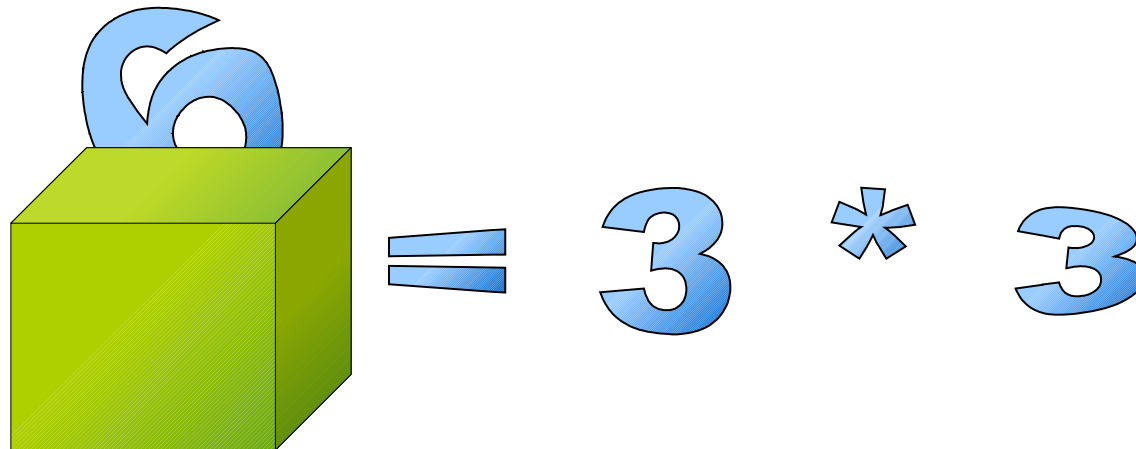


# Access 2010 – Programmierung

## Anweisungen in VBA



## Beispiel

- Aufgabe:
  - Jeweils ein Wert wird in die obersten Textfelder eingegeben.
  - Die eingegebenen Werte werden addiert und mit zwei multipliziert.
- Lösung:
  - An das Ereignis „Beim Klicken“ der Schaltfläche wird eine Prozedur angehängt.
  - In dieser Prozedur werden die Werte der beiden Textfelder ausgelesen.
  - Die Berechnung wird durchgeführt.
  - Das Ergebnis der Berechnung wird in die jeweiligen Textfelder geschrieben.

## Objekt ...

- ist ein Formular, Bericht oder ein Steuerelement. In der Aufgabe ist zum Beispiel das Textfeld ein Objekt.
- hat Eigenschaften. Eigenschaften beschreiben ein Objekt. Die Eigenschaften eines Objektes werden auf den Registerkarten *Format*, *Daten* und *Andere* im Eigenschaftenblatt abgebildet. In VBA werden die englischsprachigen Bezeichnungen genutzt. Das Objekt und deren Eigenschaften werden durch ein Punkt in VBA getrennt.
- reagiert mit Hilfe von Ereignisprozeduren auf Benutzeraktionen. In diesem Beispiel wäre ein Ereignis „Klick auf die Schaltfläche“.
- hat Methoden, die Eigenschaften automatisiert verändern.

## Ereignis „Beim Klicken“ auswählen

- Das Formular ist in der Entwurfsansicht geöffnet.
- Die Schaltfläche ist aktiviert.
- Das Eigenschaftensblatt ist geöffnet und das Register Ereignis ist eingeblendet.
- Mit Hilfe des Pfeils nach unten am rechten Rand des Eintrags Beim Klicken wird eine Liste geöffnet. Aus der Liste wird das Element [Ereignisprozedur] ausgewählt.
- Mit Hilfe der Schaltfläche *Drei Punkte* wird der VBA-Editor geöffnet. Die gewählte Prozedur wird automatisch angelegt.

## Ereignisprozeduren ...

- haben den Kopf « Private Sub cmdAddition\_Click() ».
- werden mit Hilfe des Schlüsselwortes « Sub » als Prozedur gekennzeichnet. Prozeduren geben keinen Wert an den Aufrufer zurück.
- enden mit der Zeile « End Sub ».
- haben das Zugriffsrecht « Private ». Die Prozedur kann nur vom Eigentümer und keinen Außenstehenden gestartet werden. In diesem Beispiel kann die Ereignisprozedur nur die Schaltfläche „cmdAddition“ starten.

