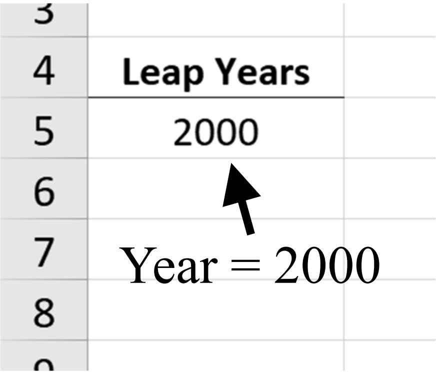
- Here is the code of the example:

```
Dim Year As Integer, Row As Integer

' Start from Row 5
Row = 5
For Year = 2000 To 2100 Step 4
    Cells(Row, 1).Value = Year

    ' Move to the next row
    Row = Row + 1
Next Year
```

Loop body

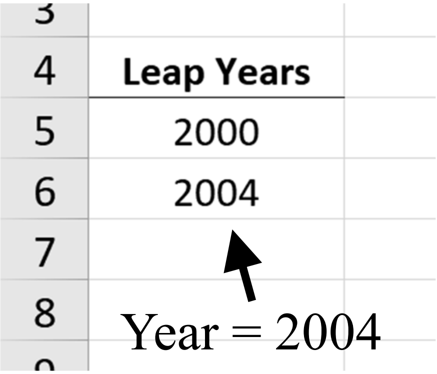
Loop counter, increasing by 4 each time the loop is run

