3 Eingebaute Funktionen

Wenn Sie eine Funktion suchen, die VBA zur Verfügung stellt, dann können Sie den Objektkatalog verwenden. Dort werden in der Bibliothek »VBA« in der Liste der Klassen alle Funktionen aufgelistet.

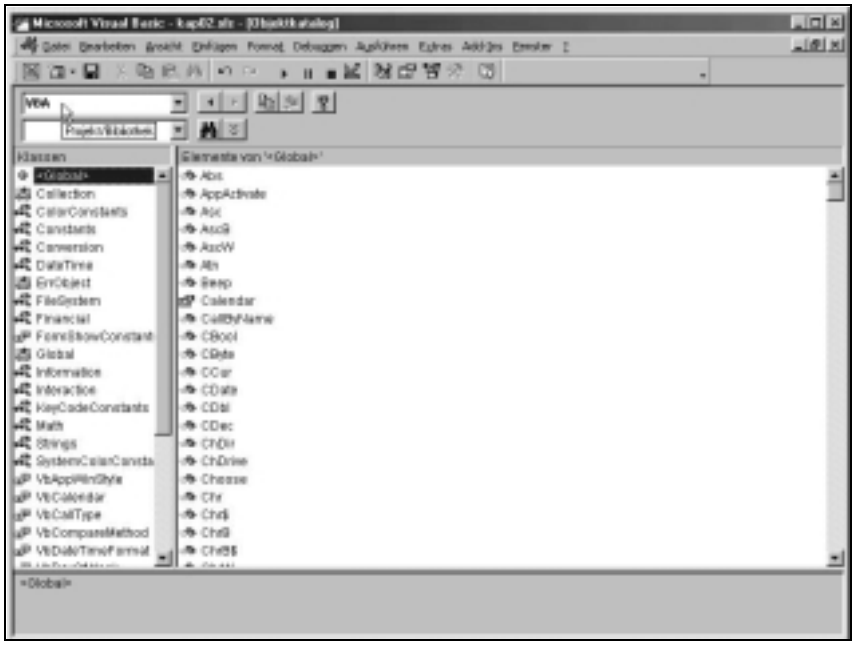

Abbildung 3.1: Der Objektkatalog

Hier nun ein Überblick über die wichtigsten Funktionen:

3.1 Die mathematischen Funktionen:

Funktions-name	Bedeutung
Sqr	Quadratwurzel
Sin	Sinus
Cos	Cosinus
Tan	Tangens
Atn	der Arkustangens, die Umkehrfunktion des Tangens
Exp	Exponentialfunktion auf Basis e

Tab. 3.1: Die mathematischen Funktionen

Funktions-name	Bedeutung
Log	der natürliche Logarithmus zur Basis e
Abs	gibt den Absolutwert einer Zahl zurück: 3 = Abs(3) 3 = Abs(-3)
Int	gibt einen Wert zurück, der den gleichen Typ wie der übergebene Wert hat und den ganzzahligen Anteil einer Zahl enthält. 8 = Int(8.4) -9 = Int(-8.4)
Fix	gibt einen Wert zurück, der den gleichen Typ wie der übergebene Wert hat und den ganzzahligen Anteil einer Zahl enthält. 8 = Fix(8.4) -8 = Fix(-8.4)
Sgn	das Vorzeichen einer Zahl: Wert von Zahl Rückgabewert von Sgn Größer als Null 1 Gleich Null 0 Kleiner als Null -1 Beispiel: 1 = Sgn(7), -1 = Sgn(-7), 0 = Sgn(0)
Round	Rundet eine Zahl auf oder ab. Beispiel: Round(2.4824, 2) ergibt 2,48; Round(2.4824, 1) ergibt 2,5
Rnd Randomize	Eine Zufallszahl Wert von Zahl generierte Zufallszahl kleiner als Null immer dieselbe Zahl, die als Startwert Zahl verwendet wird. größer als Null die nächste Zufallszahl in der Folge gleich Null die zuletzt generierte Zahl nicht angegeben die nächste Zufallszahl in der Folge Wichtig: Die Rnd-Funktion gibt einen Wert zurück, der kleiner als 1, aber größer oder gleich Null ist.

