



5 DATEN

5.1. Variablen

Variablen können beliebige Werte zugewiesen und im Gegensatz zu **Konstanten** jederzeit im Programm verändert werden.



Als Variablen können beliebige Zeichenketten verwendet werden. Das können einzelne Buchstaben aber auch längere Texte sein. Sinnvoll sind aber aussagekräftige Bezeichnungen.

Beispiele

Q = 12

R = 3

S = Q * R

ZwischenWert = 3456

Range("B5").Value = ZwischenWert

Deklarieren

Übungen:

Liter in Kg..... 113

Tabellenblätter
sortieren 135

Dateiliste
erstellen..... 138

Eine Deklaration ist nicht erforderlich aber sinnvoll, da standardmäßig der Typ Variant verwendet wird der unter Umständen mehr Speicherplatz benötigt, bzw. bei der Verarbeitung etwas langsamer ist.

Dim Q, R, S

Sinnvoll ist es bei der Deklaration den Datentyp festzulegen.

Variablen werden in dieser Form deklariert

Dim Zahl As Double

Dim Zaehler As Integer

Dim Brutto As Currency, Netto As Currency

Dim Text As String



Mehrere Variable können Sie hintereinander in eine Zeile schreiben. In diesem Fall ist jedoch nur die Variable *Drei* als Integer deklariert die anderen sind Variant

```
Dim Eins, Zwei, Drei as Integer
```

Auch bei der Schreibweise in einer Zeile müssen Sie bei jeder Variablen den Typ angeben.

```
Dim VName As String, NName As String
```

Deklaration erzwingen

Sie können festlegen, dass Variablen grundsätzlich deklariert werden müssen, in dem Sie vor der ersten Routine in Ihr Modul diese Option schreiben:

```
Option Explicit
```

Damit können Variablen nur verwendet werden, wenn diese vorher deklariert wurden. Das hat den Vorteil, dass Sie auf Fehler durch falsche Schreibweise hingewiesen werden.



5.2. Datentypen



In VBA sind Variablen die nicht besonders deklariert sind vom Typ Variant. Das Deklarieren bringt Vorteile so können Berechnungen mit Integer-Zahlen schneller ausgeführt werden als mit Zahlen vom Typ Double. Außerdem haben die unterschiedlichen Datentypen unterschiedlichen Speicherbedarf.

Datentypen

Datentyp	Byte	Wertebereich
Integer	2	Ganzzahl von -32.768 bis 32.767
Long	4	32-Bit-Ganzzahl (-2.147.483.646 bis 2.147.483.467)
Decimal	8	96-Bit-Ganzzahl (+/- 79.228.162.514.264.337.593.543.950.335)
Single	4	Gleitkommazahl einfacher Genauigkeit (+/- 3,402823E38)
Double	8	Gleitkommazahl doppelter Genauigkeit (+/- 1,79769313486232E308)
Currency	8	Gleitkommazahl für Währungsberechnungen von - 922.337.203.685.447,5808 bis 922.337.203.685.447,5807
Date	8	Datum von 1.1.100 bis 31.12.9999
String	10 + Länge	Zeichenkette mit ca. 2 Milliarden Zeichen
Variant	mind. 16	Zeichenketten und numerische Werte im Bereich des Datentyps Long



5.3. Datenfelder

Diese werden auch als **Arrays** bezeichnet. Es handelt sich dabei um Gruppen von Variablen. Der Einsatz ist vor allem dann angebracht, wenn mehrere gleichartiger Variablen benötigt werden.



Sie wollen z. B. die Wochentage als Variablen festlegen, dann wäre dies damit möglich.

Datenfelder werden genauso verwendet und deklariert wie Variablen, nur das hinter dem Namen der Variable eine Indexzahl angegeben wird.

```
...Dim WT(7) As String
```

Das bedeutet, dass dem Datenfeld WT sieben Felder zur Verfügung stehen. Anstelle von **Dim** können auch die Schlüsselwörter **Private**, **Public** und **Static** verwendet werden.

Die Werte werden genau so zugewiesen wie bei Variablen:

```
WT(1) = "Montag"  
WT(2) = "Dienstag"  
WT(3) = "Mittwoch"  
..usw.
```

Die Werte können nun sehr einfach mit einer Schleife an beliebiger Stelle eingetragen werden, dabei kann für den Index auch eine Variable verwendet werden.

```
Sub Wochentage_eintragen()  
    Dim i As Integer  
    For i = 1 To 7  
        Cells(2, i) = WT(i)  
    Next i  
End Sub
```

Mehrdimensionale Datenfelder

```
Dim ZweiDimensional(50, 10) As Integer  
Dim DreiDimensional(12, 31, 20) As Integer
```



5.4. Konstanten



Diese können im Gegensatz zu Variablen nur im Code festgelegt und während des Programmablaufes **nicht** verändert werden.