## Der Namen einer Ereignisprozedur ...

- beginnt mit dem Namen des auslösenden Objekts. In diesem Beispiel wird der Name der Schaltfläche „cmdAddition“ genutzt.
- endet mit dem ausgelösten Ereignis. In diesem Beispiel „Click“, ausgelöst durch Mausklick des Benutzers auf die Schaltfläche.
- Der Namen des auslösenden Objekts wird durch ein Unterstrich von der Bezeichnung des Ereignisses getrennt.

## Argumente einer Ereignisprozedur ...

- werden als Liste in den runden Klammern im direkten Anschluss an den Prozedurnamen aufgeführt.
- werden durch ein Kommata in der Liste getrennt.
- Die Prozedur « `_Click()` » bekommt keine Argumente übergeben. Die runden Klammern sind leer.
- Die Prozedur « `_KeyPress(KeyAscii As Integer)` » hat exakt ein Argument. Der Prozedur wird die gedrückte Taste übergeben.
- Die Prozedur « `_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)` » hat eine Argumentliste. Es wird als Argument die Maustaste und deren Position übergeben. Die Argumente werden durch Kommata getrennt.

## Code eingeben

- Mit einem Mausklick auf eine freie Zeile zwischen dem Prozedurkopf und « End Sub » wird die Einfügemarke eingeblendet.
- Der Code wird dort mit Hilfe der Tastatur eingegeben.

## Code-Schnipsel

```
Dim wert01 As Integer  
Dim wert02 As Integer  
Dim summe As Integer  
Dim ergebnis As Integer
```

```
wert01 = txtValueFirst.Value  
wert02 = txtValueSecond.Value
```

```
summe = wert01 + wert02  
ergebnis = summe * 2
```

```
txtSumme.Value = summe  
txtErgebnis.Value = ergebnis
```

Deklaration  
von Variablen

Anweisungen

## Objekte und deren Eigenschaften

« txtValueSecond.Value »

- « txtValueSecond » ist der Name eines Steuerelements.
- « Value » ist die Eigenschaft „Wert“ des Steuerelements. Der Inhalt des Textfeldes, der Wert eines Optionsfeldes etc. wird zurückgegeben.

## Anweisungen ...

- werden aus Ausdrücken gebildet.
- werden meist sequentiell ausgeführt. Das heißt, die Anweisungen werden Zeile für Zeile ausgeführt.
- symbolisieren einen bestimmten Arbeitsschritt.
- sind die kleinste Einheit eines VBA-Programms.
- In VBA steht pro Zeile eine Anweisung.

## Ausdrücke ...

- bestehen aus Operanden und Operatoren, die nach bestimmten Regeln zusammengesetzt werden.
- berechnen mit Hilfe von Operatoren und Operanden einen neuen Wert.
- sind Verarbeitungsvorschriften, die ein Ergebnis zurückgeben.
- verändern den Wert von Variablen entsprechend des angegebenen Datentyps.

## Operanden

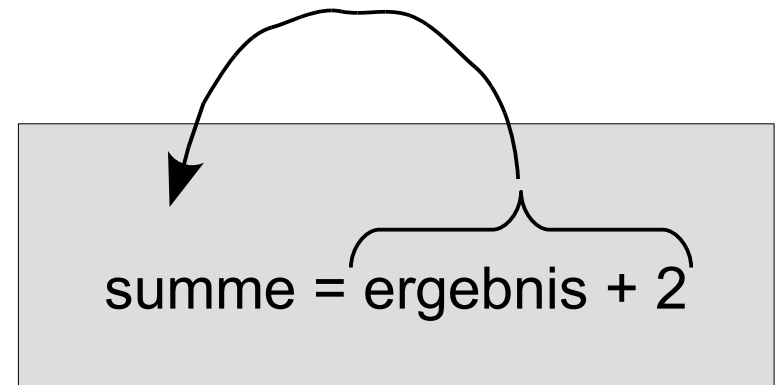
- können Variablen sein. Variablen ...
  - sind benannte Platzhalter für einen Wert.
  - können durch das Programm verändert werden.
  - müssen vor der Nutzung initialisiert werden.
- können Konstanten sein. Konstanten ...
  - werden direkt in den Code eingegeben.
  - sind Ganzzahlen, Dezimalzahlen oder Text.
  - sind nicht durch Code veränderbar.