3.2 Die finanzmathematischen Funktionen:

Tab. 3.2: Die finanzmathematischen Funktionen

Funktions-name	Funktions-name in Excel	Bedeutung
DDB	GDA	die Abschreibung eines Anlageobjekts nach der geometrisch degressiven Methode für einen spezifischen Zeitraum
SYD	DIA	die Abschreibung eines Anlageobjekts nach der arithmetisch degressiven Methode für einen spezifischen Zeitraum
SLN	LIA	die Abschreibung eines Anlageobjekts nach der linearen Methode für einen spezifischen Zeitraum

Funktions-name	Funktions-name in Excel	Bedeutung
FV	ZW	der Endwert einer Annuität ausgehend von regelmäßigen Zahlungen und einem konstanten Zinssatz
RATE	ZINS	der Zinssatz einer Annuität
IRR	IKV	der interne Zinsfluss für regelmäßige Cash-Flows
MIRR	QIKV	der modifizierte interne Zinsfluss für regelmäßige Cash-Flows
IPMT	ZINSZ	der Zinsanteil einer Annuität für einen spezifischen Zeitraum
PMT	RMZ	die Zahlung einer Annuität
PPMT	KAPZ	der Kapitalanteil einer Annuität
NPV	NBW	der Netto-Barwert für regelmäßige Cash-Flows
PV	BW	der Barwert einer Annuität

3.3 Die String-Funktionen:

Tab. 3.3: Die String-Funktionen

Funktions-name	Bedeutung	Beispiel
Left$	schneidet eine bestimmte Anzahl von Zeichen von links ab.	Left$("Heinrich Hoffmann", 4) ergibt "Hein"
Right$	schneidet eine bestimmte Anzahl von Zeichen von rechts ab.	Right$("Heinrich Hoffmann ", 4) ergibt "mann"
Mid$	schneidet eine bestimmte Anzahl von Zeichen aus der »Mitte« heraus, das heißt, ab einer bestimmten Position.	Mid$("Heinrich Faust", 5, 4) ergibt "rich Hoffmann"
InStr	überprüft, ob eine Zeichenfolge innerhalb einer Zeichenkette vorhanden ist und gibt die Position an.	InStr("Heinrich Hoffmann", "ff") ergibt 12 InStr("Heinrich Hoffmann", "h") ergibt 8 InStr("Heinrich Hoffmann", "öy") ergibt 0
Ltrim$	löscht Leerzeichen am Anfang eines Strings.	LTrim(" Heinrich Hoffmann ") ergibt ("Heinrich Hoffmann ")
Rtrim$	löscht Leerzeichen am Ende eines Strings.	RTrim(" Heinrich Hoffmann ") ergibt (" Heinrich Hoffmann")
Trim$	löscht Leerzeichen am Anfang und Ende eines Strings.	Trim(" Heinrich Hoffmann ") ergibt ("Heinrich Hoffmann ")
Len	ermittelt die Länge einer Zeichenkette.	Len("Heinrich Hoffmann") ergibt 17

Funktions-name	Bedeutung	Beispiel
Chr	wandelt einen ASCII-Code in einen String um.	Chr(13) ergibt ¶
Asc	wandelt einen String in die entsprechende Zahl des ASCII-Codes um.	Asc("A") ergibt 65
Str$	wandelt eine Zahl in einen String um.	Str$(65) = "65"
Val	wandelt einen String in eine Zahl um.	Val("65") = 65
Lcase	wandelt eine Zeichenkette in Kleinbuchstaben um.	Lcase("Heinrich Hoffmann") ergibt "heinrich hoffmann"
Ucase	wandelt eine Zeichenkette in Großbuchstaben um.	Ucase("Heinrich Hoffmann") ergibt "HEINRICH HOFFMANN"
StrConv	wandelt eine Zeichenkette um.	StrConv("heinrich hoffmann", vbProperCase) ergibt "Heinrich Hoffmann". Ebenso stehen die beiden Parameter vbLowerCase und vbUpperCase zur Verfügung
StrComp	»vergleicht« zwei Zeichenketten, das heißt, es wird überprüft, welche zuerst im Alphabet steht. Ist die erste Zeichenkette »größer« als die zweite, wird -1 zurückgegeben, im umgekehrten Fall 1. Sind beide gleich: 0. Ist einer der beiden Strings leer, so wird Null übergeben.	StrComp("Heinrich", "Heinrich") ergibt 0 StrComp("Heinrich", "Hoffmann") ergibt – 1 StrComp("Robert", "Heinrich") ergibt 1
Space	gibt eine Folge von Leerzeichen aus.	"Heinrich" & Space(2) & "Hoffmann" ergibt "Heinrich Hoffmann"
String	wiederholt eine Zeichenfolge.	"Heinrich" & String("/", 2) & "Hoffmann" ergibt "Heinrich//Hoffmann"
Split	trennt eine Zeichenfolge und liefert einen Array.	Split("Heinrich Hoffmann") ergibt "Heinrich" "Hoffmann"
Join	setzt eine Zeichenfolge zusammen.	
Filter	durchsucht eine Zeichenfolge.	
InStrRev(»abcd«, »ä«)	überprüft, ob eine Zeichenfolge in einer anderen vorhanden ist.	InStrRev("Heinrich Hoffmann", "ä") ergibt 0 ("falsch"), InStrRev("Heinrich Hoffmann", "a") ergibt 1 ("wahr")

