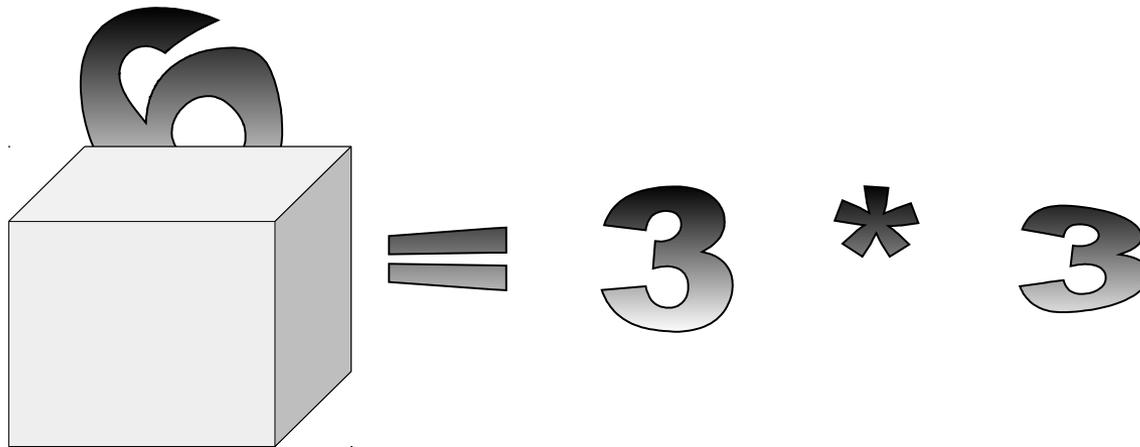


V(isual) B(asic for) A(pplication) Variablen, Konstanten und Ausdrücke



Kommentare

- Hilfe für den Entwickler.
- Wer hat wann welche Änderung vorgenommen?
- Warum werden diese Anweisungen hier ausgeführt?
- Bei Codeänderungen müssen Kommentare entsprechend angepasst werden.

Beispiel

- Warum wird an dieser Position im Code diese Berechnung durchgeführt? Nicht: Wie werden die Werte berechnet?
- Warum wird der String an diesem Zeichen getrennt? Nicht: Wie wird der String getrennt?

Kommentare

```
Public Sub Kommentare()
```

```
    ' Kommentar am Anfang einer Prozedur
```

```
    Dim zeichenkette As String ' Kommentar zur Erläuterung
```

```
End Sub
```

- Kommentare beginnen mit einem Apostroph.
- Kommentare werden durch die Zeile beendet.
- Kennzeichnung durch eine grüne Schriftfarbe.

... am Anfang eines Moduls

- ' Welches Projekt wird in dem Modul implementiert
- ' Auflistung der Prozeduren plus eine Kurzbeschreibung
- ' Variablen und deren Nutzung

Public Sub Kommentare()
End Sub

- Kurze Beschreibung des Projekts.
- Wenn in einem Team gearbeitet wird, sollten auch Informationen über Bearbeiter und den Bearbeitungsstatus angegeben werden.

... am Anfang einer Prozedur

```
Public Sub Kommentare()
```

```
    ' Kommentar am Anfang einer Prozedur
```

```
End Sub
```

- Die Aufgabe kann in einem Kommentar beschrieben werden. Besser: Der Name der Prozedur spiegelt die Aufgabe wieder.
- Beschreibung der Eingabeparameter und eventuell des Rückgabewertes.

... im Anschluss einer Anweisung

```
Public Sub Kommentare()
```

```
    Dim zeichenkette As String ' Kommentar zur Erläuterung
```

```
End Sub
```

- Beschreibung der Aktion in der Zeile.
- Hinweise für den Entwickler bei komplizierten Berechnungen.
- Nutzung der links vom Kommentar deklarierten Variablen in der Prozedur.

... setzen mit Hilfe der Symbolleiste „Bearbeiten“



Markierten Text als
Kommentar kennzeichnen

Kommentar aufheben

Anweisungen

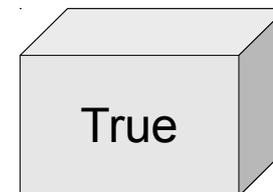
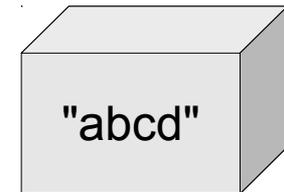
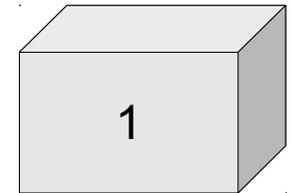
- Pro Zeile steht eine Anweisung.
- Die Anweisung endet automatisch mit der Zeile.

