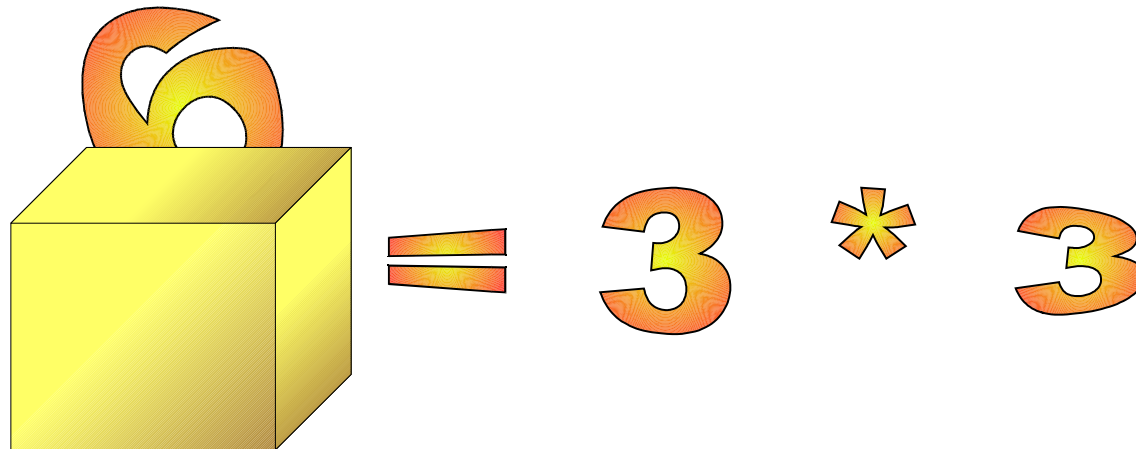


Excel – VBA

Variablen und Konstanten



Anweisungen

- Pro Zeile steht eine Anweisung, die aus einem Ausdruck gebildet wird.
- ... werden meist Zeile für Zeile ausgeführt.
- ... symbolisieren eine bestimmte Aktion, die der Computer ausführen soll.
- ... ist die kleinste Einheit in einem VBA-Programm.
- Beispiele:
 - `Range("A1").Value = 2`
 - `rabatt = preis * 0.01`
 - `summe = preis * menge`
 - `button = MsgBox("Eingabe von ...")`

Ausdruck

- ... besteht aus Operanden und Operatoren, die nach bestimmten Regeln zusammengesetzt werden.
- ... formuliert einzelne Schritte eines Programms.
- ... ist eine Verarbeitungsvorschrift, die einen Wert als Ergebnis liefert.
- ... verändert den Wert von Variablen entsprechend des angegebenen Datentyps.
- Beispiele:
 - Arithmetische Berechnung: $\text{preis} * 0.16$
 - Vergleichsoperatoren nutzen: $\text{messpunkt} > 0$
 - Ausdrücke miteinander verknüpfen: $(a \geq b) \text{ AND } (a \geq c)$
 - Prozeduren aufrufen: $\text{addition}(1,2)$

Operanden und Operatoren

- Operanden können
 - ... Variablen sein. Der Wert ist mit Hilfe der Tastatur oder durch das Programm veränderbar.
 - ... Konstanten sein. Der Wert ist unveränderbar.
- Operatoren
 - Arithmetische Operatoren berechnen Werte.
 - Vergleichsoperatoren vergleichen zwei Werte.
 - Logische Operatoren verknüpfen Werte oder Ausdrücke.


Kommentare

- ... erleichtern das Verstehen eines Programms.
- ... bieten Informationen für den Entwickler.
- ... müssen bei Codeänderungen oder Programm-Erweiterungen angepasst werden.
- ... werden vom Computer überlesen.
- ... beziehen sich auf das Warum und nicht so sehr auf das Wie.
Beispiele:
 - Warum wird an dieser Stelle diese Berechnung durchgeführt?
Nicht: Wie werden die Werte berechnet?
 - Warum wird der String an diesem Zeichen getrennt? Nicht:
Wie wird der String getrennt.

Beispiel

```
Sub NewTabellenblatt()  
    ' Erstellung eines Arbeitsblatts für neue Umsatzzahlen  
  
    ThisWorkbook.Activate  
    Worksheets.Add  
    ActiveSheet.Name = "UmsatzQuartal_2_2009"  
  
    ActiveSheet.Cells(1, 1).Value = "Niederlassung"  
    ActiveSheet.Cells(1, 2).Value = "Umsatz"  
  
End Sub
```

Code auskommentieren

- Jeder Kommentar beginnt mit dem Apostroph. Alle Zeichen hinter dem Apostroph werden vom Computer nicht gelesen.
- Mehrere hintereinander folgende Zeilen können mit Hilfe des rechten Symbols  in der Symbolleiste Bearbeiten auskommentiert werden. Das linke Symbol hebt die Kommentierung auf.