3.4 Die Uhrzeit- und Datumsfunktionen

Tab. 3.4: Die Uhrzeit- und Datumsfunktionen

Funktionsname	Bedeutung	
Date	setzt das aktuelle Systemdatum ein oder stellt das Systemdatum um.	
Now	gibt das Systemdatum und die aktuelle Systemzeit zurück.	
Timer	gibt einen Wert vom Typ Single zurück, der die Anzahl der seit Mitternacht vergangenen Sekunden angibt. Diese Funktion wird verwendet, wenn Zeitdifferenzen berechnet werden sollen.	
Time	setzt die aktuelle Systemzeit ein oder stellt die Systemzeit um.	
DateSerial	gibt die fortlaufende Datumszahl eines Datums zurück: DateSerial (Year, Month, Day).	
DateValue	gibt das Datum eines String-Arguments zurück.	
TimeSerial	gibt einen fortlaufenden Zeitwert für eine aus Stunden, Minuten und Sekunden bestehenden Uhrzeit zurück.	
TimeValue	wandelt einen String in eine Uhrzeit um.	
DateAdd	addiert oder subtrahiert ein angegebenes Intervall zu einem oder von einem Datum. Syntax: DateAdd(Intervall, Anzahl, Datum) Dabei wird das Intervall als String ausgegeben (vergleiche DateDiff).	
DateDiff	gibt die Anzahl der Zeitintervalle zurück, die zwischen zwei Datumsangaben liegen Syntax: DateDiff(Intervall, Date1, Date2[, FirstDayofWeek] [, FirstDayofYear]	
	Intervall	wird als String angegeben. Es bedeuten:
	D	Tag
	Y	Kalendertag
	W	Wochentag
	WW	Woche
	M	Monat
	Q	Quartal
	YYYY	Jahr
	S	Sekunde
	N	Minute
	H	Stunde
	Date1, Date2	die beiden Datumsangaben, deren Differenz berechnet werden soll
	FirstDayOfWeek	gibt den ersten Wochentag an. Ohne Angaben wird Sonntag als erster gesetzt, sonst:

Funktionsname	Bedeutung		
	VbUseSytem	0	Einstellung der Applikation
	VbSunday	1	Sonntag
	VbMonday	2	Montag
	VbTuesday	3	Dienstag
	VbWednesday	4	Mittwoch
	VbThursday	5	Donnerstag
	VbFriday	6	Freitag
	VbSaturday	7	Samstag
	FirstWeekofYear		gibt die erste Woche des Jahres an. Ohne Angabe wird die Woche verwendet, die den 1. Jan. enthält. (Wichtig zur Berechnung von Kalenderwochen!)
	VbUseSystem	0	Einstellung der Applikation
	VbFirstJan1	1	Woche mit 1. Januar
	VbFirstFourDays	2	Woche mit mindestens vier Tagen des neuen Jahres (zur Berechnung von Kalenderwochen (laut DIN 1355))
	VbFirstFullWeek	3	erste komplette Woche im neuen Jahr
DatePart	berechnet, zu welchem Teil eines angegebenen Intervalls ein Datum gehört: DatePart(Intervall, Date [, FirstDayofWeek] [, FirstDayofYear] Die Zahlen und Variablen entsprechen denen von DateDiff.		
Day	filtert den Tag aus einem Datum.		
Month	filtert den Monat aus einem Datum.		
Year	filtert das Jahr aus einem Datum.		
Weekday	gibt eine Zahl zwischen 1 und 7 zurück, die dem Wochentag entspricht: Weekday(Date, [FirstDayofWeek]) Dabei entsprechen Date einem Datum und FirstDayofWeek der gleichen Variable wie bei DateDiff. Der zurückgegebene Wert ist ebenso eine Zahl von 1 bis 7 oder von vbSunday bis vbSaturday		
Hour	filtert die Stunde aus einer Uhrzeit.		
Minute	filtert die Minutenanzahl aus einer Uhrzeit.		
Second	filtert die Sekundenanzahl aus einer Uhrzeit.		