Block

- Zusammenfassung von Anweisungen.
- Prozeduren fassen zum Beispiel Blöcke von Anweisungen zusammen. Der Block beginnt mit dem Prozedur-Kopf Sub und endet mit End Sub.

Speicherung von Werten in Variablen

- Eine Variable ist ein Container für Werte von einem bestimmten Typ.
- Eine Variable kann einen Wert speichern. Der Wert kann der Variable zugewiesen werden. Der Wert kann durch einen Ausdruck berechnet werden.
- Der, in einer Variablen gespeicherte Werte, kann durch einen anderen Wert vom gleichen Typ überschrieben werden.



Beispiel

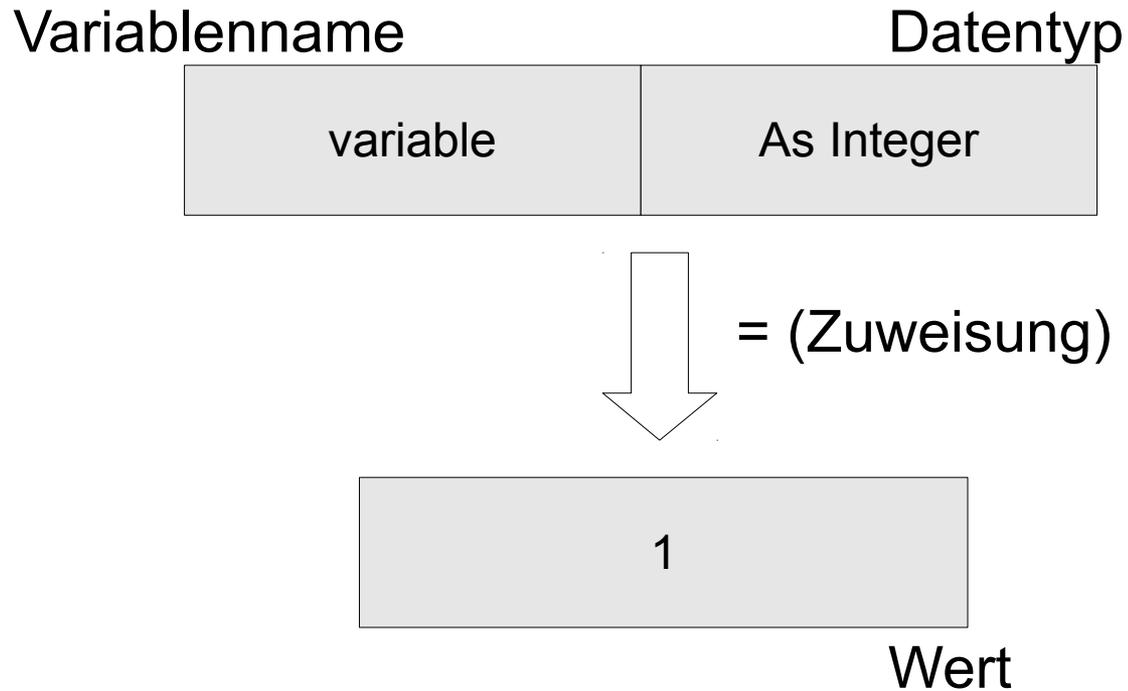
1	ganzzahl As Integer
0.5	gleitkommazahl As Double
"abcd"	wort As String
True	lichtAn As Boolean
#1/30/2017 10:12:00 AM#	datumAngabe As Date

... in VBA

```
Dim ganzzahl As Integer  
Dim gleitkommazahl As Double  
Dim wort As String  
Dim lichtAn As Boolean  
Dim datumAngabe As Date
```

```
ganzzahl = 1  
gleitkommazahl = 0.5  
wort = "abcd"  
lichtAn = True  
datumAngabe = #1/30/2017 10:12:00 AM#
```

Variablen bestehen aus ...



Variablennamen

- Kennzeichnung einer Position im Speicher.
- Jeder Name kommt nur einmal in einem Block vor.
- Namen für Variablen beginnen mit einem Kleinbuchstaben.