## Operatoren

- Arithmetische Operatoren berechnen einen Wert.  
Beispiel: «  $a + b$  ».
- Vergleichsoperatoren vergleichen zwei Werte.  
Beispiel: «  $a > b$  ». Als Ergebnis liefert der Ausdruck einen boolschen Wert zurück.
- Logische Operatoren verknüpfen boolsche Werte. Boolsche Werte sind wahr oder falsch.  
Beispiel: «  $(a \geq 10) \text{ And } (a \leq 20)$  ».
- Der Zuweisungsoperator weist einer Variablen einen Wert zu.

## Zuweisungsoperator

- Als Zuweisungsoperator wird das Gleichheitszeichen genutzt.
- Rechts vom Gleichheitszeichen steht ein Ausdruck. Der Ausdruck besteht aus festen Werten (Konstanten), Variablen und Operatoren.
- Der, durch den Ausdruck berechnete Wert, wird der Variablen links vom Gleichheitszeichen zugewiesen.



## Beispiele

- « `ergebnis = summe * 2` ».
  - Der Ausdruck rechts vom Gleichheitszeichen wird berechnet und der Variablen „ergebnis“ zugewiesen.
  - Die Konstante „2“ wird mit dem, in der Variablen « `summe` » multipliziert.
- « `txtSumme.Value = summe` »
  - In diesem Beispiel wird ein variabler Wert genutzt, um eine Eigenschaft eines Objekts zu setzen.
  - „txtSumme“ ist der Name des Objekts. Es wird die Eigenschaft „Value“ des Objekts neu gesetzt. Das Objekt und deren Eigenschaft werden durch einen Punkt getrennt.

# Arithmetische Operatoren

Operator	Rechenart	Beispiel
+	Positives Vorzeichen	ergebnis = +3
-	Negatives Vorzeichen	ergebnis = -3
+	Addition	ergebnis = 3 + 4
-	Subtraktion	ergebnis = 3 - 4
^	Potenzrechnung	ergebnis = 3 <sup>4</sup>

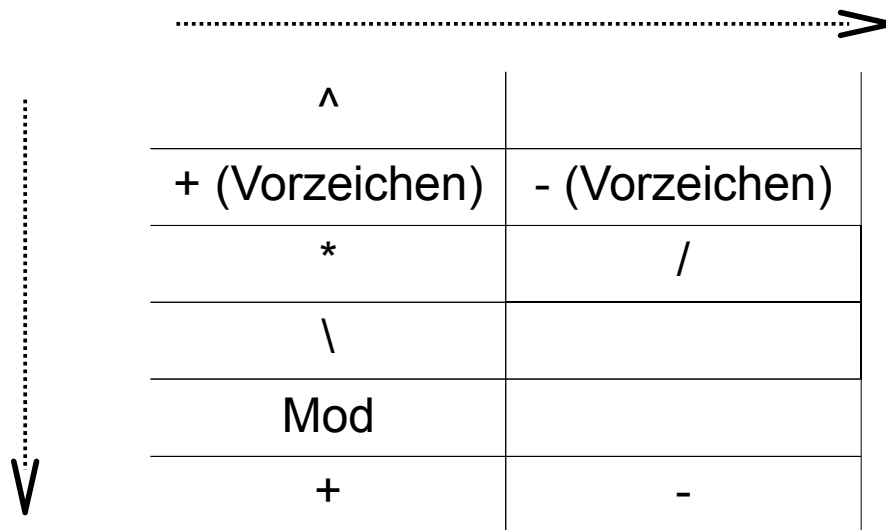
# Arithmetische Operatoren

Operator	Rechenart	Beispiel
*	Multiplikation	ergebnis = $3 * 4$
/	Division	ergebnis = $3 / 4$ ergebnis = 0.75 Fehler = $3 / 0$
\	Ganzzahlige Division	ergebnis = $3 \setminus 4$ ergebnis = 0
Mod	Modula (Division mit Rest). Nur für Ganzzahlen.	ergebnis = $3 \% 4$ ergebnis = 3 ergebnis = $4 \% 3$ ergebnis = 1