3.5 Die Funktion Format

`Format(Ausdruck[,Format[,firstdayofweek[,firstweekofyear]]])`

Ihre Argumente:

Tab. 3.5: Die Funktion Format und Zahlen

Zeichen	Beschreibung	Beispiel
Kein Zeichen	zeigt die Zahl ohne Formatierung an.	
0	beliebige Ziffer	Format(1234, "0") liefert 1234 Format(1234, "0.00") liefert 1234,00 Format(1234.5678, "0") liefert 1235 Format(1234.5678, "0.00") liefert 1234,57
#	Platzhalter für eine Ziffer, die nur angezeigt wird, wenn sich an dieser Stelle eine Ziffer befindet, gedacht für Tausendertrennzeichen.	Format(1234, "0") liefert 1234 Format(1234, "#,##0") liefert 1.234 Format(123, "#,##0") liefert 123
. und ,	Der Punkt dient als Trennzeichen für Dezimalzeichen, das Komma für Tausendertrennzeichen. Also umgekehrt als im Deutschen!	Siehe oben
%	multipliziert die Zahl mit 100 und fügt ein %-Zeichen an.	Format(0.15, "#,##0.00%") liefert 15,00% Format(0.125, "0.00 %") liefert 12,50 %
E- E+ e- e+	wissenschaftliche Zahlenschreibweise	Format(1250000, "0.00 E+00") liefert 1,25 E+06 Format(1250000, "0.00 E-00") liefert 1,25 E06 Format(1250000, "0.00 e+0") liefert 1,25 e+6 Format(0.125, "0.00 e-0") liefert 1,25 e-1 Format(0.000125, "0.00 E-00") liefert 1,25 E-04
+, – Leerzeichen und $	können zur Darstellung direkt in die Formatierung eingefügt werden.	Format(1234, "#,##0.00 $") liefert 1.234,00 $
\	Das nächste Zeichen wird als Zeichen und nicht als Formatierung ausgegeben. Das »\« verschwindet in der Anzeige.	Format(1234, "#,##0.00 \L\i\r\e") liefert 1.234,00 Lire

Es können bis zu vier verschiedene Zahlenformate in Abschnitten ausgegeben werden. Dabei bedeuten:

Tab. 3.6: Die Funktion Format und Zahlen und Abschnitte

Zeichen	Beschreibung	Beispiel
nur ein Abschnitt	alle Werte	
zwei Abschnitte	Der erste Abschnitt bezieht sich auf positive Werte und die Null, der zweite auf negative.	Format(123, "0;(0)") liefert 123
drei Abschnitte	Wie zwei Abschnitte; der dritte Abschnitt bezieht sich auf die Null.	Format(-123, "0;(0);\N\i\x") liefert (123)
vier Abschnitte	Wie drei Abschnitte; der vierte Abschnitt bezieht sich auf Null-Werte.	Format(0, "0;(0);\N\i\x") liefert Nix

Datumsformatierungen:

Tab. 3.7: Die Funktion Format und Datumsformatierungen

Zeichen	Beschreibung	Beispiel
/	Datumstrennzeichen	
d	zeigt den Tag als Zahl.	Format("2.3.2000", "d/m/yy") liefert 2.3.00
dd	zeigt den Tag als Zahl und Tage zwischen 1 und 9 mit führender Null an.	Format("2000", "dd/mm/yyyy") liefert 02.03.2000
ddd	zeigt den Wochentag als Abkürzung.	Format("2.3.2000", "ddd") liefert So
dddd	zeigt den Wochentag ausgeschrieben.	Format("2 / 3 / 2000", "dddd, \d\e\n dd. mmmm. yyyy") liefert Sonntag, den 02. März 2000
ddddd	zeigt das vollständige Datum gemäß der Systemsteuerung im Kurzformat an (dd.mm.yy).	Format("2 / 3 / 2000", "ddddd") liefert 02.03.00
dddddd	zeigt das vollständige Datum gemäß der Systemsteuerung im Langformat an (dd.mmmm.yyyy).	Format("2 / 3 / 2000", "dddddd ") liefert Sonntag, 02 März 2000
w	der Wochentag als Zahl (1 = Sonntag, 7 = Samstag	Format("2.3.2000", "w") liefert 5

Zeichen	Beschreibung	Beispiel
ww	die Kalenderwoche (1–54)	Format("2.3.2000", "ww") liefert 10, was falsch ist. Die korrekte Funktion für Europa muss lauten: Format("2.3.2000", "ww", , vbFirstFourDays)
m	der Monat als Zahl (1–12)	
mm	der Monat als Zahl. Monate zwischen 1 und 9 mit führender Null	
mmm	Monat als Text mit drei Buchstaben (Jan–Dez)	
mmmm	vollständiger Monatsname	
q	Quartal	Format("2.3.2000", "q") liefert 1
yy	Jahr als zweistellige Zahl (00-99)	
yyyy	Jahr als vierstellige Zahl (100-9999)	

Die Zeitangaben:

Zeichen	Beschreibung	Beispiel
:	Zeittrennzeichen	
h	Stunde als Zahl ohne führende Null	Format(»2:4«, »h:m«) liefert 2:4
hh	Stunde als Zahl mit führender Null	Format(»2:4«, »hh:mm«) liefert 02:04
n oder m	die Minute ohne führende Null	Format(»15:4«, »hh:mm«) liefert 15:04
nn oder mm	die Minute mit führender Null	
s	die Sekunden ohne führende Null	
ss	die Sekunden mit führender Null	
ttttt	die vollständige Zeitangabe	Format(»2:4«, »ttttt«) liefert 02:04:00
AM/PM, am/pm, A/P, A/p, AMPM	verschiedene 12-Stunden-Formate	