Benutzerdefinierte Namen

- Namen kennzeichnen eindeutig eine Variable, Prozedur etc.
- Namen sind Platzhalter für Werte im Programmiercode.
- Prozedur-Namen sind Platzhalter für Blöcke von Anweisungen. Mit Hilfe des Namens wird ein Block von Anweisungen gestartet.

Regeln

- Jeder benutzerdefinierter Name beginnt mit einem Buchstaben.
- Leerzeichen, Punkt, Ausrufezeichen und die Zeichen @, &, \$ und das Hash-Zeichen dürfen nicht in einem Namen genutzt werden.
- Der Name ist maximal 255 Zeichen lang.

Weitere Regeln

- Jeder Name ist in seinem Gültigkeitsbereich (in seiner Prozedur) einmalig.
- Schlüsselwörter aus der Programmiersprache dürfen nicht genutzt werden.
- VBA beachtet nicht die Groß- und Kleinschreibung.

Reddick-VBA-Namenskonvention

Präfix	Tag	Basisname	Suffix
Kennzeichnung der Lebensdauer	Variablentyp	Basisname	Weitere Informationen
		radius	
	dbl	TemperaturCelsius	
	dblc	Pi	
m	dblc	Pi	

Erläuterung der Namen

- Die Variable `radius` speichert den berechneten Kreisradius.
- Die Variable `dblTemperaturCelsius` speichert Temperaturwerte in Celsius. Der Basisname nutzt die Kamel-Notation, um die einzelnen Worte zu kennzeichnen.
- Der Name `dblPi` kennzeichnet den konstanten Wert π .
- Die Variable `mdblPi` ist als Konstante (`c`) am Anfang eines Moduls (`m`) deklariert.

Informationen im Web

- <https://msdn.microsoft.com/de-de/library/office/gg264773.aspx>
- [https://msdn.microsoft.com/en-us/library/aa240858\(v=vs.60\).aspx](https://msdn.microsoft.com/en-us/library/aa240858(v=vs.60).aspx)
- <https://msdn.microsoft.com/de-de/library/0b283bse.aspx>
- <http://www.xoc.net/standards/rvbanc.asp>

Variablen in einer Prozedur

```
Dim ganzzahl As Integer  
Dim gleitkommazahl As Double  
Dim wort As String  
Dim lichtAn As Boolean  
Dim datumAngabe As Date
```

- Bereitstellung von Speicherplatz für einen Wert entsprechend des Datentyps.
- Die Variable ist innerhalb einer Prozedur (Sub...End Sub) deklariert und kann nur in dieser genutzt werden.

... deklarieren

Dim	wort	As	String
Dim	[name]	As	[datentyp]

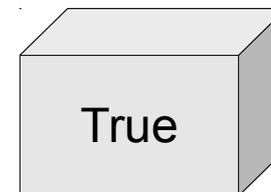
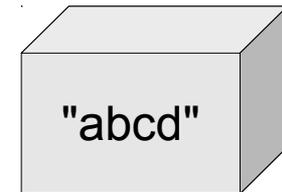
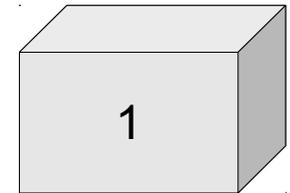
- Die Deklaration in einer Prozedur beginnt mit dem Schlüsselwort Dim.
- Der Name der Variablen ist frei wählbar.
- As weist der Variablen einen Datentyp zu.

Datentypen

- Welche Art von Wert wird in der Variablen gespeichert?
- Wie wird der Wert verarbeitet?
- Bauplan für Zahlen, boolesche Werte (wahr, falsch), Zeichen und Datums-/Zeitangaben.

... in VBA

- Ganz- oder Dezimalzahlen, die als Literale direkt in den Code geschrieben werden. Zahlen können mit Hilfe eines Ausdrucks berechnet werden.
- Einzelne Buchstaben, die zu einem Wort zusammengesetzt werden. Zahlen, die als Text interpretiert werden.
- Datums- und Zeitwerte.
- True (Wahr) oder False (Falsch) als Ergebnis eines Vergleichs von zwei Werten. Boolesche Werte können einer Variablen zugewiesen werden.



Boolsche Werte

Dim lichtAn As Boolean

- Zwei Zustände: wahr (True) oder falsch (False).
- Entspricht einem Lichtschalter. Das Licht ist an oder aus.
- Beantwortung von Ja / Nein-Fragen. Ist der Wert A kleiner als der Wert B?

