

Wörterbuch VBA – Deutsch

&	Operator (wie „+“, „*“ etc.), der beliebige Dinge zu einem Text verknüpfen kann	<pre>s = "Pi hat den Wert " pi = 3.1415 MsgBox (s & pi)</pre>
=	Zuweisungsoperator. Funktioniert immer nach dem selben Prinzip: rechte Seite ausrechnen, Ergebnis in linker Seite speichern. Damit kann man auch den aktuellen Wert einer Variablen verändern. Nicht mit mathematischer Gleichung verwechseln!	<pre>x = 1 x = x + 1</pre>
Call	ruft ein mit → Sub definiertes Makro auf	<pre>Call f_ausgeben (a,b,4)</pre>
Debug.Print	gibt etwas in einem kleinen Dialogfenster aus. Das kann ein Text, eine Zahl, das Ergebnis einer Berechnung oder auch ein längerer Text, der mit → & aus Einzelteilen zusammengesetzt wurde, sein Um statt in einem Dialogfenster die Ausgabe in den Direktbereich zu schreiben, ist → MsgBox der richtige Befehl	<pre>betrag = 19.99 waehrung = "EUR" s = betrag & " " & waehrung Debug.Print (s)</pre>
Dim	legt die Art einer Variablen fest. Wir verwenden sie nur für Feldvariablen. Es ist dabei nötig, den „Typ“ der Variable anzugeben, meist → Integer (ganzzahlig), → Single (Zahl mit Nachkommastellen) oder → String (Texte)	<pre>Dim s(5) As Integer For i = 1 To 5 z = z + s(i) Next i Dim w(3) As Single</pre>
Do	leitet die Schleife ein, die nach den Befehlen prüft, ob die Bedingung erfüllt ist. Im Gegensatz zur → While -Schleife. Am Ende steht die Bedingung hinter → Loop While .	<pre>z = 0 i = 1 Do z = z + i i = i + 1 Loop While i < 5</pre>
Else	leitet den alternativen Fall bei einer Fallunterscheidung ein → If	
ElseIf	gehört zur Fallunterscheidung mit → If und wird nur benötigt, wenn mehr als zwei Fälle zu unterscheiden sind. Achtung: ElseIf wird zusammengeschrieben	<pre>If x < 0 Then Befehle für x<0... ElseIf x > 0 Then Befehle für x>0... Else Befehle für x=0 End If</pre>
End Function	kennzeichnet das Ende eines mit → Function definierten Funktion	
End If	beendet eine Fallunterscheidung, → If	
End Sub	kennzeichnet das Ende eines mit → Sub definierten Makros	
For	leitet die For-Schleife ein und benötigt immer eine Zählvariable (oft i, j), die vom Startwert bis zum Endwert „durchgezählt“ wird. Was zwischen „For“ und „Next“ steht, wird dann für jedes i ausgeführt.	<pre>For i = 1 To 10 z = z + i Next i</pre>

Function	<p>Leitet die Definition einer Funktion ein. Funktionen kann man (anders als Makros) nicht starten, sondern nur aufrufen. Sie arbeiten ähnlich wie mathematische Funktion $f(x) = \dots$, und sie erwarten fast immer Argumente (x-Werte), für die dann ein Ergebnis berechnet wird.</p> <p>Wichtig: Das Ergebnis wird zurückgegeben, indem am Ende der Funktionsdefinition ein Befehl steht, der wie eine Variablenzuweisung aussieht; auf der linken Seite steht dabei der Name der Funktion.</p> <p>Funktionen enden immer mit \rightarrow End Function, der VBA-Editor fügt diesen Befehl automatisch ein, wenn Sie eine neue Funktion erstellen.</p>	<pre>Function quadrat(x) quadrat = x * x End Function Function maximum(x,y) If x > y Then ergebnis = x Else ergebnis = y End If maximum = ergebnis End Function maximum(3,4) → 4 quadrat(5) → 25</pre>
If	<p>leitet eine Fallunterscheidung ein. Gleich nach dem If kommt eine Bedingung und dahinter das Schlüsselwort \rightarrow Then. In den folgenden Zeilen stehen dann die Befehle, die auszuführen sind, wenn die Bedingung zutrifft („Ja-Fall“). Danach kann die Fallunterscheidung mit \rightarrow End If beendet werden, falls im „Nein-Fall“ nichts zu tun ist. Falls doch, geht es mit \rightarrow Else (andernfalls) weiter. Sollen mehr als zwei Fälle unterschieden werden, hilft das Schlüsselwort \rightarrow ElseIf weiter</p>	<pre>Eingabe = InputBox() If Eingabe = 1 Then MsgBox ("1 gewählt") Else MsgBox ("was anderes gewählt") End If</pre>
InputBox	<p>öffnet ein kleines Dialogfenster, das die Eingabe eines Wertes erlaubt. Wird meist zusammen mit einer Zuweisung (\rightarrow =) verwendet. Als Argument kann man einen Hinweistext angeben</p> <p>Für die Ausgabe in einem Dialog ist \rightarrow MsgBox zuständig</p>	<pre>s = InputBox ("Bitte s eingeben")</pre>
Integer	<p>Typ einer Variablen, die nur ganzzahlige Werte (z. B. -3, 5, 2) annehmen kann</p>	<pre>i = 3</pre>
Loop While	<p>beendet die \rightarrow Do-Schleife</p>	
MsgBox	<p>gibt etwas in einem kleinen Dialogfenster aus. Das kann ein Text, eine Zahl, das Ergebnis einer Berechnung oder auch ein längerer Text, der mit \rightarrow & aus Einzelteilen zusammengesetzt wurde, sein</p> <p>Um statt in einem Dialogfenster die Ausgabe in den Direktbereich zu schreiben, ist \rightarrow Debug.Print der richtige Befehl.</p> <p>Für Eingaben ist die \rightarrow InputBox zuständig</p>	<pre>betrag = 19.99 waehrung = "EUR" s = betrag & " " & waehrung MsgBox (s)</pre>
Next	<p>Beendet die \rightarrow For-Schleife. Oft gibt man hier nochmal die Variable an („Next i“), es geht aber auch ohne („Next“)</p>	
Single	<p>Typ einer Variablen, die Werte mit Nachkommastellen (z. B. 3.1415, 5.0, 2.121212) annehmen kann. Achtung: In VBA-Programmen muss der Dezimalpunkt verwendet werden, nicht das deutsche Dezimalkomma.</p>	<pre>p = 3.1415 r = 4.0 flaeche = p * r * r</pre>
String	<p>Typ einer Variablen, die nur Texte (z. B. "Bitte Wert eingeben") annehmen kann. Ein String wird immer in Anführungszeichen eingeschlossen</p>	<pre>s = "Hallo" MsgBox (s)</pre>