Kommentare platzieren

- Kommentare am Anfang eines Moduls platzieren:
 - Informationen zur Historie des Moduls.
 - Beschreibung des Inhalts des Moduls in Stichpunkten.
- Kommentare oberhalb oder unterhalb des Prozedurkopfes (Sub...) platzieren:
 - Welche Funktionalität wird abgebildet?
 - Ein- und Ausgabeparameter der Prozedur.
 - Wer hat die Prozedur geschrieben und wann geändert?
- Kommentare oberhalb oder rechts von einer Anweisung platzieren:
 - Warum wird diese Anweisung gerade ausgeführt?
 - Welches Resultat wird erwartet?

Überflüssige Kommentare

- Kommentare sollten offensichtliche Dinge nicht in Prosa wiederholen.
Ein negatives Beispiel: $a = c + b$; ' a ist die Summe von c und b
- Kommentare, die dem Programmcode widersprechen, sind redundant.

Konstanten

- ... sind fest mit einem bestimmten Wert verankert.
- ... besitzen einen Wert, der innerhalb ihres Gültigkeitsbereichs nicht verändert werden darf.
- ... sind Platzhalter für einen bestimmten Wert in einem Modul oder in einer Prozedur.
- ... werden auch als Literale bezeichnet.
- Der Name einer Konstanten ist frei wählbar. Ein Konstantenname sollte mit dem Unterstrich oder einem Großbuchstaben beginnen.

Konstanten definieren

```
Const Faktor As Integer = 100  
Const Messwert As Single = 3.4  
Const Ausgabe = "Hello World  
Const Richtig = False
```

- ... werden durch das Schlüsselwort Const gekennzeichnet.
- ... müssen gleichzeitig deklariert und initialisiert werden.

Konstanten deklarieren

Const

Das Schlüsselwort leitet eine Konstanten-deklaration ein.

Faktor

Der Name der Konstanten ist frei wählbar.

As Integer

Die Angabe des Datentyps ist optional. Der Datentyp beschreibt die Art des konstanten Wertes.

... plus Initialisierung

Const Faktor As Integer = 100

- Der Wert rechts vom Gleichheitszeichen wird der Konstanten zugewiesen.
- Die Konstante Faktor ist ein Platzhalter für den Wert 100.
- Der Wert der Konstanten darf innerhalb des Programms nicht verändert werden.

Datentyp und Wert der Konstanten

Const Faktor As Integer = 100

- Der Datentyp beschreibt die Art des Wertes.
- Der Datentyp muss zu dem angegebenen Wert passen.

Const Faktor = 100

- Der Datentyp wird automatisch in Abhängigkeit des Wertes festgelegt.

Systemdefinierte Konstanten

```
ActiveSheet.Range("A8").Borders.LineStyle = xlDot  
ActiveSheet.Range("A8").Borders.Color = vbRed
```

- ... sind von VBA oder der Anwendung vordefiniert.
- Die ersten zwei Buchstaben weisen auf die Herkunft der systemdefinierten Konstanten hin. Beispiele:
 - vb... , definiert von der Sprache VBA.
 - xl... , definiert von Excel.
 - mso... , definiert von Microsoft Office.
 - fm... , definiert für Formulare und Steuerelemente.

Variablen

- Zahlen und Zeichen werden in Variablen gespeichert.
- ... werden durch das Programm verarbeitet.
- ... können durch Ausdrücke verändert werden.
- ... können mit Hilfe der Tastatur ein Wert zugewiesen bekommen.
- ... sind Platzhalter für einen bestimmten Typ von Wert.

Beispiele aus der realen Welt

- Zutaten in einem Kochrezept. Die Menge der Zutaten ändert sich in Abhängigkeit der Anzahl der Personen.
- Das Waschpulver beim Waschen der Wäsche. Je nach Menge und Verschmutzungsgrad wird mehr oder weniger Pulver benötigt.
- Für die Suche von Wörtern in einem Text, wird der zu durchsuchende Text, das zu suchende Wort und ein Positionszähler benötigt.

Bestandteile

- Der Variablenname
 - ... symbolisiert eine Speicherstelle.
 - ... ist ein Platzhalter für einen bestimmten Wert.
- Der Datentyp der Variablen
 - ... legt das Format für den Wert sowie
 - ... den Speicherbedarf fest.