## Zusammengesetzte Ausdrücke

- Der Ausdruck «  $a * b + c$  » entspricht dem Ausdruck «  $(a * b) + c$  ». Das Malzeichen hat eine höherer Bindung als das Minuszeichen. Die Abarbeitung der Rechenschritte in einem zusammengesetzten Ausdruck erfolgt in Abhängigkeit der Rangfolge der Operatoren.
- Der Ausdruck «  $a - b - c$  » entspricht dem Ausdruck «  $(a - b) - c$  ». Gleiche Operatoren werden immer von links nach rechts abgearbeitet.
- Um Fehler zu vermeiden und die Lesbarkeit zu erhöhen, werden häufig Klammern in Ausdrücken gesetzt.

# Rangfolge der arithmetischen Operatoren



$\wedge$	
+ (Vorzeichen)	- (Vorzeichen)
*	/
\	
Mod	
+	-

- In der Tabelle werden die Operatoren, beginnend mit der höchsten Priorität aufgelistet.

## Kommentare ...

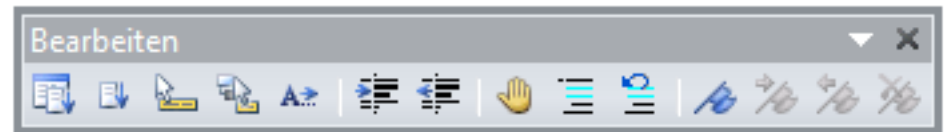
- erleichtern das Verstehen eines Programms.
- bieten Informationen für den Entwickler.
- müssen bei Code-Änderungen oder Programm-Erweiterungen angepasst werden.
- werden vom Computer überlesen.
- beziehen sich auf die Frage  
„Warum wird in dieser Zeile diese Aktion ausgeführt?“ und nicht  
„Wie wird diese Aktion in dieser Zeile ausgeführt?“

## Einzeilige Kommentare ...

« ' Dies ist ein Kommentar ».

- beginnen immer mit dem Apostroph.
- stehen rechts oder oberhalb von der zu kommentierenden Zeile.
- werden standardmäßig mit einer grünen Schriftfarbe gekennzeichnet.

## Mehrzeilige Kommentare ...



- können am einfachsten mit den entsprechenden Icons in der Symbolleiste Bearbeiten hergestellt werden.
- werden zuerst geschrieben. Anschließend werden alle Zeilen des Kommentars mit der Maus markiert. Mit einem Klick auf das Symbol *Block auskommentieren* in der Symbolleiste *Bearbeiten* werden die ausgewählten Zeilen automatisiert als Kommentar gekennzeichnet.
- werden mit Hilfe des Symbols *Auskommentierung des Blocks aufheben* automatisiert in Code umgewandelt

## Variablen ...

- sind Platzhalter für Zahlen und Zeichen.
- sind Container für einen Wert.
- werden durch das Programm verarbeitet.
- müssen vor der ersten Nutzung initialisiert werden.
- können mit Hilfe der Tastatur oder durch Ausdrücke verändert werden.
- beziehen sich nicht auf einen bestimmten Wertetyp.
- haben einen eindeutigen Namen.

## ... aus der realen Welt

- Die Menge der Zutaten in einem Kochrezept werden entsprechend der Anzahl der Personen angepasst. Die Mengenangaben sind variabel.
- Je nach Menge oder Verschmutzungsgrad der Wäsche wird mehr oder weniger Waschmittel in die Waschmaschine gegeben.
- Es wird ein bestimmtes Wort in einem Text gesucht. Es wird ein Positionszähler benötigt, der an den einzelnen Buchstaben entlang wandert.

## ... in VBA

« Dim variable »

« Dim wert02 »

- Mit Hilfe des Schlüsselworts « Dim » werden Variablen gekennzeichnet.
- Dem Schlüsselwort folgt ein frei wählbarer Name.
- Die Variable wird am Anfang eines Moduls oder einer Prozedur / Funktion deklariert.