Strings (Zeichenketten)

```
Dim wort As String  
wort = "ab"
```

- Alphanumerische und numerische Zeichen.
- Zusammenfassung von Buchstaben und Zeichen zu einem Wort, Satz etc.
- Zahlen wie Postleitzahlen, IBAN-Nummer.
- Strings als Literale beginnen und enden mit den Anführungszeichen.

Ganzzahlen

```
Dim bytWert As Byte  
Dim intWert As Integer  
Dim lngWert As Long
```

- Zahlen ohne Nachkommastellen.
- Der Datentyp legt den Wertebereich der Ganzzahl fest.

Möglichkeiten

Datentyp	Wertebereich	Speicherbedarf
Byte	0 - 255	8 Bit
Integer	-32768 - 32767	16 Bit
Long	-2.147.483.648 – 2.147.483.647	32 Bit

Laufzeitfehler „Überlauf“

```
Dim bytWert As Byte
```

```
bytWert = 3000
```

- ... bei Überschreitung des Wertebereiches des angegebenen Datentyps.

Gleitkomma- und Dezimalzahlen

- Zahlen mit Nachkommastellen.
- Als Dezimaltrennzeichen wird der Punkt genutzt.
- `3#` symbolisiert die Zahl 3.0. Die 0 nach dem Dezimaltrennzeichen wird durch das Hash-Zeichen symbolisiert.

Gleitkommazahlen

Dim sngWert As Single
Dim dblWert As Double

- Zahlen, die sich einem Wert annähern.
- Zahlen, die auf x Stellen nach dem Komma genau sind.
- Der Datentyp legt die Genauigkeit der Gleitkommazahl fest.

Möglichkeiten

Datentyp	Wertebereich	Speicherbedarf
Single	Einfache Genauigkeit	32 Bit
Double	Doppelte Genauigkeit	64 Bit

Sonderfall: Währungswerte

Dim curWert As Currency

- Währungswerte sind immer auf vier Nachkommastellen genau.
- Nutzung für Geldbeträge.

Datums- und Zeitangaben

```
Dim datDatum As Date  
Dim datZeit As Date
```

- Datums- und Zeitangaben werden intern als 64-Bit-Gleitkommazahlen (8 Bytes) nach IEEE gespeichert.

Datumsangaben

Dim datDatum As Date

- `datDatum = #1/30/2017#` entspricht dem Datum „30.01.2017“. Datumswerte werden in der Form `monat/tag/jahr` angegeben.
- Datumswerte als Literale beginnen und enden immer mit einem Hash-Zeichen.

Hinweise

- Gültige Datumswerte liegt zwischen dem 1. Januar 100 n. Chr (-657434) und dem 31. Dezember 9999 n. Chr. (2958465).
- `datDatum = 1` entspricht dem Datum „31.12.1899“. Der 30. Dezember 1899 hat den Wert null. Alle nachfolgenden Werte werden durch positive Zahlen dargestellt.
- `datDatum = -1` entspricht dem Datum „29.12.1899“. Der 30. Dezember 1899 hat den Wert null. Alle vor diesem Datum liegenden Werte werden intern als negative Zahlen gespeichert.
- Nutzung des gregorianische Kalenders.

Zeitangaben

```
Dim datDatum As datZeit
```

- Zeitangaben werden in der Form Stunde:Minute:Sekunde angegeben.
- Literale werden wie Datumsangaben durch das Hash-Zeichen begrenzt.

AM / PM

- `datZeit = #12:24:45 AM#` entspricht der Zeit 00:24:45. Zeiten zwischen 00:00 und 12:00 werden mit AM gekennzeichnet.
- `datZeit = #12:24:45 PM#` entspricht der Zeit 12:24:45. Zeiten zwischen 12:00 und 00:00 werden mit PM gekennzeichnet.

Weitere Hinweise

- Gültige Uhrzeiten liegen zwischen Mitternacht 00:00:00 (0.0) bis 23:59:59 (0.99999).
- `datZeit = 1.5` entspricht der Angabe 31.12.1899 12:00:00. Zeitangaben werden als Nachkommastellen eines Datumwertes dargestellt. Jede Stunde wird als $1/24$ eines Tages abgelegt. Jede Minute wird als $1/1440$ des Tages gespeichert. Jede Sekunde wird als $1/86400$ des Tages gespeichert.