Tab. 3.8: Die Funktion Format und Zeitformatierungen

Zeichenformatierungen:

Tab. 3.9: Die Funktion Format und Zeichenformatierungen

Zeichen	Beschreibung	Beispiel
@	Platzhalter für ein Zeichen. Ist die Variable leer, wird ein Leerzeichen ausgegeben.	Format("Sonnenschein", "@") liefert "Sonnenschein"
&	Platzhalter für ein Zeichen. Ist die Variable leer, wird ein Nichts ausgegeben.	Format("Sonnenschein", "&") liefert "Sonnenschein"
!	Auffüllen aller Platzhalter von links nach rechts.	Format("Sonnenschein", "!") liefert "Sonnenschein"
<	zeigt den Text in Kleinbuchstaben.	Format("Sonnenschein", "<") liefert "sonnenschein"
>	zeigt den Text in Großbuchstaben.	Format("Sonnenschein", ">") liefert "SONNENSCHEIN"

3.6 Umwandlungsfunktionen

Tab. 3.10: Umwandlungsfunktionen

Funktion	Rückgabetyp	Bereich des Arguments Ausdruck
Cbool	Boolean	Eine gültige Zeichenfolge oder ein gültiger numerischer Ausdruck
Cbyte	Byte	0 bis 255
Ccur	Currency	-922.337.203.685.477,5808 bis 922.337.203.685.477,5807.
Cdate	Date	Ein beliebiger gültiger Datumsausdruck.
CDbl	Double	-1,79769313486231E308 bis -4,94065645841247E-324 für negative Werte; 4,94065645841247E-324 bis 1,79769313486232E308 für positive Werte.
CDec	Decimal	+/-79.228.162.514.264.337.593.543.950.335 für skalierte Ganzzahlen, d.h. Zahlen ohne Dezimalstellen. Für Zahlen mit 28 Dezimalstellen gilt der Bereich +/-7,9228162514264337593543950335. Die kleinste mögliche Zahl ungleich Null ist 0,0000000000000000000000000001.
CInt	Integer	-32.768 bis 32.767; Nachkommastellen werden gerundet.
CLng	Long	-2.147.483.648 bis 2.147.483.647; Nachkommastellen werden gerundet.
CSng	Single	-3,402823E38 bis -1,401298E-45 für negative Werte; 1,401298E-45 bis 3,402823E38 für positive Werte.

Funktion	Rückgabetyp	Bereich des Arguments Ausdruck
Cvar	Variant	Numerische Werte im Bereich des Typs Double. Nicht-numerische Werte im Bereich des Typs String.
CStr	String	Rückgabe für CStr hängt vom Argument Ausdruck ab.

3.7 Übungen zu den mathematischen Funktionen

Übung 1

Berechnen Sie Ostersonntag! Die Lösung dieses Problems stammt vom Mathematiker, Astronom und Physiker Carl Friedrich Gauß, und sieht wie folgt aus: Die Jahreszahl sei J und J – 1900 sei a. Der Rest von a/19 wird schlicht b genannt. Jetzt wird vom Ausdruck (7*b+1)/19 der ganzzahlige Quotient genommen, der c genannt wird. Mit d wird der Rest von (11*b+4-c)/29 bezeichnet und der Quotient von a/4 mit e. Dann bleibt noch der Rest von (a+e+31-d)/7. Und dieser soll f genannt werden. Daraus folgt, dass für das Osterdatum April die Formel 25 – d – f gilt.

Soll beispielsweise von 1999 der Ostersonntag berechnet werden, so ergeben sich folgende Werte:

- J = 1999
- a = 99
- b = REST(99;19) = 4
- c = QUOTIENT(7*4+1;19) = 1
- d = REST(11+4+4-1;29) = 18
- e = QUOTIENT(99;4) = 24
- f = REST(99+24+31-18;7) = 3
- Ostern =DATUM(1999;4;25-18-1) = 04.04.1999

Analog für das Jahr 2000:

- a = 100; b = 5; c = 1; d = 0; e = 25; f = 2; Ostern = 23.04.2000

Übung 2

Die Funktion Round rundet nur Stellen nach dem Dezimalzeichen. Lassen Sie mit ihrer Hilfe vor dem Komma runden, also beispielsweise 1.234.567 ergibt auf 2 Stellen vor dem Komma 1.234.600 und auf 6 Stellen: 1.000.000.

3.8 Übungen zu den Textfunktionen

Übung 3

Schreiben Sie eine Prozedur, die einen Namen mit Vornamen und Zunamen in Vor- und Zunamen zerlegt.