Variablen deklarieren

Dim

Wer hat Zugriff auf die Variable?
In diesem Beispiel hat nur der Block, in dem sie deklariert ist, Zugriff.

faktor

As Single

Der Name der Variablen ist frei wählbar.

Variablennamen

- ... müssen mit einem Buchstaben beginnen.
- ... sollten nur die Buchstaben A...Z, a...z, die Zahlen 0...9 und den Unterstrich enthalten
- ... müssen aus einer zusammenhängenden Zeichenfolge bestehen. Das heißt, Leerzeichen zur Trennung von Wörtern kommen in Variablennamen nicht vor.
- ... sind einzigartig. Sie kommen nur einmal in ihrem Gültigkeitsbereich vor.
- Als Bezeichnung dürfen keine Schlüsselworte aus VBA genutzt werden.
- ... sollten nicht länger als 32 Zeichen sein. Bezeichnungen können maximal 255 Zeichen besitzen.

Wahl eines Namens

- Der Variablenname sollte über die Art und Nutzung des Wertes Auskunft geben.
 - Beispiel: Für die Berechnung eines Kreisradius wird eine Variable mit dem Namen radius erzeugt.
 - Ungeeignete Variablennamen sind x2, a22 oder ähnlich kryptische Bezeichner.
- Ein Variablenname sollte den Sachverhalt, den die Variable repräsentiert, vollständig und genau beschreiben. Auf diese Weise ergeben sich oft bereits gute Namen.
- Gute Namen sind einfach zu lesen, da sie keine geheimnisvollen Abkürzungen enthalten und eindeutig sind.
- Verzichten Sie auf Namen, die keinerlei Beziehung zum Inhalt der Variablen erkennen lassen.

Wahl eines Namens

- Variablennamen sollten keine Verben als Bezeichnung nutzen. Verben werden in Funktions- oder Methodennamen genutzt. Sie beschreiben eine Aktion.
- Variablennamen sollten immer aus Substantiven bestehen. Variablen beschreiben ein Objekt oder ein bestimmtes Attribut eines Objekts. Die Bezeichnung sollte sich an die reale Welt anlehnen.
- Variablennamen, die nur aus einem einzelnen Zeichen bestehen, werden nur als Zähler oder Index genutzt.

Konventionen

- Variablennamen sollten im Gegensatz zu Konstanten immer mit einem Kleinbuchstaben beginnen.
- Variablennamen werden teilweise aus mehreren Namen zusammengesetzt. In der ungarischen Notation beginnt der Variablenname mit einem kleinen Buchstaben. Alle Teilnamen beginnen mit einem Großbuchstaben.
 - Beispiel: `lineBuffer`, `nErrFlag`.
 - Andere Möglichkeit: `line_buffer`, `n_err_flag`.
- Als Präfix (erste Buchstabe des Variablennamens) wird häufig eine Abkürzung für den Datentyp genutzt. Beispiel:
 - `booAusdruck` für einen boolschen Ausdruck,
 - `intKM` für eine Integer-Variable oder
 - `curPreis` für einen Währungswert.

Beispiele für Namen

Aufgabe	Beispiele für Variablennamen:	Schlecht verständliche Variablennamen:
Bestellnummer	bestellNr, bestellnummer bestellung_Nr	bnr bstnr bn
Rechnungsdatum	rechnungDatum bonDatum	rgD rechnung RechnungMaerz
Kundenname	kunde kundeName NameKunde	bezeichnung knd name
Farbe eines Autos	farbeAuto farbe_Auto	faAu AutoGelb au_fa Farbe Eines Autos

Ähnliche Zeichen

- Finden Sie die Variable, die nicht mit den zwei anderen Bezeichnung übereinstimmt.

CONFUSION

C0NFUSION

CONFUSION

hard2read

hardZread

hard2Read

GRANDTOTAL

GRANDTOTAL

GRANDT0TAL

- Folgende Zeichen sind sich sehr ähnlich:
 - 1 und l (Kleines L)
 - l und I (kleines L und großes L)
 - 1 und I (1 und kleines L)
 - 0 und O
 - 2 und Z
 - S und 5
 - G und 6

Variablen deklarieren

Dim

Wer hat Zugriff auf die Variable?
Nur der Block, in dem sie deklariert ist, hat Zugriff.

faktor

Der Name der Variablen ist frei wählbar.

As Single

Der Datentyp legt die Art des zu speichernden Wertes und deren Speicherbedarf fest.