An Example of For…Next with Step 3/3



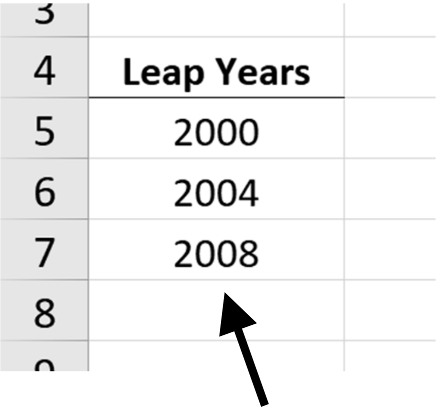
3	
4	**Leap Years**
5	2000
6	
7	
8	

Year = 2000



3	
4	**Leap Years**
5	2000
6	2004
7	
8	

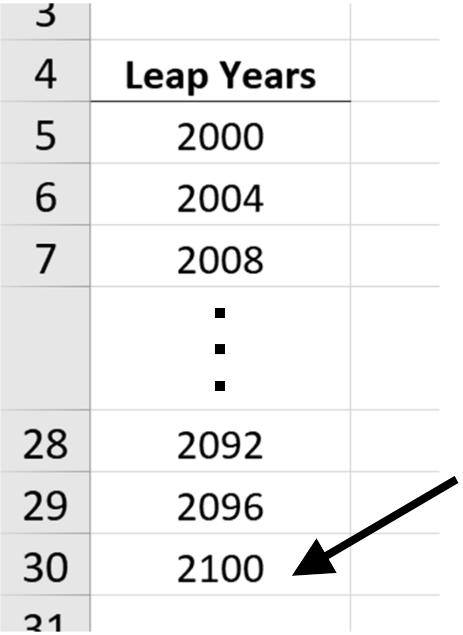
Year = 2004



3	
4	**Leap Years**
5	2000
6	2004
7	2008
8	

Year = 2008

Repeat loop
26 times
in total

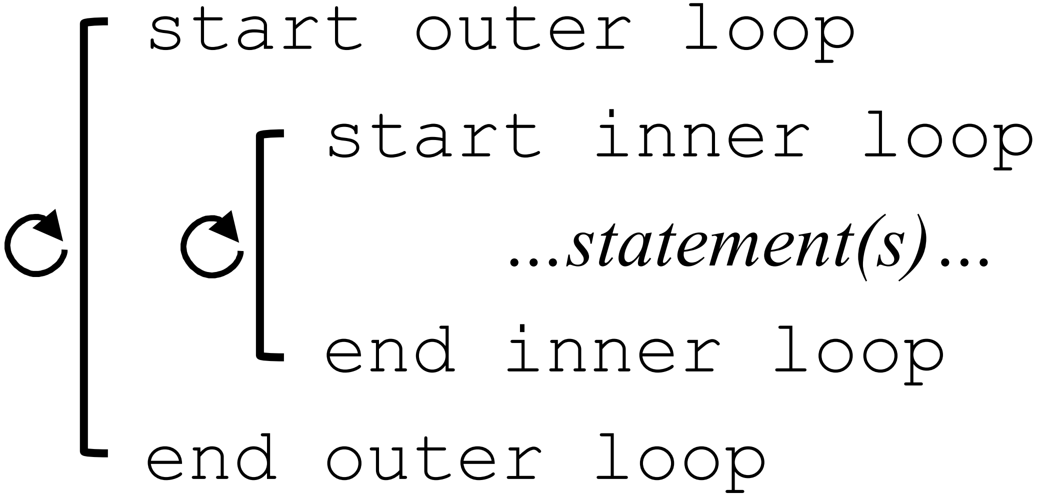


3	
4	**Leap Years**
5	2000
6	2004
7	2008
	⋮
28	2092
29	2096
30	2100
31	

Year = 2100

Nested Loops

- A *nested loop* is a loop within a loop (the loops can be any kind of loop, e.g., while loops or do loops)



```
start outer loop
    start inner loop
            ...statement(s)...
    end inner loop
end outer loop
```

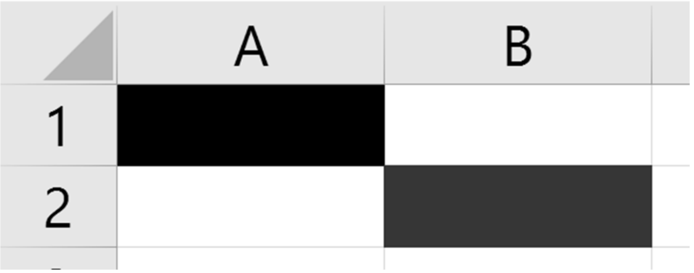
- In the following examples, we demonstrate nested loops using for loops, to show all the background colours and to generate a chess board

ColorIndex Numbers

These are discussed in another presentation

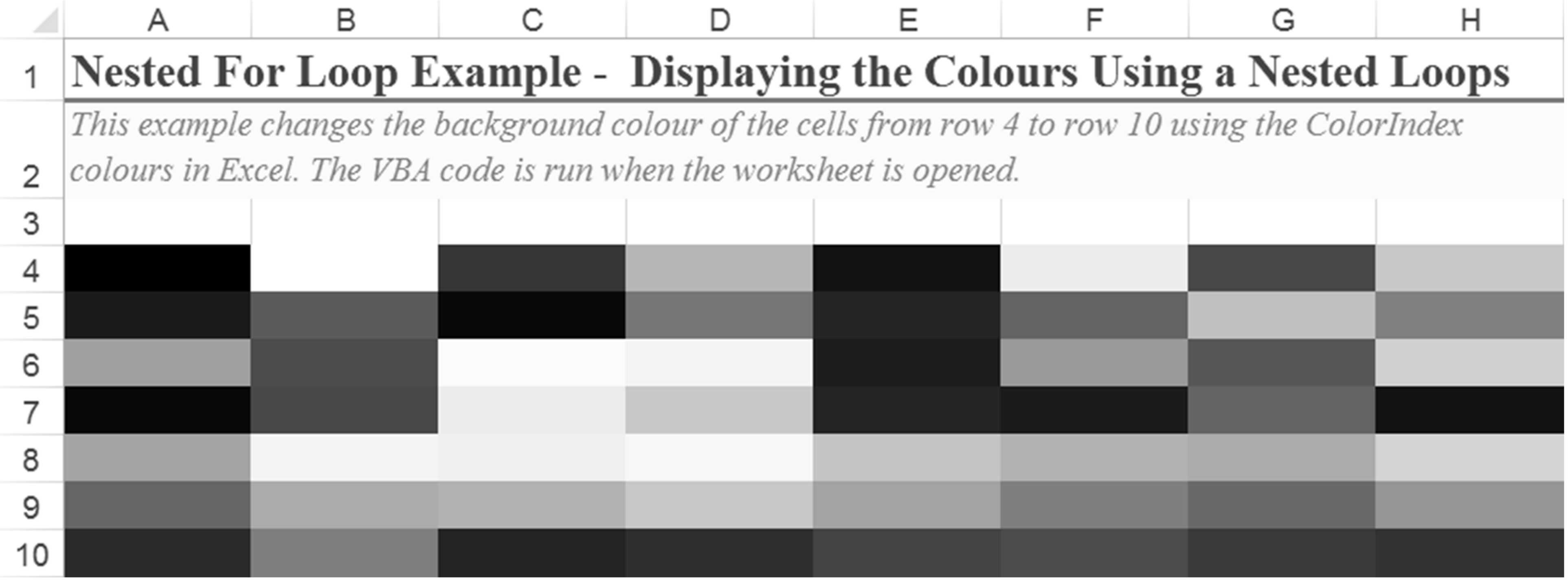
- You can change the colour of a cell by using a simple number which is called the ColorIndex
- In the following example, the background colour of cell A1 is changed to black by setting the ColorIndex to 1, and the background colour of cell B2 is changed to red by setting the ColorIndex to 3:

```
Range("A1").Interior.ColorIndex = 1
Range("B2").Interior.ColorIndex = 3
```