## Regeln für den Variablennamen

- Jeder benutzerdefinierter Name beginnt mit einem Buchstaben.
- In der Programmierung sollten nur die Buchstaben a..z, A..Z, der Unterstrich und die Zahlen 0..9 genutzt werden.
- Es dürfen keine Schlüsselwörter wie zum Beispiel « Dim », « Sub » als Variablennamen genutzt werden.
- Jeder Variablenname wird in einer Prozedur / Funktion oder Modul nur einmal genutzt.
- Ein Variablenname hat maximal 255 Zeichen.

## Geeignete Variablennamen ...

- geben Auskunft über die Nutzung des Platzhalters.
- beschreiben den zu speichernden Wert.
- entsprechen dem Sprachraum des Entwicklers.
- sind an die Sprache der realen Welt angelehnt.

## Beispiele

Variable für ...	Geeignet	Nicht lesbar
Bestellnummer	bestellnummer strBestellNr	bnr bn
Rechnungsdatum	rechnung_Datum dateRechnung	rgD dat1203
Bestellmenge	bestellmenge intMenge	m bm
Warenpreis	preisProEinheit artikelpreis	cur pr

## Datentypen ...

- beschreiben den zu speichernden Wert.
- legen Regeln für die Verwendung der Variablen fest.
- beschreiben den Datenbereich des Platzhalters.
- legen den Speicherbedarf fest.
- sind für Ganzzahlen, Dezimalzahlen, Datums- und Zeitwerte sowie Zeichenfolgen vorhanden.

## ... angeben

« Dim variable As Datentyp »

« Dim wert02 As Integer »

- Mit Hilfe des Schlüsselworts « As » wird der Variablen ein bestimmter Datentyp zugewiesen.
- « As Integer » deklariert einen Platzhalter für Ganzzahlen (Integer).

## ... für Ganzzahlen

	Speicherbedarf in Bytes	Datenbereich
As Byte	1	0 - 255
As Integer	2	-32.768 - +32.767
As Long	4	-2.147.483.648 - +2.147.483.647
As Boolean	2	0 (falsch, false) <> 0 (wahr, true)

## ... für Dezimalzahlen

	Speicherbedarf in Bytes	Datenbereich
As Single	4	-3,402823E38 bis -1,401298E-45 für negative Werte; 1,401298E-45 bis 3,402823E38 für positive Werte
As Double	8	-1,79769313486232E308 bis -4,94065645841247E-324 für negative Werte; 4,94065645841247E-324 bis 1,79769313486232E308 für positive Werte

## ... für Darstellung von Währungen

	Speicher- bedarf in Bytes	Datenbereich
As Currency	8	-922.337.203.685.477,5808 bis 922.337.203.685.477,5807

## Genauigkeit von Dezimalzahlen

- « As Single » beschreibt ein Wert von einfacher Genauigkeit. Der Platzhalter enthält eine Näherung an eine reelle Zahl.
- « As Double » beschreibt ein Wert von doppelter Genauigkeit. Der Platzhalter enthält eine Näherung an eine reelle Zahl.
- « As Currency » besitzt 15 Stellen vor dem Dezimalkomma und vier Stellen nach dem Dezimalkomma.

## Hinweise

- Als Dezimaltrennzeichen wird ein Punkt genutzt.
- Führende Nullen werden entfernt.
- Eine Null als Nachkommastelle wird durch das #-Zeichen ersetzt.
- Boolesche Variablen können die Schlüsselwörter « True » (Wahr) oder « False » (Falsch) zugewiesen werden.
- Das Programm kennt keine Maßeinheiten, Längenangaben etc.

## ... für Datums und Zeitwerte

	Speicherbedarf	Datenbereich
As Date	8 Bytes	1. Januar 100 – 31. Dezember 9999 00:00:00 - 23:59:59

## Hinweise

« datum = #2/23/2011# »

« uhrzeit = #1:20:00 PM# »

- Konstante Datums- und Zeitwerte werden durch das #-Zeichen begrenzt.
- Datumswerte werden in dem Format Monat/Tag/Jahr angegeben. Es wird das kurze Datumsformat genutzt. Wenn möglich, sollten keine zweistelligen Jahresangaben genutzt werden.
- Zeitwerte werden in dem Format Stunde : Minute : Sekunde eingegeben. Zeitangaben werden in einem 12- oder 24-Stunden-Format in Abhängigkeit des Computers dargestellt.