Option Explicit

Option Explicit

- Die Anweisung befindet sich immer am Anfang eines Moduls.
- Die Deklaration von Variablen wird erzwungen. Falls eine Variable nicht deklariert ist, wird die Fehlermeldung „Variable ist nicht definiert“ angezeigt.
- Die Anweisung kann manuell von Hand gesetzt werden. Andere Möglichkeit: Die Anweisung wird automatisiert bei Neuanlage des Moduls gesetzt.

... automatisiert bei Neuanlage setzen

- *Extras – Optionen.*
- Aktivierung der Registerkarte *Editor.*
- Aktivierung der Option *Variablendeklaration erforderlich.*

Initialisierung von Variablen

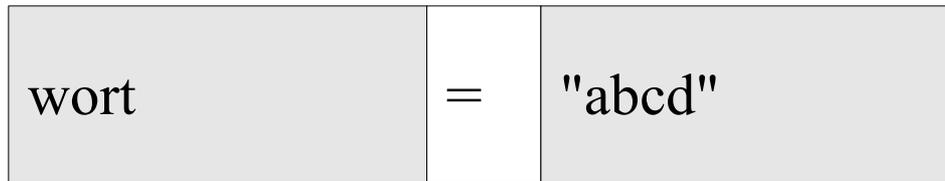
```
ganzzahl = 1  
gleitkommazahl = 0.5  
wort = "abcd"  
lichtAn = True  
datumAngabe = #1/30/2017 10:12:00 AM#
```

- Mit Hilfe des Zuweisungsoperators = wird der Variablen ein Anfangswert zugewiesen.
- Der Startwert rechts vom Gleichheitszeichen wird der Variablen links vom Gleichheitszeichen zugewiesen.

Standardwerte in Abhängigkeit des Datentyps

- Variablen für Ganz- und Gleitkommazahlen haben den Standardwert 0.
- Variablen vom Datentyp String sind leer.
- Variablen vom Datentyp Boolean haben den Wert falsch.
- Datums- und Zeitangaben haben den Standardwert 00:00:00.
Die Angabe entspricht Mitternacht am 1. Januar 0001.

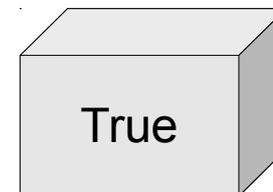
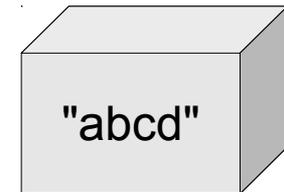
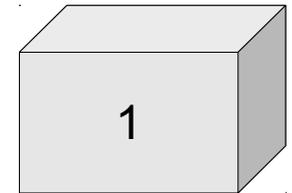
Zuweisungsoperator



- Der Variablen ... wird Wert ... zugewiesen.
- Der Wert rechts vom Operator wird der Variablen links vom Operator zugewiesen.
- Der Wert rechts kann wie in diesem Beispiel direkt angegeben werden.
- Der Wert rechts vom Operator kann mit Hilfe eines Ausdrucks berechnet werden.

Speicherung von konstanten Werten

- Der Wert wird in einer Prozedur oder Modul nicht verändert.
- Variablen werden mit einem Schreibschutz versehen.
- Der Wert der „Variablen“ kann nur gelesen werden.



... in einer Prozedur

```
Const ganzzahl As Integer = 1  
Const gleitkommazahl As Double = 3.14159  
Const wort As String = "abcd"  
Const lichtAn As Boolean = True  
Const datumAngabe As Date = #1/30/2017 10:12:00 AM#
```

- Konstanten müssen deklariert und gleichzeitig initialisiert werden.
- Der Initialisierungswert kann während der Lebensdauer einer Konstanten nicht verändert werden.

... deklarieren

Const	wort	As	String	=	"abcd"
Const	[name]	As	[datentyp]	=	[wert]

- Die Deklaration in einer Prozedur beginnt mit dem Schlüsselwort `Const`.
- Der Name der Variablen ist frei wählbar.
- `As` weist der Variablen einen Datentyp zu.

... und gleichzeitig initialisieren

Const	wort	As	String	=	"abcd"
Const	[name]	As	[datentyp]	=	[wert]

- Das Gleichheitszeichen weist der Konstanten einen Wert zu.
- Die Konstante wird initialisiert. Der Initialisierungswert entspricht dem Datentyp.
- Konstanten müssen bei der Deklaration initialisiert werden.

