VBA Cheat Sheet

Logic and Loops

Comparison Operators

=	Equal To
<>	Not Equal To
<	Less Than
>	Greater Than
<=	Less Than or Equal To
>=	Greater Than or Equal To

Logical Operators

- Logical operators return True or False based on the conditions you're evaluating.
- Great for pairing with If … Then statements.

AND	Both conditions must be true. `If a = 1 And b = 2 Then`
OR	Either condition (or both) must be true. `If a = 1 Or b = 2 Then`
XOR	One **and only one** condition must be true. `If a = 1 Xor b = 2 Then`
NOT	Negates a condition. The NOT keyword must appear in front. `If Not a = 1 Then`

If … Then Statements

- Runs a group of actions if certain conditions are met
- `ElseIf` and `Else` are optional
- `Else` must appear last

```
If [condition 1 is true] Then
    [do something]
ElseIf [condition 2 is true] Then
    [do something else]
Else
    [do something else]
End If
```

- Even if multiple conditions are true, VBA stops at the first true condition it finds.
- Can check multiple conditions at the same time with logical operators
- Can nest If … Then statements.

Nested Conditionals and Logical Operators

```
Sub IF_THEN_DEMO()
If IsNumeric(Range("A1")) And _
 Not IsEmpty(Range("A1")) Then
  If Range("A1") > 10 Then
    MsgBox "A1 is greater than 10"
  ElseIf Range("A1") > 5 Then
    MsgBox "A1 is between 5 and 10"
  ElseIf Range("A1") >= 0 Then
    MsgBox "A1 is between 0 and 5"
  Else
    MsgBox "A1 is negative"
  End If
Else
  MsgBox "A1 is not numeric"
End If
End Sub
```

For… Next Loops

- Quickly cycles through a series of values.
- Let's you repeat an action over and over.

```
Sub Practice_ForLoops()
Dim i as Integer
For i = 1 To 1000
    Range("A" & i) = i ^ 2
Next i
End Sub
```

Using Step

- Defines the increment for the counter
- Can be fractional, like 0.5
- Can step forward (+) or backward (-)

```
Sub Practice_ForLoops()
Dim i as Integer
For i = 5 To 50 Step 5
    Range("A" & i) = i ^ 2
Next i
End Sub
```

Nested Loops

```
Sub Practice_ForLoops()
Dim i as Integer
Dim j as Integer
For i = 1 To 1000
    For j = 5 to 10
        Cells(i, j) = i * j
    Next j
Next i
```

Exiting a For … Next loop early

- Exit loop early with `Exit For`.

```
Sub Practice_ForLoops()
Dim i as Integer
For i = 1 To 1000
    Range("A" & i) = i ^ 2
    If Range("A" & i) > 100 Then
        Exit For 'exits when i=11
    End If
Next i
End Sub
```

For … Each Loops

- Loop through each object in a group of objects

Loop Through Each Cell in Range

```
Sub FOR_EACH_DEMO()
Dim cell As Range
For Each cell In Range("A1:D5")
    cell = cell.Address
Next cell
End Sub
```

Loop Through Each Sheet in Workbook

```
Sub FOR_EACH_DEMO()
Dim Sh As Worksheet
For Each Sh In ActiveWorkbook.Sheets
    MsgBox Sh.Name
Next Sh
End Sub
```

Exiting a For … Each loop early

```
Sub FOR_EACH_DEMO()
Dim cell As Range
For Each cell In Range("A1:D5")
    cell = cell.Address
    If cell.Address = "$B$3" Then
        Exit For
    End If
Next cell
End Sub
```

Do … While Loops

- Continues to loop until the condition **is no longer true** (opposite of Do Until).
- Care must be taken to avoid an infinite loop.
- Can take 2 forms:

Check condition at the beginning

```
Do While [condition is true]
    [do something]
Loop
```

- If condition is already false, the contents of the loop will never be executed
- The MsgBox in this example never appears because i is already less than 0 before starting:

```
Sub DO_WHILE_DEMO()
i = 1
Do While i < 0
    MsgBox "hello"
Loop
End Sub
```

Check condition at the end

```
Do
    [do something]
Loop While [condition is true]
```

- When condition is checked at the end, the contents of the loop are always executed at least once, even if the condition is false the first time.
- The MsgBox in this example appears, even though i is less than 0 at the beginning:

```
Sub DO_WHILE_DEMO()
i = 1
Do
    MsgBox "hello"
Loop While i < 0
End Sub
```

Exiting a Do … While loop early

```
Sub DO_WHILE_DEMO()
i = 1
Do While i < 10
    If i = 5 Then
        Exit Do
    End If
    i = i + 1
Loop
End Sub
```

Do … Until Loops

- Continues to loop until the condition **is true** (opposite of Do While).
- Care must be taken to avoid an infinite loop.
- Can take 2 forms:

Check condition at the beginning

```
Do Until [condition is true]
    [do something]
Loop
```

- If condition is already true, the contents of the loop will never be executed

Check condition at the end

```
Do
    [do something]
Loop Until [condition is true]
```

- When condition is checked at the end, the contents of the loop are always executed at least once, even if the condition is true the first time.

Exiting a Do … Until loop early

```
Sub DO_UNTIL_DEMO()
i = 1
Do Until i > 10
    If i = 5 Then
        Exit Do
    End If
    i = i + 1
Loop
End Sub
```

While … Wend Loops

- Similar to Do … While with slightly different syntax

```
While [condition is true]
    [do something]
Wend
```

Can only check condition at the beginning

```
Sub WHILE_WEND_DEMO()
i = 1
While i < 10
    Range("A" & i) = i
    i = i + 1
Wend
End Sub
```

- The Do … While method is preferred since it's more structured and easier to understand.