Datentypen

- ... sind Baupläne für die Art der Variablen.
- ... geben über das Format eines Wertes, der in einer Variablen gespeichert wird, Auskunft.
- ... legen Regeln für die Interpretation und Verwendung eines Wertes fest.
- Folgende Kategorien sind in VBA vorhanden:
 - Zahlen als Ganz- oder Dezimalzahlen.
 - Boolesche Werte.
 - Datumswerte.
 - Zeichenfolgen.

Ganze Zahlen

Datentyp	Größe (Bytes)	Datenbereich
AS Byte	1	0 – 255 (ohne Vorzeichen)
AS Integer	2	32.768 – -32.768
AS Long	4	-2.147.483.648 – 2.147.483.647
AS Boolean	2	-1 (True) und 0 (False)

Dezimalzahlen

Datentyp	Größe (Bytes)	Datenbereich	Genauigkeit
AS Single	4	ca. $\pm 3.4 * 10^{38}$... ca. $\pm 1.4 * 10^{45}$	Einfach. Ca. 7 Stellen
AS Double	8	ca. $\pm 1.8 * 10^{308}$... ca. $\pm 4.9 * 10^{324}$	1Doppelt. Ca. 15 Stellen
AS Currency	8	ca. $-9.22 * 10^{14}$... ca. $9.22 * 10^{15}$	15 Vor- und 4 Nachkomma- stellen.

Hinweise zu Zahlen

- In VBA wird als Dezimaltrennzeichen ein Punkt genutzt.
- Die Datentypen Single und Double können Rundungsfehler erzeugen.
- Führende Nullen werden entfernt.
- Es werden keine Maßeinheiten oder Gewichte berücksichtigt. Der Computer weiß nicht, ob die eingegebene Zahl eine Zentimeter- oder Gramm-Angabe ist. Die korrekte Umrechnung muss der Programmierer in Code umsetzen.
- Über- oder Unterschreitungen des Wertebereichs führen zu einem Abbruch des Programms. Es wird ein Laufzeitfehler ausgegeben.

Zeichen-Datentypen

Datentyp	Größe	Datenbereich
AS String	maximal Größe der Festplatte	variable Länge

- ... können jedes beliebige Zeichen enthalten.
- ... werden immer durch Anführungszeichen begrenzt.
- ... können alle ANSI-Zeichen (<http://www.torstenhorn.de/techdocs/ascii.htm>) und Unicode-Zeichen (<http://www.unicode.org/charts/>) enthalten.
- Zahlen als Strings können nicht in Berechnungen genutzt werden.
- ... müssen für Postleitzahlen oder Telefon-Vorwahlnummern genutzt werden.

Beispiel

```
Sub variablen()  
  Dim satz As String  
  Dim plz As String * 5  
  
  satz = "Eisbären leben in der Arktis"  
  plz = "30159"  
  
End Sub
```

Mit Hilfe von * 5 wird der String auf fünf Zeichen begrenzt. Ein längerer String wird ohne Warnung auf die angegebene Länge gekürzt. Ein kürzerer String wird mit Leerzeichen aufgefüllt.

Datums- und Zeitwerte

Datentyp	Größe	Datenbereich
AS Date	8 Bytes	1. Januar 100 bis 31. Dezember 9999

- ... werden durch das Hash (#) - Zeichen begrenzt.
- ... werden in der Form #monat/tag/jahr# sowie #hh:mm:ss AM/PM# eingegeben.

Beispiel

```
Sub variablen()  
    Dim dateWert As Date  
  
    dateWert = #2/21/2009 12:00:00 PM#  
    dateWert = #2/27/2009#  
    dateWert = #1:15:25 PM#  
    dateWert = #12:12:00#  
  
End Sub
```

Der Februar ist auch in VBA auf 28 Tage begrenzt. Eine Tagesangabe größer als 28 wird als Fehler gemeldet.

Hinweise zu Datumswerten

- Datumswerte werden intern als Integer-Zahl interpretiert. Zum Beispiel wird das Datum #7/9/2008# intern als 39638 gespeichert. Die Zählung der Tage beginnt am 30.12.1899.
- Datumswerte werden in Abhängigkeit der Ländereinstellungen und Formatierungen in Excel gespeichert.
- Für Europa wird der Gregorianischen Kalender genutzt. Laut Microsoft gilt dieser Kalender seit 1582.
- Zweistellige Jahresangaben sollten sehr vorsichtig genutzt werden:
 - Die zweistelligen Jahreszahlen 0 bis 30 werden als Jahre im 21. Jahrhundert interpretiert.
 - Die zweistelligen Jahreszahlen 31 bis 99 werden als Jahre im 20. Jahrhundert interpretiert.

Hinweise zu Zeitwerten

- Zeitwerte werden intern als prozentualer Anteil eines Tages interpretiert. Zum Beispiel wird Mitternacht als 0.5 gespeichert.
- Zeitwerte werden in Abhängigkeit der Ländereinstellungen und Formatierungen in Excel gespeichert.