Systemdefinierte Konstanten

- Definition in der Programmiersprache VBA oder der Microsoft Anwendung.
- 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.

Hinweise zur Deklaration

```
Option Compare Database  
Option Explicit
```

```
Sub KreisFlaeche()  
    Const dblPi As Double = 3.1415  
  
End Sub
```

- Die Konstante ist in der Prozedur deklariert.
- Die Konstante kann nur in dieser Prozedur genutzt werden.

```
Option Compare Database  
Option Explicit  
Const dblPi As Double = 3.1415
```

```
Sub KreisFlaeche()  
  
End Sub
```

- Die Konstante ist am Anfang des Moduls deklariert.
- Die Konstante kann in jeder Prozedur in dem Modul genutzt werden.

Ausdruck

- Nach bestimmten Regeln werden Operanden und Operatoren zusammengesetzt.
- Berechnung eines Wertes mit Hilfe von Operanden und Operatoren.
- Ausdrücke können geklammert werden. Die runden Klammern erhöhen die Lesbarkeit bei komplexen Ausdrücken und verändern die Rangfolge der Operatoren.

Operanden

- Variablen, die einen dynamischen Wert speichern.
- Konstanten, die einen festen Wert speichern.
- Literale. Statische Werte, die direkt in einer Anweisung stehen. Direkte Darstellung von Zahlen, Zeichenfolgen etc. im Ausdruck.

Operatoren

- Regeln zur Berechnung von Werten.
- Verknüpfung von Operanden.
- Vorschriften zur Bildung von Ausdrücken aus mehreren Operanden.

Operatoren in VBA

- Arithmetische Operatoren berechnen mit Hilfe eines Ausdrucks einen Wert. Der Wert wird einer Variablen zugewiesen, in einer Tabelle gespeichert etc..
- Operatoren vergleichen zwei Werte und geben einen boolschen Wert zurück. Falls der Vergleich stimmt, wird true (wahr) zurückgegeben. Andernfalls wird false (falsch) zurückgegeben. Bedingte Anweisungen in VBA nutzen Vergleichsoperatoren.
- Logische Operatoren verknüpfen Ausdrücke, die einen boolschen Wert zurückgeben. Logische Operatoren wie AND, OR und NOT verknüpfen Bedingungen in Anweisungen.

Arithmetische Operatoren

Operator	Berechnung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division
\	Ganzzahlige Division
Mod	Ganzzahlige Division mit Rest
^	Potenzierung

... bei Ganzzahlen

Operator	Berechnung	Beispiel
+	Addition	$3 + 5 = 8$
-	Subtraktion	$3 - 5 = -2$
*	Multiplikation	$3 * 5 = 15$
/	Division	$3 / 5 = 1$
\	Ganzzahlige Division	$3 \setminus 5 = 0$
Mod	Ganzzahlige Division mit Rest	$3 \text{ Mod } 5 = 3$
^	Potenzierung	$3^5 = 243$

... bei Gleitkommazahlen

Operator	Berechnung	Beispiel
+	Addition	$3.5 + 5.5 = 9$
-	Subtraktion	$3.5 - 5.5 = -2$
*	Multiplikation	$3.5 * 5.5 = 19.25$
/	Division	$3.5 / 5.5 = 0,636363$
\	Ganzzahlige Division	$3.5 \setminus 5.5 = 0$
Mod	Ganzzahlige Division mit Rest	$3.5 \text{ Mod } 5.5 = 4$
^	Potenzierung	$3.5^{5.5} = 982.594307804838$

Rangfolge

$$\text{dblWertL} = 3.5$$

$$\text{dblWertR} = 5.5$$

$$\text{dblErgebnis} = \text{dblWertL} + \text{dblWertR} * \text{dblWertL}$$

$$\text{dblErgebnis} = \text{dblWertL} + (\text{dblWertR} * \text{dblWertL})$$

- Punktrechnung geht vor Strichrechnung.
- Mit Hilfe von Klammern kann die Rangfolge der Operatoren verändert werden.