Übung 4

Schreiben Sie eine Prozedur, die aus einem Dateipfad den Dateinamen (ohne Endung) herausliest.

Übung 5

Schreiben Sie eine Prozedur, die überprüft, ob von einem eingegebenen Text das erste Zeichen eine Ziffer ist.

Übung 6

Der Benutzer oder die Benutzerin wird aufgefordert, ein »m« oder ein »w« für das Geschlecht einzugeben. Gibt er oder sie weder ein »w«, »W«, »m« noch ein »M« ein, dann wird er oder sie darauf hingewiesen.

3.9 Übungen zu den Datumsfunktionen

Übung 7

Der Benutzer gibt sein Geburtsdatum ein und erhält als Ergebnis sein Alter.

Übung 8

Der Benutzer gibt sein Geburtsdatum ein. Daraufhin wird überprüft, ob er Geburtstag hat (oder wie viele Tage bis zu seinem Geburtstag fehlen), an welchem Wochentag er Geburtstag hatte und an welchem Wochentag er im aktuellen Jahr Geburtstag hatte oder haben wird.

3.10 Übungen zu den Formatfunktionen

Übung 9

In Übung 1 des Kapitels 2 (»Operatoren, Verknüpfungen und Verzweigungen«) wird die Hypothenuse aus der Länge zweier Katheten berechnet. Formatieren Sie die Ausgabe so, dass das Ergebnis immer auf 4 Stellen nach dem Komma erscheint.

Übung 10

Ein Benutzer gibt seinen Arbeitsbeginn und sein Arbeitsende ein. Daraus wird die Differenz berechnet. Für jede Stunde erhält er 15,75 Euro. Schließlich wird sein berechneter Arbeitslohn angezeigt.

3.11 Tipps

Tipp zu Übung 3

Wenn Sie mit der Funktion `InStr` arbeiten, dann achten Sie darauf, dass sie ein bestimmtes Zeichen von links ermittelt. Was für den Vornamen problemlos funktioniert, klappt für den Nachnamen nicht! Neben `InStr` stehen Ihnen allerdings auch noch andere Funktionen zur Verfügung.

Tipp zu Übung 7

Achtung: Um das Alter zu bestimmen, dürfen Sie nicht einfach die Jahreszahl des heutigen Datums von der Jahreszahl des Geburtsdatums abziehen. Wenn heute beispielsweise der 20. Oktober 2000 ist, der Benutzer allerdings 10. November 1980 eingibt, so ist er nicht 20 Jahre, sondern nur 19. Auch die Differenz in Tagen geteilt durch 365,25 kann zu Ungenauigkeiten führen.

3.12 Lösungen

Lösung 1

```
Sub Ostersonntag()
Dim intJahreszahl As Integer
Dim a As Integer, b As Integer, c As Integer
Dim d As Integer, e As Integer, f As Integer
intJahreszahl = InputBox("Bitte eine Jahreszahl eingeben!")
a = intJahreszahl - 1900
b = a Mod 19
c = (7 * b + 1) \ 19
d = (11 * b + 4 - c) Mod 29
e = a \ 4
f = (a + e + 31 - d) Mod 7
MsgBox Format(DateSerial(intJahreszahl, 4, 25 - d - f), _
   "dd.mm.yyyy")
End Sub
```

Lösung 2

```
Sub RundenVorKomma()
Dim dblZahl As Double
Dim intRundZahl As Integer

dblZahl = InputBox("Bitte eine Zahl eingeben!")
intRundZahl = InputBox("Auf wie viele Stellen vor dem " & _
   "Komma soll gerundet werden?")

dblZahl = dblZahl / 10 ^ intRundZahl
dblZahl = Round(dblZahl, 0)
dblZahl = dblZahl * 10 ^ intRundZahl
MsgBox dblZahl
End Sub
```

Lösung 3

```
Sub NamenZerlegen1()
Dim strGanzName As String
Dim strVorname As String
Dim strNachname As String

strGanzName = InputBox("Wie heißen Sie?", "Vor- und Zuname")
If strGanzName = "" Then
   MsgBox "Sie haben nichts eingegeben!"
ElseIf InStr(strGanzName, " ") = 0 Then
   MsgBox "Sie haben nur einen Namen eingegeben!"
   Exit Sub
End If
strVorname = Left$(strGanzName, InStr(strGanzName, " ") - 1)

strNachname = Right$(strGanzName, Len(strGanzName) - _
   InStr(strGanzName, " "))

MsgBox "Der Vorname lautet: " & strVorname & _
   " und der Nachname lautet: " & strNachname
End Sub
```

Der Zuname kann auch anders herausgelöst werden:

```
strNachname = Right(strGanzName, _
   Len(strGanzName) - Len(strVorname))
```