Als Namen können beliebige Zeichenketten verwendet werden. Das können einzelne Buchstaben aber auch längere Texte sein. Sinnvoll sind aussagekräftige Bezeichnungen.

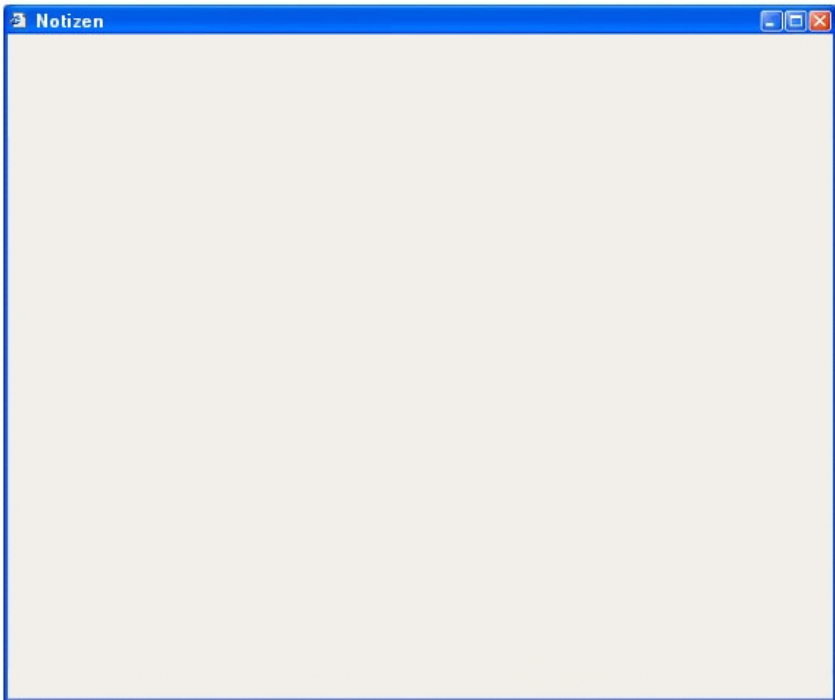
Auch für Konstanten sollten Sie den Variablentyp festlegen.

Eine Konstante wird im Modulkopf, also außerhalb von *Sub ... End Sub*, mit dem Schlüsselwort *Const* festgelegt

Eigene Konstanten

```
Const EUR = 1,95583
```

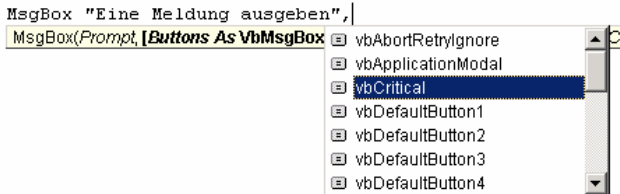
```
Const Version As String = "1.02.01"
```





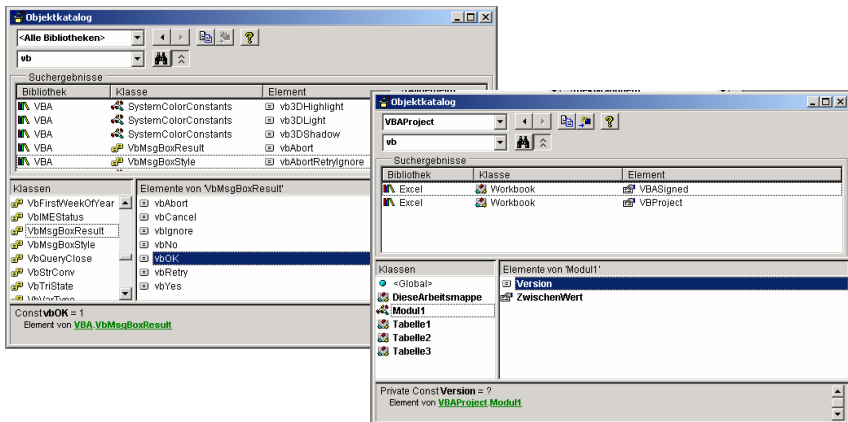
Eingebaute Konstanten

In VB sind einige Konstanten bereits festgelegt. Diese werden z.B. als Argumente beim Erzeugen von Steuerelementaufrufen in einer Liste vorgeschlagen und brauchen nur ausgewählt werden.



