

Arbeitsblatt

Name:

Klasse / Kurs:

Fach:

Datum:

RI2010/11

Anzahl der Seiten: 3



- Datentypen
- Variable / Konstante
- Operand / Operator

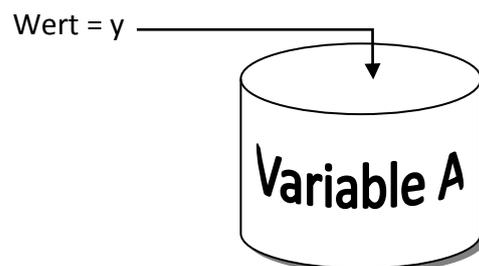
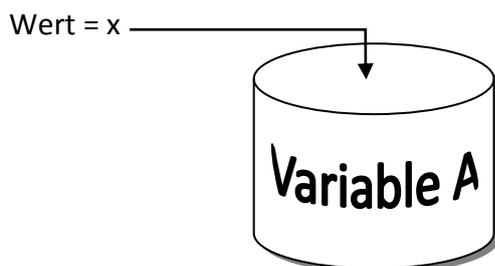
Datentypen

Datentyp	Abk.	Verwendung	Speicherbedarf
string	str	Zeichenkette (z. B. Buchstaben) – beliebige ASCII-Zeichen	10 Byte
boolean	bln	true oder false	2 Byte
Byte	byt	Ganze Zahl zwischen 0 und 255	1 Byte
integer	int	ganze positive und negative Zahlen(von -32768 bis +32767)	2 Byte
long	lng	„lange“ ganze positive und negative Zahlen (von -2.147.483.648 bis +2.147.483.647)	4 Byte
single	sng	Fließkommazahl (Dezimalzahl) mit 8 Stellen Genauigkeit	4 Byte
double	dbl	Fließkommazahl (Dezimalzahl) mit 16 Stellen Genauigkeit	8 Byte
currency	cur	Dezimalzahlen (Währungsberechnungen) 4-fache Genauigkeit – 15 Stellen vor und vier Stellen nach dem Dezimalpunkt- für z. B. kaufm. Berechnungen	8 Byte
Date	dtm	Datumsangaben	8 Byte
variant	vt	„Chamäleon“ – kann für nahezu alle Datentypen verwendet werden. Der Datentyp sollte wegen des hohen Speicherbedarfs vermieden werden.	16 - 22 Byte
Object	obj	Verweist auf ein Objekt	4 Byte

Variable

Unter einer „Variablen“ versteht man einen Ausdruck, dem ein Wert zugewiesen wird. Dieser Wert ist von einem bestimmten Datentypen.

Das spezifische einer Variablen ist nun, dass dieser Wert im Verlauf des Programms einmal oder mehrmals geändert werden kann.



Beispiel:

„Dim Int_Zahl1 as Integer“

bedeutet, dass eine Variable festgelegt wird, die den Namen „Int_Zahl1“ erhält und vom Datentyp „Integer“ ist. Diese Variable kann nun im Laufe des Programms verschiedene Werte erhalten, die aber alle mit dem Datentyp „Integer“ übereinstimmen müssen.

Variablendeklarationen werden immer mit dem Wort „Dim“ eingeleitet.

Nach den syntaktischen Regeln in VBA darf der Bezeichner einer Variablen (= Variablenname) keine Leerzeichen und keine Satzzeichen enthalten und muss mit einem Buchstaben beginnen.

Um einer Variablen Werte oder das Ergebnis von Berechnungen zuzuweisen, wird der Zuweisungsoperator“ = „ nach dem Muster „Variablenname = Wert“ verwendet, z. B.

```
cent = 700
cent = cent + 20 (cent hat jetzt den Wert 720)
geld = cent + 50 (geld hat den Wert 770)
```

Es handelt sich dabei um eine in VBA zwar mögliche, aber keine empfehlenswerte Vorgehensweise, da es sehr fehleranfällig sein kann, mit Variablen zu arbeiten, die vorher hinsichtlich ihres Typs nicht festgelegt (nicht deklariert) wurden, d. h. für die zunächst kein Speicherplatz reserviert wurde.

