Topic	Explanation in VBA/Excel	
Case sensitive	No, but editor will change your typing to standard upper and lower case notation for VBA	
Statement structure	Normally one statement per line, use space+underscore(_) to continue a statement to a new line. For multiple statements on one line, end initial statement(s) with a colon.	
Variable types	Long, Boolean, Double, Date, String, Variant	
Declaration statement	Dim <variable> As <type>, where <variable> is any variable and <type> is one of the available data types such as the ones listed above	
Symbolic Constants	Start with Const instead of Dim; Example: Const PI As Double = 3.14159265358979	
Operator Precedence	(Expressions in parentheses), ^, unary -, / and *, \, Mod, + and -, &, relational, Not, And, Or	
Relational operators	>, >=, =, <=, and < with usual meanings and not equal (<>)	
Logical operators	A **And** B is True if both A and B are true; A **Or** B is true if either A or B is true; **Not** A is true if A is false and is false if A is true	
Functions	**Function <name> (<arguments>) As <type>** <name> is the name of the function <type> is the data type for the function <arguments> may be blank or have one or more entries of the following forms <variable> As <type>	(multiple entries separated by commas) <variable> is a variable used in the function <type> is the data type for that variable Arguments provide input data to function Set function name equal to return value
Subs	Sub <name>(<arguments>) : <statements> : End Sub See functions for definition of <arguments>. A macro is a sub with no arguments.	

If statements
These may have zero or many ElseIf's.
The Else statement is optional
	If <condition1> Then
	<Done if condition1 is true>
	ElseIf <condition2> Then
	<Done if condition2 is true>
	ElseIf <condition3> Then
	<Done if condition3 is true>
	(continued in next column)
		<May be other ElseIf conditions>
		Else
		<Done if all conditions false>
		End If
		<Execute here after any statements done>
		Note: Else statement is optional

Count-controlled loops	For <counter> = <start> To _ <end> Step <increment> <statements> Next <counter>	**Alternative if <increment> = 1** For <counter> = <start> To <end> <statements> Next <counter>
Conditional loops	Do While <condition> <statements> Loop	Do <statements> Loop Until <condition>
Strings	Dim s as String : s = "This is a string"	
Concatenation	s = "This is a " & "string" (can use + instead of &)	
Format function	<string expression> = Format(<value>, <format string>)	
Format specifications used in <format string>	**0** place a digit **even if** it is a leading or trailing zero **#** place a digit **unless** it is a leading or trailing zero **%** multiply number by 100 and create percentage	**.** place decimal point here **E** use E format for output
Declaring Arrays	Dim <array name>(<array dimensions>) as <type>, where dimensions can be a single number or a range like 5 To 12; use two dimensions for a 2D array.	
Lowest array subscript	0 by default, default can be changed to 1 by Option Base 1 statement and can be set to any value for individual arrays	
Array Examples The default lower array index is zero. This can be set to one by an Option Base 1 statement or set for individual arrays.	Dim x(0 To 200) As Double Dim y(0 To 200) As Double Dim z(55 To 180) As Double For k = 0 To 99 x(k) = k y(k) = sin(PI*k/100) x(k+100) = x(k)*y(k) Next k	Dim I(1 To 10) as Double Dim V(1 To 10) as Double Dim P(1 To 10, 1 To 10) As Double For k = 1 To 10 For m = 1 To 10 P(m,k) = I(m) * V(k) Next m Next k
A few objects	x = Range(<cell>).Value Range(<cell>).Value = y x = Cells(<row>,<col>).Value Cells(<row>,<col>).Value = y	<range specification>.NumberFormat = <range specification>.ClearContents Worksheets(<name>).<range specification>