ColorIndex Numbers

- In the next example, we put the 56 colours into 7 rows so that the colours are arranged nicely like this:



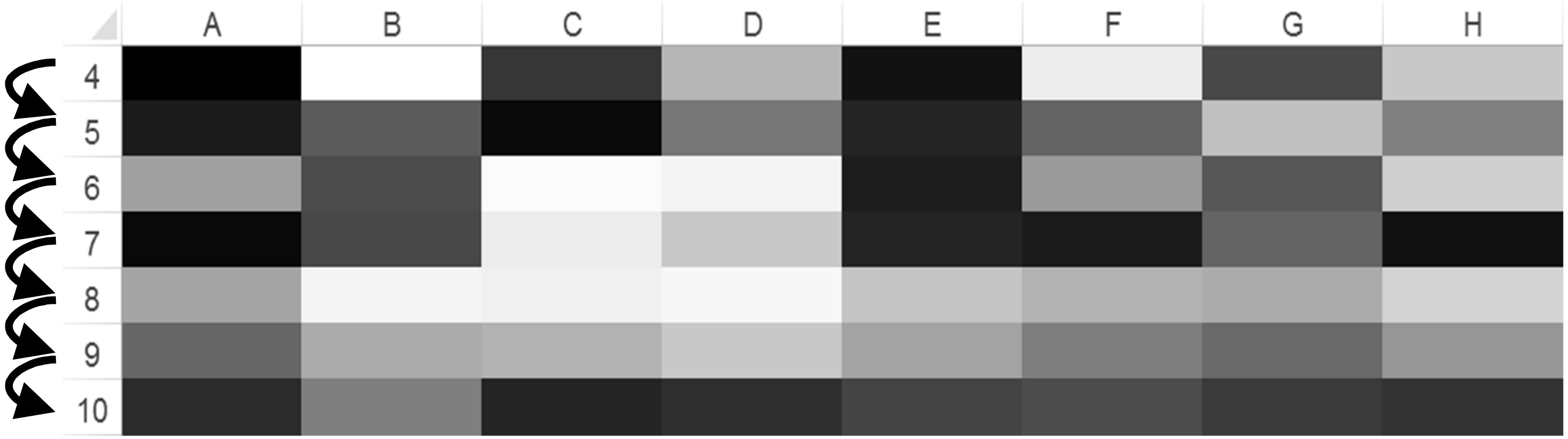
Nested For Loop Example - Displaying the Colours Using a Nested Loops

This example changes the background colour of the cells from row 4 to row 10 using the ColorIndex colours in Excel. The VBA code is run when the worksheet is opened.

ColorIndex Numbers – The Outer Loop

- We start with a loop that goes through the rows, i.e. from row 4 to row 10:

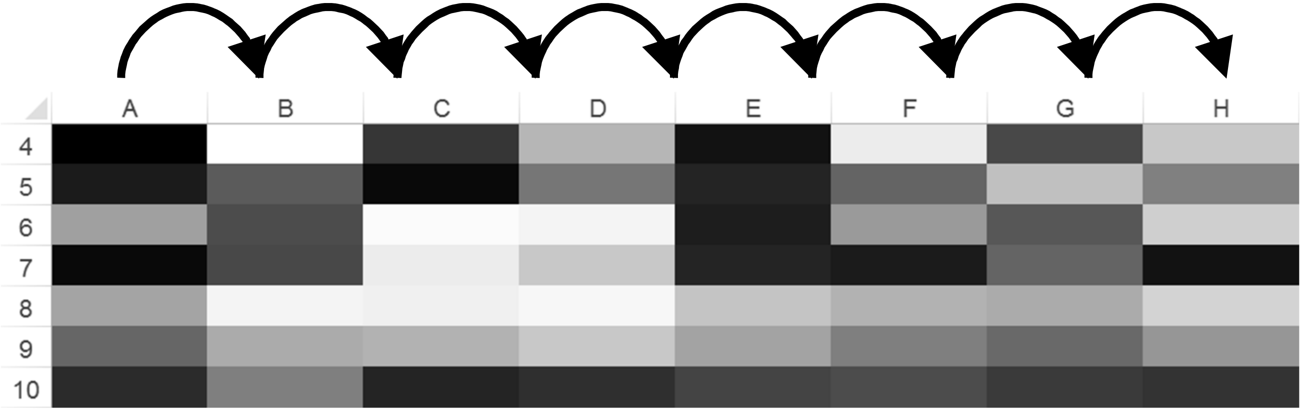
```
For Row = 4 To 10
    ...
Next Row
```