Objektkatalog

Dieser kann mit der (F2) Taste aufgerufen werden. Hier sehen Sie sowohl die eingebauten Konstanten als auch die von Ihnen festgelegten.





5.5. Variablen Gültigkeit

Lokale Variablen



In der Regel sind Variablen **lokal** d. h. sie sind nur innerhalb einer Routine oder Funktion gültig. Das heißt, wenn eine Routine verlassen wird erlischt der Wert der Variablen. Das ist sehr nützlich, da ein gleicher Variablenname in verschiedenen Routinen und Funktionen verwendet werden kann und diese sich gegenseitig nicht beeinflussen. Lokale Variablen werden innerhalb einer Prozedur oder Funktion festgelegt.

Private Variablen

Wird eine Variable im Modul außerhalb einer Routine deklariert so steht sie allen Routinen in diesem Modul zur Verfügung und braucht beim Aufruf nicht übergeben werden. Statt dem Schlüsselwort *Dim* kann hier auch *Private* verwendet werden. Letzteres sollten Sie auch verwenden um klar zu stellen, dass es sich nicht um eine lokale Variable handelt.

```
Dim Allgemein As Integer
Private Jeder As Long
```

Öffentliche Variablen

Im Unterschied zu privaten Variablen stehen öffentliche Variablen allen Modulen einer Arbeitsmappe zur Verfügung. Öffentliche Variablen werden im Modul mit dem Schlüsselwort *Public* deklariert.

```
Public General As Integer
```

Static Variablen

Dies sind lokale Variablen, die allerdings im Gegensatz zu normalen Variablen ihren Wert behalten wenn die Routine erneut aufgerufen wird. Das Schlüsselwort ist *Static*.

```
Static Betrag As Long
```

Die Variable wird erst zurückgesetzt, wenn die Mappe geschlossen wird, oder im VBA-Editor **Ausführen / Beenden** bzw. **Ausführen / Zurücksetzen** gewählt wurde.



Gültigkeit Demonstration

Durch unterschiedliche Deklaration der Variablen steht ein Wert nur in der ersten Sub zur Verfügung der zweite in allen Modulen.

Werden Variablen als *Private* deklariert, so stehen sie in anderen Routinen des gleichen Moduls zur Verfügung. Wenn die Variablen jedoch in allen Modulen zur Verfügung stehen sollen, so müssen sie als *Public* deklariert werden.

```

VariablenGültigkeit.xls - Modul1 (Code)
(Allgemein) | Anzeige_Priv
Private Wert_priv As Long
Public Wert_publ As Long

Sub Aufruf()
    Wert_priv = 100
    Wert_publ = 500
    Call Anzeige_Priv
    Call Anzeige_Publ
End Sub

Sub Anzeige_Priv()
    MsgBox "priv. " & Wert_priv & ", publ. " & Wert_publ
End Sub

VariablenGültigkeit.xls - Modul2 (Code)
(Allgemein) | Anzeige_Publ
Public Sub Anzeige_Publ()
    MsgBox "priv. " & Wert_priv & ", publ. " & Wert_publ
End Sub
    
```

Um die Gültigkeit der Variablen zu testen ist es unbedingt nötig, dass Sie die Routinen in zwei Modulen erstellen, da Sie sonst nicht den Unterschied zwischen den einzelnen Gültigkeitsarten testen können.

Deklariieren Sie einmal alle Variablen als Public und testen dann die Routine und achten Sie darauf, wie sich das Ergebnis ändert:

- 1 Deklarieren Sie im **Modul1** je eine Private Variable namens *Wert_priv* und eine Public Variable namens *Wert_publ*.
- 2 Erzeugen Sie ebenfalls im **Modul1** eine Routine mit dem Namen *Aufruf* die den Variablen *Wert_priv* und *Wert_publ* die 100 und 500 zuwiesnen und danach die Routinen **Anzeigen_eins** und **Anzeigen_zwei** aufruft.
- 3 Anschließend erzeugen Sie eine weitere Routine im **Modul1** mit dem Namen **Anzeigen_priv** in der die zwei Variablen in einer MsgBox ausgegeben werden.
- 4 Erstellen Sie nun im **Modul2** die Routine mit dem Namen **Anzeigen_publ**.
- 5 Beim Ausführen erhalten Sie nacheinander diese Fenster angezeigt.