Variablen im Speicher ablegen

- Variablen sind ein Synonym für eine Speicheradresse im Rechner. Die Speicheradresse bezeichnet den Beginn eines Speicherbereichs.
- Die Größe des Speicherbereichs wird durch den Datentyp angegeben.
- In diesem Speicherbereich wird ein Wert von einem bestimmten Datentyp gespeichert.
- Ein Speicher kann man sich als Schrank mit ganz vielen Schubladen vorstellen. Die Schubladen haben in Abhängigkeit des Datentyps unterschiedliche Größen. In jeder dieser Schubladen wird ein Wert abgelegt.

Beispiele

Name der Variablen:	gelb
Wert der Variablen:	4.5
Adresse im Speicher:	0x00401004
Datentyp:	Single
Benötigte Größe:	32 Bit (4 Bytes)

Name der Variablen:	gruen
Wert der Variablen:	"ABC"
Adresse im Speicher:	0x0040100D
Datentyp:	String
Benötigte Größe:	104 Bit (13 Bytes)

Beispiele

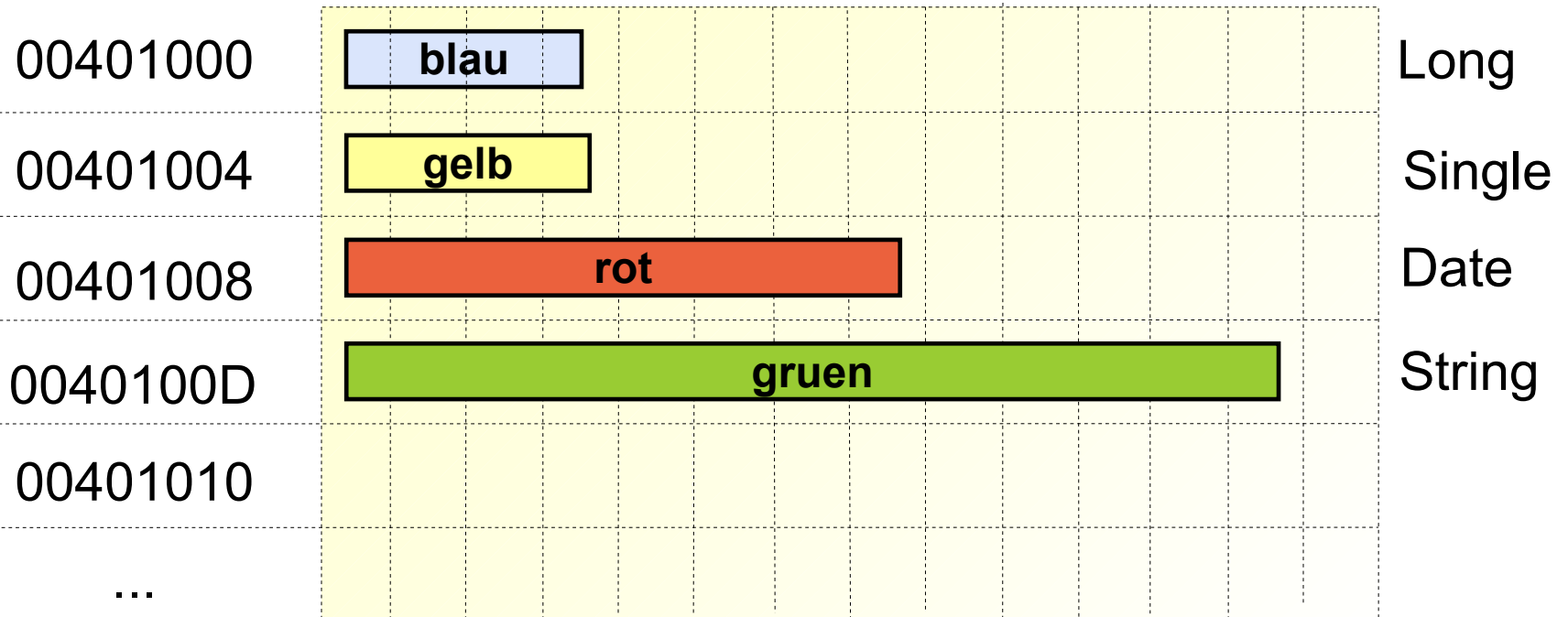
Name der Variablen:	blau
Wert der Variablen:	45000
Adresse im Speicher:	0x00401000
Datentyp:	Long
Benötigte Größe:	32 Bit (4 Bytes)