Oder so:

```
strNachname = Mid$(strGanzName, InStr(strGanzName, " ") + 1)
```

VBA 6.0 stellt neue String-Funktionen zur Verfügung: `Split`, `Join`, `Filter` und `InStrRev`. Mit der Funktion `Split` kann die Aufgabe ebenso gelöst werden:

```
Sub NamenZerlegen3()
Dim strTeilnamen() As String
Dim strGanzName As String
strGanzName = InputBox("Wie heißen Sie?", "Vor- und Zuname")
strTeilnamen = Split(strGanzName)

If strTeilnamen(UBound(strTeilnamen)) = _
   strTeilnamen(0) Then
   MsgBox "Der Vorname lautet: " & strTeilnamen(0)
Else
   MsgBox "Der Vorname lautet: " & strTeilnamen(0) & _
      " und der Nachname lautet: " & _
      strTeilnamen(UBound(strTeilnamen))
End If
End Sub
```

Umgekehrt setzt die Funktion `Join` einen String wieder zusammen, wobei sie ebenfalls einen Array verlangt. Im folgenden Beispiel werden die einzelnen Komponenten wieder zusammengefügt:

```
Sub NamenZerlegenUndZusammensetzen()
Dim strTeilnamen() As String
Dim i As Integer
Dim strGanzName As String

strGanzName = InputBox("Wie heißen Sie?", "Alle Namen")
strTeilnamen = Split(strGanzName)

For i = 0 To UBound(strTeilnamen)
   strGanzName = Join(strTeilnamen, "")
Next

MsgBox strGanzName
End Sub
```

Lösung 4

Die folgenden Befehle können nur in Word verwendet werden. In Excel heißt der vollständige Dateiname `ActiveWorkbook.FullName`.

```
Sub DateiNamenZerlegen1()
Dim strDateiName As String
Dim i As Integer

strDateiName = ActiveDocument.FullName
MsgBox strDateiName
```

```
For i = Len(strDateiName) To 1 Step -1
   If Mid(strDateiName, i, 1) = "\" Then
      strDateiName = Right(strDateiName, _
         Len(strDateiName) - i)
      Exit For
   End If
Next i

strDateiName = Left(strDateiName, _
   InStr(strDateiName, ".") - 1)
MsgBox strDateiName
End Sub
```

Ein andere Variante (ohne Schleife) sieht folgendermaßen aus:

```
Sub DateiNamenZerlegen2()
Dim strDateiName As String
Dim strTeilNamen() As String

strDateiName = ActiveDocument.FullName
MsgBox strDateiName
strTeilNamen = Split(strDateiName, "\")
strDateiName = strTeilNamen(UBound(strTeilNamen))
strDateiName = Left(strDateiName, _
   InStr(strDateiName, ".") - 1)
MsgBox strDateiName
End Sub
```

Lösung 5

```
Sub ZifferPrüfung()
Dim strEingabeText As String

strEingabeText = InputBox("Bitte einen Text eingeben")
If IsNumeric(Left(strEingabeText, 1)) Then
   MsgBox "Ziffer"
Else
   MsgBox "Zeichen"
End If

End Sub
```

Lösung 6

```
Sub Geschlechterprüfung1()
Dim strGeschlecht As String
strGeschlecht = InputBox("Bitte ein ""m"" oder ein ""w""" & _
   "für das korrekte Geschlecht eingeben!")
If strGeschlecht <> "m" And strGeschlecht <> "M" And _
```

```
   strGeschlecht <> "w" And strGeschlecht <> "W" Then
   MsgBox "Es wurde weder ""m"" noch ""w"" eingegeben."
   Exit Sub
End If

End Sub
```

Man kann ebenso die eingegebene Zeichenkette in Kleinbuchstaben verwandeln:

```
Sub Geschlechterprüfung2()
Dim strGeschlecht As String
strGeschlecht = InputBox("Bitte ein ""m"" oder ein ""w""" & _
   "für das korrekte Geschlecht eingeben!")
If LCase(strGeschlecht) <> "m" And _
   LCase(strGeschlecht) <> "w" Then
   MsgBox "Es wurde weder ""m"" noch ""w"" eingegeben."
   Exit Sub
End If
End Sub
```

Die Bedingung kann auch folgendermaßen abgefragt werden:

```
If Format(strGeschlecht, "<") <> "m" And _
   Format(strGeschlecht, "<") <> "w" Then
```

Da die Anzahl der Möglichkeiten überschaubar sind, kann auch mit einer `Select Case`-Anweisung gearbeitet werden:

```
Sub Geschlechterprüfung4()
Dim strGeschlecht As String
strGeschlecht = InputBox("Bitte ein ""m"" oder ein ""w""" & _
   "für das korrekte Geschlecht eingeben!")
Select Case strGeschlecht
Case "w", "W", "m", "M"
Case Else
   MsgBox "Es wurde weder ""m"" noch ""w"" eingegeben."
End Select
End Sub
```