## ... für Zeichenfolgen

	Länge
As String	Variable Länge.
As String * 5	Die Länge des Strings wird auf eine bestimmte Anzahl eingeschränkt. In diesem Beispiel besteht die Zeichenfolge aus fünf Zeichen.

## Hinweise

« `ausgabeText = "Guten Tag"` »

- Zeichenfolgen werden durch Anführungsstriche begrenzt.
- Telefonnummern, Postleitzahlen werden in diesem Format gespeichert.
- Zeichenfolgen können Zahlen enthalten, mit denen nicht gerechnet wird.

## Text-Zeichen ...

- werden mit Hilfe von Zahlen im Bereich von 0 bis 255 codiert.
- Die ersten 128 Zeichen entsprechen dem ASCII-Zeichensatz. Dieser Zeichensatz enthält alle Zeichen der US-amerikanischen Tastatur.
- Die Zeichen von 128 bis 255 sind Sonderzeichen wie zum Beispiel ö, ä etc.

# ASCII-Zeichentabelle

0 =	18 = ↑	36 = \$	54 = 6	72 = H	90 = Z	108 = l
1 = ☺	19 = !!	37 = %	55 = 7	73 = I	91 = [	109 = m
2 = ☹	20 = ¶	38 = &	56 = 8	74 = J	92 = \	110 = n
3 = ♥	21 = §	39 = '	57 = 9	75 = K	93 = ]	111 = o
4 = ♦	22 = −	40 = (	58 = :	76 = L	94 = ^	112 = p
5 = ♣	23 = ↓	41 = )	59 = ;	77 = M	95 = _	113 = q
6 = ♠	24 = ↑	42 = *	60 = <	78 = N	96 = `	114 = r
7 = •	25 = ↓	43 = +	61 = =	79 = O	97 = a	115 = s
8 = ◼	26 = →	44 = ,	62 = >	80 = P	98 = b	116 = t
9 = ◊	27 = ←	45 = -	63 = ?	81 = Q	99 = c	117 = u
10 = ◐	28 = L	46 = .	64 = @	82 = R	100 = d	118 = v
11 = ♂	29 = ↔	47 = /	65 = A	83 = S	101 = e	119 = w
12 = ♀	30 = ▲	48 = 0	66 = B	84 = T	102 = f	120 = x
13 = ♪	31 = ▼	49 = 1	67 = C	85 = U	103 = g	121 = y
14 = ♫	32 =	50 = 2	68 = D	86 = V	104 = h	122 = z
15 = ✳	33 = !	51 = 3	69 = E	87 = W	105 = i	123 = {
16 = ▼	34 = "	52 = 4	70 = F	88 = X	106 = j	124 =
17 = ▲	35 = #	53 = 5	71 = G	89 = Y	107 = k	125 = }

## Seiten im Web für ...

- Standard-ASCII-Code
  - <http://www.torsten-horn.de/techdocs/ascii.htm>
  - <http://www.asciitable.com/>
- Standard-Windows-Zeichensatz
  - [http://www.vbarchiv.net/commands/cmd\\_ascii-codes.html](http://www.vbarchiv.net/commands/cmd_ascii-codes.html)

## Code eines Text-Zeichens

- « Asc("A") » gibt die Zeichenkodierung für das Zeichen „A“ zurück. A wird mit der Ganzzahl 65 kodiert.
- « Chr(65) » gibt für das Zeichen die Kodierung zurück. Die Ganzzahl 65 kodiert den Buchstaben „A“.

## Zeichenfolgen (String) verketteten

« ort = "30165" & " Hannover" »

« anzahlTiere = 3 & " Eisbären" »

« anschrift = firma & " " & strasse & " " & ort »

- Strings werden durch das kaufmännische Und verbunden.
- Es können String-Variablen miteinander verkettet werden.
- Es können Zahlen mit Strings verbunden werden. Die angegebene Zahl wird automatisch in einem String umgewandelt.