Name der Variablen:	rot
Wert der Variablen:	#02 24 2008#
Adresse im Speicher:	0x00401008
Datentyp:	Date
Benötigte Größe:	64 Bit (8 Bytes)

Ablage im Speicher

Speicheradresse
(hexadezimal)

1 Byte
┌



Variablen initialisieren

```
Dim faktor As Integer  
Dim messwert as Double
```

```
faktor = 4  
messwert = 3.4 + faktor
```

- Jede Variable hat in Abhängigkeit des Datentyps einen Standardwert.
- Mit Hilfe der Initialisierung wird der Variablen ein definierter Startwert zugewiesen.
- Der zugewiesene Wert
 - ... kann ein, im Code fest verdrahteter Wert sein.
 - ... kann das Ergebnis eines Ausdrucks sein.

Zuweisungsoperator

```
Const faktor As Integer = 1000
```

```
Dim zentimeter As Double
```

```
Dim meter As Double
```

```
zentimeter = Val(txtZentimeter.Text)
```

```
meter = zentimeter / faktor
```



Zuweisungsoperator nutzen

- Mit Hilfe des Gleichheitszeichens wird der Variablen ein Wert zugewiesen.
- Die Variable wird nicht mit einem Wert verglichen!
- Der Variablen links vom Gleichheitszeichen wird das Ergebnis des Ausdrucks rechts vom Gleichheitszeichen zugewiesen.
- Konstanten dürfen in Anweisungen nur rechts vom Gleichheitszeichen stehen. Der Wert einer Konstanten darf nicht verändert werden.

Deklaration erzwingen

Option Explicit

Sub cmdUmrechnen()

Dim zentimeter As Double

Dim txtMilimeter As String

Dim txtMeter As String

Dim txtZentimeter As String

txtZentimeter = "5"

zentimeter = Val(txtZentimeter)

txtMilimeter = (zentimeter * 10) & " mm"

txtMeter = (zentimeter / 1000) & " m"

End Sub

Variablen in dem
Modul müssen
deklariert werden.



Option Explicit

- ... wird am Anfang eines Moduls geschrieben.
- ... kann mit Hilfe des Menüs *Extras – Optionen*; Registerkarte *Editor* für neue Module automatisch gesetzt werden.
- ... gilt für das gesamte Modul in einem Projekt.
- Vorteile:
 - Es werden keine Variablen mit dem gleichen Namen, aber unterschiedlichen Datentypen erzeugt.
 - Falls Variablen genutzt werden, die für das gesamte Modul gelten, erhöht sich die Lesbarkeit und Wartbarkeit des Moduls.

Informationen zu Variablen

- `IsDate(datum)` liefert `true` zurück, wenn der Wert als Datum interpretiert werden kann.
- `IsNumeric(variable)` liefert `true` zurück, wenn mit dem Wert gerechnet werden kann. Die Variable kann als Zahl interpretiert werden.
- `IsNull(variable)` liefert `true` zurück, wenn die Variable oder der Ausdruck keine gültigen Daten enthält. Der Variablenwert ist nicht definiert.
- `VarType(variable)` gibt eine Zahl zurück. Die Zahl symbolisiert einen bestimmten Datentyp.
- `TypeName(variable)` gibt den Datentyp in Textform (String) zurück.

Datentyp Variant

- ... ist ein universeller Datentyp. Die Variable kann jeden Datentyp aufnehmen.
- ... benötigt bei numerischen Werten ca. 16 Bytes und bei Strings 22 Bytes plus die Textlänge. Der Datentyp benötigt sehr viel Speicherplatz.
- ... sollte nur genutzt werden, wenn der Datentyp des zu speichernden Werts unbekannt ist.
- ... kann zu unnötigen Fehlermeldungen führen. Es findet keine automatische Typ-Überprüfung statt.

Informationen zu Variant

- `IsEmpty(variable)` liefert `true` zurück, wenn die Variable nicht initialisiert ist.
- `IsNull(variable)` liefert `true` zurück, wenn die Variable ungültige Daten enthält.

Operatoren

- Einstellige Operatoren:
 - Vorzeichen für positive und negative Zahlen.
- Zweistellige Operatoren:
 - Arithmetische Operatoren.
 - Vergleichsoperatoren.
 - Logische Operatoren zum Verknüpfen von Ausdrücken.
 - Zuweisungsoperator.
 - Textverknüpfungen.