Hinweise zu Datentypen von Ausdrücken

`dblWertL = 3.5`

`dblWertR = 5.5`

`dblErgebnis = dblWertL + dblWertR`

- Zwischen den Operatoren und den Operanden stehen Leerzeichen.
- Der Laufzeitfehler 13 „Typen unverträglich“ wird angezeigt, wenn der berechnete Wert nicht der Variablen zugewiesen werden kann. Der Datentyp des berechneten Wertes entspricht nicht dem Datentyp der Variablen oder kann in diesen umgewandelt werden.

Hinweise zur Division

`dblWertL = 3.5`

`dblWertR = 0`

`dblErgebnis = dblWertL / dblWertR`

`dblErgebnis = dblWertL \ dblWertR`

- Der Laufzeitfehler 11 „Division durch Null“ wird zurückgeliefert.
- Eine Division durch Null ist nicht erlaubt.

Verknüpfung von Strings

```
Const wordA As String = "Guten"  
Const wordB As String = "Tag"  
Dim gutenTag As String  
  
gutenTag = wordA & " " + wordB
```

- Mit Hilfe des kaufmännischen Unds werden Strings verknüpft.
- Es können Variablen und Literale vom Datentyp String verknüpft werden.

Verknüpfung von Strings und Zahlen

```
Const produktname As String = "Brötchen"  
Const menge As Integer = 12  
Dim bestellposten As String  
  
bestellposten = 12 & " " + produktname
```

- Zahlen, die mit Strings verknüpft werden, werden automatisiert in ein String konvertiert.

Zeilenumbruch etc.

```
Const wordA As String = "Guten"  
Const wordB As String = "Tag"  
Dim gutenTag As String  
  
gutenTag = wordA & vbCrLf + wordB
```

- Zeilenumbrüche (vbCrLf) etc. können mit Hilfe von Konstanten aus VBA abgebildet werden.
- Siehe <https://msdn.microsoft.com/en-us/library/office/gg278800.aspx>.

Kodierung von ASCII-Zeichen

```
Const wordA As String = "Guten"  
Const wordB As String = "Tag"  
Dim gutenTag As String
```

```
gutenTag = wordA & Chr(13) + Chr(10) + wordB
```

- Die Funktion Chr() wandelt eine Integer-Zahl in ein ASCII-Zeichen um.
- Dem Funktionsname Chr folgt die Parameterliste in runden Klammern. Der Funktion wird ein Parameter, die Zeichenkodierung eines ASCII-Zeichens, übergeben.

Explizite Typumwandlung

```
Dim intWert As Integer
```

```
bytWert = 4
```

```
intWert = bytWert
```

- Die Typumwandlung findet automatisch bei der Zuweisung statt.
- Umwandlung von Byte → Integer → Long oder Single → Double ist möglich.
- Boolesche Werte oder Datums- / Zeitwerte können als Zahlen interpretiert.

Hinweise zu Fehlern

```
Dim intWert As Integer
```

```
intWert = 300
```

```
bytWert = intWert
```

- Eine Umwandlung von zum Beispiel Integer → Byte liefert den Laufzeitfehler 6 „Überlauf“.
- Der Datentyp kann nicht in einen Datentypen mit einem kleineren Wertebereich konvertiert werden.

Umwandlung Gleitkommazahl in Ganzzahl

```
Dim intWert As Integer  
Dim dblWert As Double
```

```
dblWert = intWert  
intWert = dblWert
```

- Eine Ganzzahl kann in eine Gleitkommazahl explizit umgewandelt werden.
- Bei der Umwandlung von einer Gleitkommazahl zu einer Ganzzahl kann es zu einem Informationsverlust kommen. Die Nachkommastellen werden zur Ganzzahl auf- oder abgerundet.

Implizite Typumwandlung

```
Dim intWert As Integer  
Dim dblWert As Double  
  
intWert = CInt(dblWert)
```

- Zu jedem Datentyp ist eine Konvertierungsfunktion in VBA definiert.
- Konvertierungsfunktionen beginnen mit einem großen „C“.
- Siehe <https://msdn.microsoft.com/de-de/library/office/gg278896.aspx>.

Kodierung von ASCII-Zeichen als Integer

```
Dim strWert As String
Dim lngWert As Long

strWert = "A"
intWert = Asc(strWert)
```

- Die Funktion `Asc()` gibt die Kodierung eines ASCII-Zeichens zurück.
- Dem Funktionsname `Asc` folgt die Parameterliste in runden Klammern. Der Funktion wird ein Parameter übergeben. Der Parameter enthält das umzuwandelnde ASCII-Zeichen.