Lösung 7

```
Sub Alter_in_Jahren()
Dim datGebdatum As Date
Dim intAlter As Integer
datGebdatum = InputBox("Wann bist du geboren?")
intAlter = Year(Date) - Year(datGebdatum)
If Month(Date) < Month(datGebdatum) Then
   intAlter = intAlter - 1
Else
```

```
   If Month(Date) = Month(datGebdatum) Then
      If Day(Date) < Day(datGebdatum) Then
         intAlter = intAlter - 1
      End If
   End If
End If
MsgBox "Sie sind " & intAlter & " Jahre alt. "
End Sub
```

Oder noch eleganter:

```
Sub Alter_in_Jahren2()
Dim datGebDatum As Date
Dim intAlter As Integer
datGebDatum = InputBox("Wann bist du geboren?")
intAlter = Year(Date) - Year(datGebDatum)
If DateSerial(Year(Date), Month(datGebDatum), _
   Day(datGebDatum)) > Date Then
      intAlter = intAlter - 1
End If
MsgBox "Sie sind " & intAlter & " Jahre alt. "
End Sub
```

Beide Lösungen ziehen vom aktuellen Jahr das Geburtsjahr ab. Die Differenz muss allerdings noch nicht das Lebensalter sein, denn wenn der Benutzer im laufenden Jahr noch nicht Geburtstag hatte, dann muss von der Differenz 1 abgezogen werden. Das erste Programm überprüft nun Geburtsmonat und Geburtstag, während das zweite Programm aus Geburtstag, Geburtsmonat und aktuellem Jahr ein Datum zusammensetzt und dies mit dem heutigen Tag vergleicht. In beiden Lösungen wird die Zahl 1 subtrahiert.

Lösung 8

```
Sub Geburtstag()
Dim intTag As Integer
Dim intMonat As Integer
Dim datGebDatum As Date
Dim intAnzahlTage As Integer
Dim strZukunft As String

datGebDatum = InputBox("Wie lautet Ihr Geburtsdatum?")

intTag = Day(datGebDatum)
intMonat = Month(datGebDatum)

If intTag = Day(Date) And intMonat = Month(Date) Then
   MsgBox "Happy Birthday"
   strZukunft = "Heute ist "
Else
```

```
   intAnzahlTage = DateDiff("D", _
      Date, DateSerial(Year(Date) + 1, intMonat, intTag))

   If intAnzahlTage >= 365 Then
      intAnzahlTage = DateDiff("D", _
         Date, DateSerial(Year(Date), intMonat, intTag))
         strZukunft = " Geburtstag haben."
   End If

   MsgBox "Sorry, Sie haben erst in " & intAnzahlTage & _
   " Tagen Geburtstag."

End If

MsgBox "Sie sind an einem " & _
   Format(datGebDatum, "DDDD") & " geboren."

If strZukunft = " Geburtstag haben." Then
   MsgBox "Sie werden in diesem Jahr an einem " & _
      Format(DateSerial(Year(Date), intMonat, intTag), _
         "DDDD") & strZukunft
ElseIf strZukunft = "Heute ist " Then
   MsgBox "Heute ist Ihr Geburtstag." & _
      vbCr & strZukunft & Format(Date, "DDDD")
Else
   MsgBox "Sie hatten in diesem Jahr an einem " & _
      Format(DateSerial(Year(Date), intMonat, intTag), _
         "DDDD") & " Geburtstag."
End If
End Sub
```

Lösung 9

```
Sub RechtwinklDreieck()
Dim dblSeiteA As Double
Dim dblSeiteB As Double
dblSeiteA = InputBox("Bitte die Länge der ersten Kathete!")
dblSeiteB = InputBox("Bitte die Länge der zweiten Kathete!")
MsgBox "Die Länge der Hypothenuse beträgt " & _
   Format(Sqr(dblSeiteA ^ 2 + dblSeiteB ^ 2), "0.0000")
End Sub
```

Lösung 10

```
Sub Lohnberechnen()
Dim datBeginn As Date
Dim datEnde As Date
Dim datArbeitszeit As Date
```

```
Const curLohn = 15.75

datBeginn = InputBox("Bitte den Arbeitsbeginn eingeben!")
datEnde = InputBox("Bitte das Arbeitsende eingeben!")

datArbeitszeit = datEnde - datBeginn

MsgBox "Sie erhalten für Ihre " & _
   Format(datArbeitszeit, "hh:mm") & _
   " gearbeiteten Stunden " & _
   Format(datArbeitszeit * curLohn * 24, "0.00 \E\u\r\o")

End Sub
```