Arithmetische Operatoren

Operator	Rechenart	Beispiel
\wedge	Potenzrechnung	$7^3 = 343$
\backslash	Division; ganzzahliges Ergebnis	$7 \backslash 3 = 2$
$/$	Division	$7 / 3 = 2.33$
Mod	Rest einer Division mit Ganzzahlen	$7 \text{ Mod } 3 = 1$ $7.2 \text{ Mod } 3.1 = \text{Fehler}$
$*$	Multiplikation	$7 * 3 = 21$
$+$	Addition	$7 + 3 = 10$
$-$	Subtraktion	$7 - 3 = 4$

Reihenfolge

- Bei Ausdrücken gelten die Rechenregeln der Mathematik:
 - Klammer vor
 - Potenz vor
 - Punktrechnung vor
 - Strichrechnung.
- Beispiele:
 - Liefert der Ausdruck $(2 + 4 * 2)$ das gleiche Ergebnis wie $((2 + 4) * 2)$?
 - Liefert der Ausdruck $(2^4 / 4)$ das gleiche Ergebnis wie $(2^{(4 / 4)})$?
- Um fehlerhafte Ausdrücke zu vermeiden, sollten große Ausdrücke geklammert werden!

Zeichenfolgen verknüpfen

```
Const strHallo As String = "Hallo, "  
Dim benutzer As String  
  
benutzer = strHallo & "Meier"
```

- Mit Hilfe des kaufmännischen Und (&) können Zeichenfolgen miteinander verknüpft werden.
- Zahlenwerte werden automatisch in Strings umgewandelt.
- Es können Variablen und / oder Konstanten miteinander verknüpft werden.

Steuerzeichen

Konstanten	ANSI-Zeichen	Beschreibung
vbCrLf	Chr(13) & Chr(10)	Kombination aus Wagenrücklauf und Zeilenvorschub
vbNewLine	Chr(13) & Chr(10)	Plattformspezifischer Zeilenumbruch
vbCr	Chr(13)	Wagenrücklauf (Charriage Return)
vbLF	Chr(10)	Zeilenvorschub (Line Feed)
vbTab	Chr(9)	Tabulatorzeichen
vbBack	Chr(8)	Rückschrittzeichen

CHR() und ASC()

- Chr(...)
 - ... wird ein Integer-Wert von 0 bis 255 übergeben.
 - ... liefert das dazugehörige ANSI-Zeichen zurück.
 - Zum Beispiel Chr(65) liefert "A" zurück.
- Asc(...)
 - ... wird ein einzelnes Zeichen übergeben.
 - ... liefert die dazugehörige Dezimalzahl zurück.
 - ... liefert den ganzzahligen ANSI-Code des Zeichens zurück.
 - Zum Beispiel Asc("A") liefert 65 zurück.

Automatische Typumwandlung von Zahlen

- ... oder implizite Typumwandlung.
- VBA versucht einen Datentyp zu finden, in dem alle Werte dargestellt werden können.
- Eine Konvertierung von Zahlen von einem kleineren zu einem größeren Datentyp verläuft ohne Fehler.
 - Byte – Integer – Long – Single – Double
 - Zum Beispiel der Datenbereich Integer ist im Datenbereich Double eingeschlossen. Eine Konvertierung erzeugt keinen Fehler.

Strings und Zahlen

```
Sub stringZahlen()
```

```
Dim intErgebnis As Integer
```

```
Dim txtErgebnis As String
```

```
intErgebnis = "10" + 30 ' = 40
```

```
intErgebnis = "A" + 30 ' Fehler
```

```
intErgebnis = Asc("A") + 30 ' = 95
```

```
intErgebnis = "10 Bären" + 30 ' Fehler
```

```
txtErgebnis = 10 & " Bären" ' = 10 Bären
```

```
txtErgebnis = 10 + " Bären" ' Fehler
```

```
txtErgebnis = "10 Bären" + " 20 Fische" ' = 10 Bären 20 Fische
```

```
End Sub
```


Explizite Typumwandlung

CBool(-1)	' Boolean [wahr]
CByte(256)	' Byte [Fehler: Überlauf]
CInt(2.3)	' Integer [2]
CSng(22.4567878994556)	' Single [22,45679]
CDbl(24)	' Double [24.0]
CCur(24)	' Currency [24.0]
CDate("2.4.07")	' Date [#04.02.2007#]
CStr(2.3)	' String ["2,3"]

MsgBox()

result = MsgBox(prompt [, buttons] [, title] [, helpfile, context])

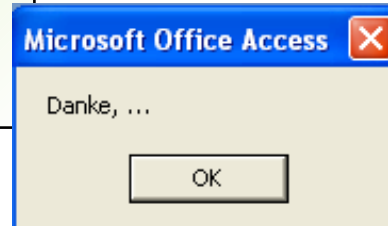
- ... ist ein Dialogfeld mit Schaltflächen zur Anzeige von Text.
- ... kann für kurze Status- oder Fehlermeldungen genutzt werden.
- ... werden Parameter übergeben, die das Aussehen des Dialogfeldes beeinflussen.
- ... gibt einen Integer-Wert zurück.
 - Der Rückgabewert symbolisiert die gedrückte Schaltfläche.
 - Zwischen dem Schließen des Dialogfeldes und der Abbrechen-Schaltfläche besteht kein Unterschied.