## Zahlen am Anfang eines Strings filtern

« 3 = Val("3 Eisbären, 4 Affen") »

« 30159 = Val("30159 Hannover") »

- Als Argument wird ein String an die Funktion « Val() » übergeben.
- Als Rückgabewert werden alle Zahlen bis zu dem ersten nicht numerischen Zeichen zurückgeliefert.
- Wenn der String mit einem nicht numerischen Zeichen beginnt, wird eine Null zurückgeliefert.

## Universeller Datentyp ...

« Dim variable As Variant »

- erlaubt keine Überprüfung des Wertes.
- sollte nur genutzt werden, wenn der zu speichernde Datentyp nicht bekannt ist.
- benötigt sehr viel Speicherplatz und hat eine etwas langsamere Performances.
- Der Datentyp des gespeicherten Wertes kann mit Hilfe von Funktionen ermittelt werden.

## Infos zum Datentyp einer Variablen

« TypeName(zahlByte) »

« VarType(456987.5) »

- Als Argument wird der Name der Variablen oder eine Konstante übergeben.
- « TypeName() » gibt die Bezeichnung des Datentyps zurück.
- « VarType() » gibt eine Ganzzahl zur Kennzeichnung des Datentyps zurück.

## Überprüfung von Variablen

- « IsNumeric(zahlCurrency) ». Ist der Wert als Zahl interpretierbar?
- « IsDate(zeitwert) ». Ist der Wert als Datum- oder Zeitwert interpretierbar?
- « IsNull(ausgabe) ». Ist der Wert der Variablen undefiniert? In VBA hat jede Variable einen definierten Standardwert.
- Als Argument kann den Funktionen der Name einer Variablen übergeben werden.
- Alle Funktionen geben true (wahr) oder false (falsch) zurück.

## Ist eine Variable deklariert?

Option Explicit

- ' Diese Anweisung gilt für das gesamte Modul.
- ' Die Anweisung steht am Anfang eines Moduls.
- ' Falls eine Variable nicht deklariert ist, wird eine Fehlermeldung ausgegeben.

Sub Addition()

Dim zahl01 As Integer

Dim zahl02 As Integer

Dim ergebnis As Integer

zahl01 = 4

zahl02 = 3                   ' Diese Variable ist nicht deklariert → Fehler!

ergebnis = zahl01 + zahl02

End Sub

## Konstante Variablen ...

- haben einen festen Wert, der nicht verändert werden kann.
- sind Platzhalter für einen bestimmten Wert.
- werden auch als Literale bezeichnet.

## ... deklarieren

« Const konstante As Integer = 2 »

« Const ausgabedatum = #2/23/2011# »

- Die Deklaration beginnt mit dem Schlüsselwort « Const ».
- Dem Schlüsselwort folgt der Name der Konstanten.
- In der Deklaration muss Konstanten mit Hilfe eines Gleichheitszeichens ein Wert zugewiesen werden.
- Der Bezeichnung kann die Angabe eines Datentyps folgen. Falls keine Angabe gemacht wird, legt der angegebene Wert den Datentyp fest.

## Systemdefinierte Konstanten ...

- werden von der Programmiersprache oder der Office-Anwendung bereitgestellt.
- Das Präfix « vb » kennzeichnet Konstanten der Programmiersprache VBA.
- Das Präfix « mso » kennzeichnet Konstanten der Office-Anwendung.
- Das Präfix « ac » kennzeichnet Konstanten der Office-Anwendung Access.