ColorIndex Numbers – The Inner Loop

- The loop going through the rows is the *outer loop*
- Since each row has 8 columns, we add an *inner loop* that goes through the 8 columns in the loop body
- The inner loop is then run for each row

```
For Row = 4 To 10
    For Col = 1 To 8
        ...
    Next Col
Next Row
```



ColorIndex Numbers – The Code

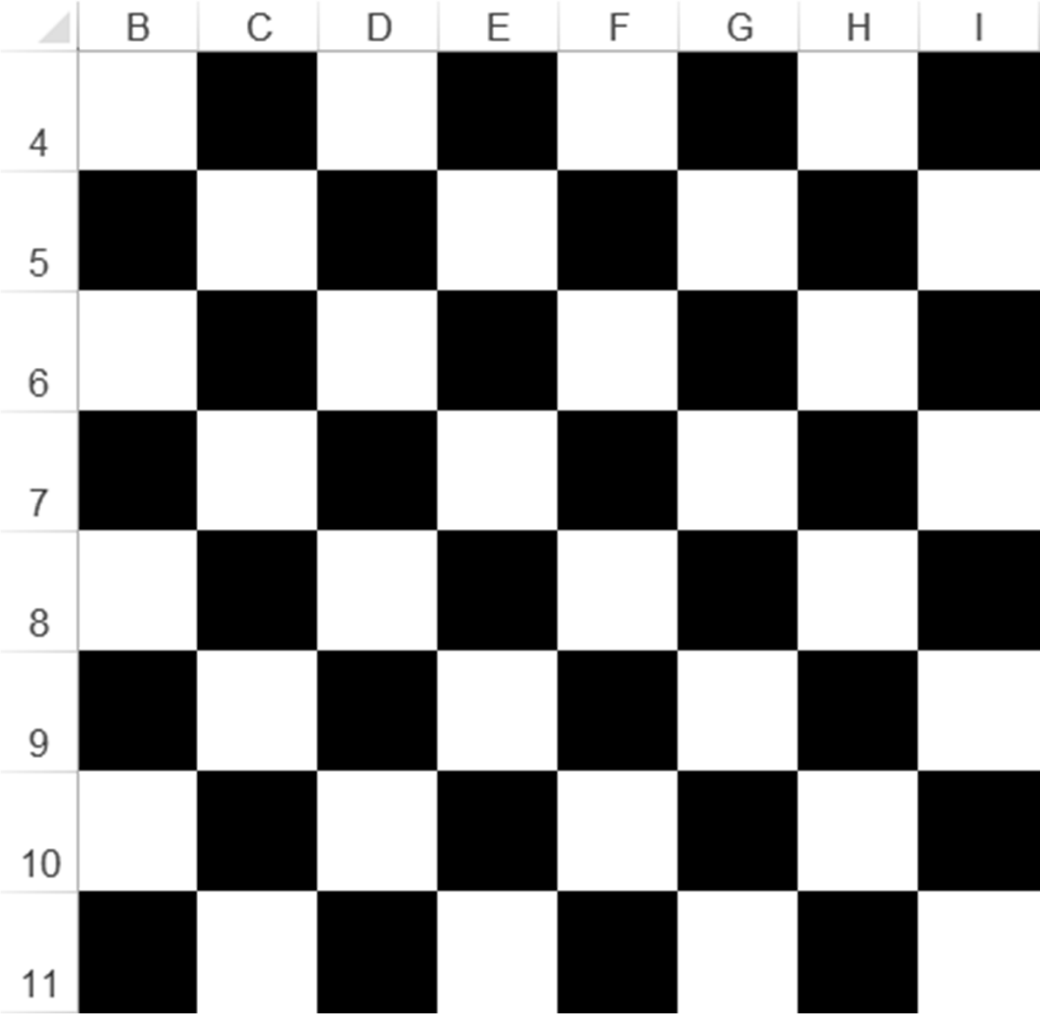
- The code we write inside the inner loop is executed on a cell in a particular row and column
- Inside the inner loop, we change the colour of the cells using the ColorIndex number

```
ColorIndex = 1
For Row = 4 To 10
   For Col = 1 To 8
      ' Change the background colour of the cell
      Cells(Row, Col).Interior.ColorIndex = _
                                   ColorIndex
      ColorIndex = ColorIndex + 1
   Next Col
Next Row
```