Parameter

- `prompt` ist ein Informationstext für den Benutzer in dem Dialogfeld. Der Informationstext hat maximal 1024 Zeichen.
- `title` legt den Text fest, der in der Titelleiste des Dialogfeldes angezeigt wird.
- `buttons`
 - ... bestimmt die angezeigten Schaltflächen.
 - ... legt ein Icon fest, welches die Art der Nachricht verdeutlicht.
 - ... wird aus vordefinierten Konstanten gebildet.

Beispiele

MsgBox "Danke, ..."



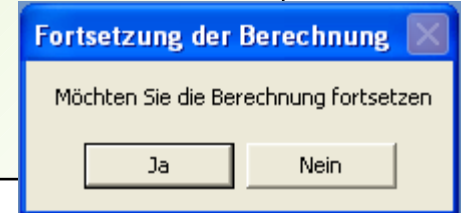
```
Const txtTitel = "Addition"  
Dim message As String  
Dim result As VbMsgBoxResult
```

```
message = x & " + 4 = " & (x + 4)  
result = MsgBox(message, vbInformation, txtTitel)
```

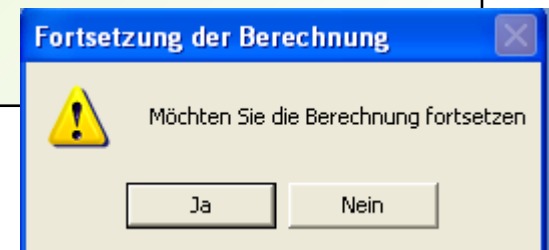


Beispiele

```
Const txtTitel = "Fortsetzung der Berechnung"  
Const message = "Möchten Sie die Berechnung fortsetzen"  
Dim result As Integer  
  
result = MsgBox(message, vbYesNo, txtTitel)
```



```
Const txtTitel = "Fortsetzung der Berechnung"  
Const message = "Möchten Sie die Berechnung fortsetzen"  
Dim result As Integer  
result = MsgBox(message, _  
vbExclamation + vbYesNo, txtTitel)
```



InputBox()

```
result = InputBox(prompt [, title] [, default] [, xPos] [, yPos]  
                [, helpfile, context ])
```

- ... kommuniziert mit dem Benutzer.
- ... ist ein Dialogfeld, die aus einem Textfeld für die Eingabe und einer Schaltfläche zur Bestätigung besteht.
- ... gibt einen String zurück, der den Inhalt des Textfeldes enthält. Der Benutzer kann als Dezimalpunkt ein Komma eingeben.

Parameter

- `prompt` ist ein Informationstext für den Benutzer in dem Dialogfeld.
- `title` legt den Text fest, der in der Titelleiste des Dialogfeldes angezeigt wird.
- `default` gibt ein Standardwert für die Eingabe vor. Der Standardwert kann vom Benutzer geändert werden.
- `xPos` und `yPos` legen die Position der linken, oberen Ecke des Dialogfeldes fest. Die Angaben werden in Twips per Pixel gerechnet (20 Twips sind 1 Pixel).

Zahlen aus einem String herausfiltern

Val([string])

zentimeter = Val(txtZentimeter)

- Die Funktion filtert beginnend vom Anfang des Strings alle Zahlen heraus.
- Sobald ein nicht numerisches Zeichen gefunden wird, bricht die Funktion ab. Dem nicht numerischen Zeichen können auch weitere Zahlen folgen. Diese werden nicht erkannt.
- Das Dezimaltrennzeichen wird von der Funktion erkannt.
- Zum Beispiel:
 - Val("30159 Hannover") liefert die Zahl 30159.
 - Val("10.99 €") liefert die Zahl 10.99.
 - Val("10,99 €") liefert die Zahl 10.
 - Val("1.000,99") liefert die Zahl 1.

Strings und Zahlen

```
Sub addition()  
    Const zahl = 10  
  
    Dim strWert As String  
    Dim intWert As Integer  
    Dim summe As Integer  
  
    strWert = InputBox("Bitte geben Sie einen Wert ein:")  
    intWert = CInt(strWert)  
    summe = intWert + zahl  
  
End Sub
```