Empfehlenswert ist daher die vorherige **Deklaration von Variablen**:

Gründe für eine Deklaration:

1. Ohne Deklaration wird automatisch der Datentyp **Variante** zugewiesen. Dieser Datentyp benötigt viel Speicherplatz und macht die Programme in vielen Fällen langsam.
2. Bei Schreibfehlern von Variablennamen reagiert ein Programm unterschiedlich:
 - Mit Deklaration: Fehlermeldung – der Schreibfehler ist markiert – sofortige Korrektur möglich
 - Ohne Deklaration: Das Programm reagiert nicht – es passiert nichts – der Fehler muss mühsam selbst erkannt und korrigiert werden.

Konstante

Manchmal werden in einem Programm aber auch Werte benötigt, die sich nicht verändern. Hier sind dann einmalige Wertzuweisungen erforderlich.

Konstantendeklarationen beginnen immer mit dem Wort „Const“.

Beispiel:

„Const byt_MWSt_1 as Byte = 19“

Bedeutet, dass einer Konstanten mit dem Namen „Byt_MWSt_1“, die vom Datentyp „Byte“ ist, die Zahl 19 fest zugeordnet wird.

Um die Deklaration von Konstanten und Variablen zu erzwingen, muss im VBA-Fenster unter

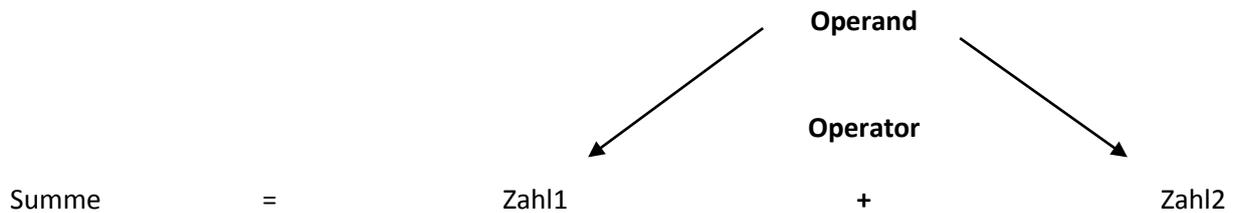
EXTRAS – OPTIONEN – EDITOR

das Feld "Variablendeklaration erforderlich" markiert sein.

(Im Programmfenster erscheint dann bei Neustart der Datei die Zeile "Option Explicit")

Operand / Operator

Beispiel:



Ein Operand kann sein: -
 - eine Variable
 - eine Konstante
 - eine Funktion

Operatoren

Folgende Operatoren können in VBA verwendet werden:

Rang	Operatoren	
arithmetische Operatoren	-	negatives Vorzeichen
	+, -, *, /	Grundrechenarten
	^	Potenz
	\	Integerdivision *
	Mod	Modulo-Operator *
Zeichenketten-Operatoren	+ &	verbindet Zeichenketten, Zahlen werden vor der Verbindung in Zeichenketten umgewandelt
Vergleichsoperatoren	=	gleich
	<>	ungleich
	<, <=	kleiner, kleiner gleich
	>, >=	größer, größer gleich
Logische Operatoren	And	logisches und
	Or	logisches oder
	Xor	exklusives oder (entweder a oder b, aber nicht beide)
	Not	logische Negation
Zuweisungsoperator	=	Zuweisung

*

- **Beispiel** für eine **Integerdivision (\)**: Hierbei wird zunächst $10 : 8$ als Integerdivision gerechnet, damit wir wissen, wie oft die 8 in die 10 hineinpasst.
 $10 \setminus 8 = 1$
- **Beispiel** für eine **Modulo-Division (Mod)**: Hierbei wird der restliche Wert von $10 : 8$ ausgegeben.
 $10 \text{ Mod } 8 = 2$