7 rows by 8 columns,
i.e. 56 cells

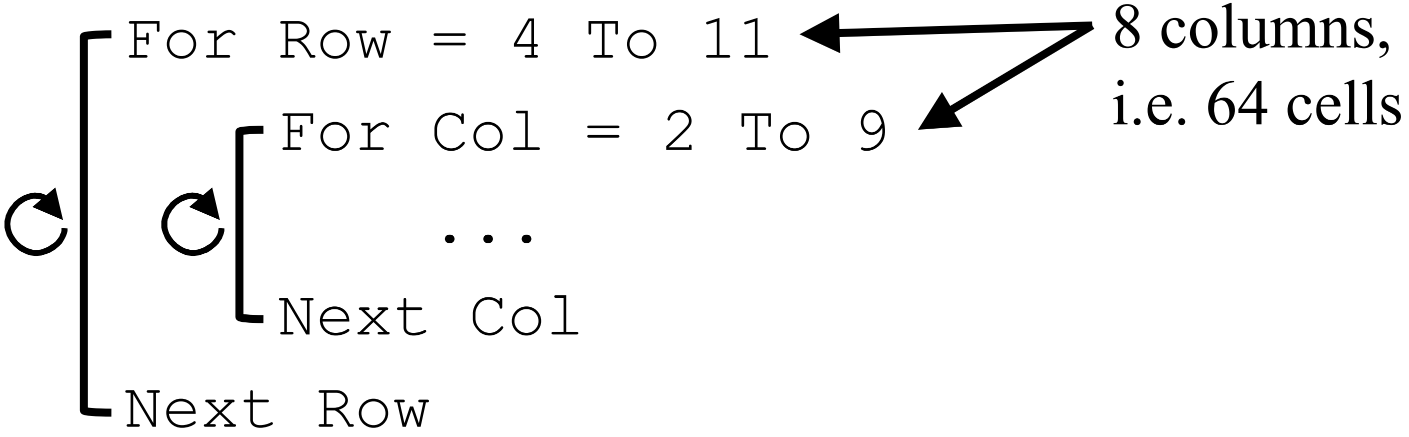
Drawing a Chess Board

- The next example draws a chess board, i.e.:
- The chess board uses 8 rows, with each row having 8 columns
- We can use a nested loop to walk through the cells



Drawing a Chess Board – The Loops

- The following loops walk through 8 rows and 8 columns, starting from cell B4



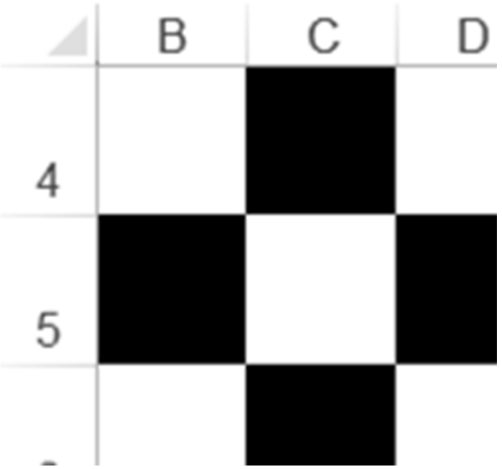
```
For Row = 4 To 11
    For Col = 2 To 9
        ...
    Next Col
Next Row
```

8 rows by
8 columns,
i.e. 64 cells

- This example again uses the ColorIndex number of the cells; setting them to white (=2) or black (=1)

Drawing a Chess Board – Colouring

- We decide what colour of a cell is based on the following observations
  - A cell is white when
    - both row and column are even numbers such as B4, or
    - both row and column are odd numbers such as C5
  - A cell is black when
    - row is even and column is odd such as C4, or
    - row is odd and column is even such as B5



Using Mod for Odd/Even Numbers

- If you divide an odd number by 2, you will always get a remainder of 1; if you divide an even number by 2, you will get a remainder of 0
- The `Mod` operator calculates the remainder of a division so the following expression can help you determine whether a number is odd or even:

```
Number Mod 2
```

- If the expression returns 0, `Number` is even; otherwise, `Number` is odd

Drawing a Chess Board – The Code

- Here is the complete code:

Both row and column are even numbers or odd numbers

```
For Row = 4 To 11
   For Col = 2 To 9
      If Row Mod 2 = Col Mod 2 Then
         ' Change the background to white
         Cells(Row, Col).Interior.ColorIndex = 2
      Else
         ' Change the background to black
         Cells(Row, Col).Interior.ColorIndex = 1
      End If
   Next Col
Next Row
```