## VBA-Konstanten für die Anzeige von Text

VBA-Konstante	ASCII-Zeichen	Beschreibung
vbCrLf	Chr(13) & Chr(10)	Wagenrücklauf und Zeilenvorschub
vbNewLine	Chr(13) & Chr(10)	Plattformspezifischer Zeilenumbruch
vbCr	Chr(13)	Wagenrücklauf
vbLf	Chr(10)	Zeilenvorschub
vbTab	Chr(9)	Tabulatorzeichen
vbBack	Chr(8)	Rückschrittzeichen

## Sammlung von Konstanten

```
Option Compare Database
```

```
Option Explicit
```

```
Enum cFontStyle
```

```
    cNormal = 400
```

```
    cBold = 700
```

```
    cThin = 100
```

```
End Enum
```

## ... deklarieren

« Enum bezeichnung ... End Enum »

- Die Deklaration beginnt mit dem Schlüsselwort « Enum » und endet mit « End Enum ».
- Dem Schlüsselwort folgt der Name der Enumeration. Die Bezeichnung ist frei wählbar, sollte aber den Regeln von Variablen folgen.
- Zwischen dem Anfang und dem Ende der Enumeration werden zeilenweise die Konstanten aufgelistet.
- Eine Enumeration wird immer am Anfang eines Moduls deklariert.

## Konstanten in einer Enumeration

« cNormal = 400 »

- Jede Konstante in einer Enumeration hat einen Namen.
- Die Bezeichnungen sind frei wählbar, sollte aber den Regeln für Variablennamen folgen.
- Mit Hilfe des Gleichheitszeichens wird der Konstanten ein fester Wert zugewiesen.
- Der zugewiesene Wert ist immer vom Typ « Long ».

## ... als Datentyp nutzen

« Dim schriftgewicht As cFontStyle »

- Statt dem Standard-Datentyp wird der Name der Enumeration angegeben.
- Variable x ist vom Datentyp der Enumeration y.

## Zuweisung

« schriftgewicht = cBold »

- Der Variablen wird mit Hilfe einer Konstanten ein ganzzahliger Wert zugewiesen.
- Die Konstante ist in der Enumeration definiert.

## Sammlung von Variablen (Array)

- Eine Array ist eine Liste von variablen Werten zu einem bestimmten Thema.
- Jede Variable in eine Array hat den gleichen Datentyp.
- Ein Array kann eindimensional (Liste) oder mehrdimensional (Tabelle) sein.
- Arrays können verschachtelt werden.

## Beispiele aus der realen Welt

- für eindimensionale Arrays:
  - Monatsnamen
  - Schulnoten
  - Temperaturwerte
- für mehrdimensionale Arrays:
  - Matrizen
  - Inhalt einer Tabelle

## Beispiel in VBA

```
«  
Dim monatsnamen(12) As String  
  
monatsnamen(0) = "Januar"  
monatsnamen(1) = "Februar"  
monatsnamen(2) = "März"  
monatsnamen(3) = "April"  
monatsnamen(4) = "Mai"  
»
```



## Deklaration eines eindimensionalen Arrays

« Dim monatsnamen(12) As String »

- « Dim monatsnamen As String » entspricht der Deklaration einer Variablen.
- In den runden Klammern wird der Index des letzten Elements angegeben. Das erste Element hat standardmäßig den Index 0.
- In diesem Beispiel besteht das Array aus 13 Elementen.

## Zuweisung von Elementen

« `monatsnamen(0) = "Januar"` »

- Das gewünschte Element wird durch den Index festgelegt.
- Der Index folgt dem Array-Namen direkt in runden Klammern.
- Mit Hilfe des Gleichheitszeichens wird dem Element ein Wert passend zum Datentyp zugewiesen.
- In diesem Beispiel wird dem ersten Element im Array der Wert Januar zugewiesen.
- Als Index können nur Ganzzahlen genutzt werden.

## Deklaration eines eindimensionalen Arrays

« Dim listJahre() As String »

- « Dim listJahre As String » entspricht der Deklaration einer Variablen.
- In den runden Klammern wird kein Index für das letzte Element festgelegt.
- Das Array hat eine undefinierte Länge. Die Dimension wird dynamisch festgelegt.

## Dimension festlegen

« ReDim listJahre(0) »

« ReDim Preserve listJahre(zaehler) »

- Das Schlüsselwort « ReDim » legt eine Dimension fest. In dem oberen Beispiel hat das Element einen Index von 0. Es wird das erste Element erstellt. Alle vorhandenen Elemente des Arrays werden gelöscht.
- Falls das Array vergrößert wird, verhindert das Schlüsselwort « Preserve » eine Löschung der vorhandenen Elemente in dem Array.