

PureBasic-Grundlagen

```
1 1. Wertzuweisung, Typen, Typumwandlung
2 =====
3 a = 2      ; Die Variable ist ein Speicherplatz mit dem Namen a (Adresse)
4            ; und dem Inhalt 2. Stelle dir eine Schachtel mit der Aufschrift a vor.
5 b = 5      ; Kommentare beginnen mit einem Semikolon.
6 c = a+b
7 a = a+b
8 Ergebnis = Zahl1 + Zahl2 ; rechts vom Gleichheitszeichen wird gerechnet, links gespeichert
9            ; auf der linken Seite steht nur eine Variable
10           ; Variablennamen sind sinnvoll zu wählen
11           ; Groß- und Kleinschreibung ist zu beachten

12 Variablen-Typen ; Es gibt große und kleine Schachteln, die Unterschiedliches aufnehmen.
13
14 a.l Long      ; ganze Zahl, Bereich -2147483648 bis +2147483647
15 a.d Double    ; Kommazahl (doppelte Genauigkeit)
16 a.s String    ; Zeichenkette
17              ; Typ nur beim ersten Auftreten festlegen (deklarieren)
18              ; weitere Typen siehe Hilfe/Allgemeine Themen/Variablen, Typen ...

19 OpenConsole( )
20   PrintN("Flächeninhalt eines Dreiecks") ; für den Text sind Anführungszeichen erforderlich
21
22   Grundseite.s = Input( )              ; Ein- und Ausgabe stets als Zeichenkette
23   Hoehe.s = Input( )
24   a.d = ValD(Grundseite)                ; Zeichenkette wird in eine Kommazahl umgewandelt (value)
25   b.d = ValD(Hoehe)
26   Inhalt.d = a*b/2
27
28   Ausgabe.s=StrD(Inhalt)                ; möglich auf 3 Stellen mit StrD(Inhalt,3)
29   PrintN("")
30   PrintN(Ausgabe)
31   Input( )                              ; Programmausführung stoppt
32                                         ; sofortiges Schließen des Fensters wird so verhindert

33 ; Beim Compilieren wird der Programmtext in Anweisungen übersetzt,
34 ; die der Prozessor des Computers versteht.
35 ; Bei einem Fehler verwende Debugger/Programm beenden.
```

```

1 Typumwandlung
2 bei der Eingabe
3 a.s=Input( )
4 b.d=ValD(a) ; Zeichenkette wird in eine Kommazahl umgewandelt
5 b.l=Val(a) ; Zeichenkette wird in eine ganze Zahl umgewandelt
6
7 bei der Ausgabe
8 b.s=Str(c) ; c (Typ ganze Zahl) wird (für die Ausgabe) in eine Zeichenkette umgewandelt
9 Print(b)
10
11 e.d=1.1237
12 b.s=StrD(e) ; e (Typ Kommazahl) wird in eine Zeichenkette umgewandelt
13 b.s=StrD(#PI)
14 Print(b)

```

```

15 OpenConsole( )
16 a.s="Informatik"
17 PrintN("Erste Schritte in " + a) ; Zeichenketten werden mit + verbunden
18 Eingabe.s=Input( )
19 Zahl.l=Val(Eingabe)
20 Potenz2.l=Zahl*Zahl
21 Ausgabe1.s=Str(Potenz2)
22 PrintN("2. Potenz " + Ausgabe1)
23
24 Potenz3.l=zahl*zahl*zahl ; möglich Potenz3.l=Pow(zahl,3)
25 Ausgabe2.s=Str(Potenz3)
26 PrintN("3. Potenz " + Ausgabe2)
27
28 Potenz4.l=zahl*zahl*zahl*zahl ; möglich Potenz3.l=Pow(zahl,4)
29 Ausgabe3.s=Str(Potenz4)
30 PrintN("4. Potenz " + Ausgabe3)
31 Input( )

```

32 2. For-Schleife

```

33 =====
34 OpenConsole( )
35 zahl.l = 3 ; Kleinschreibung üblich
36 For n = 0 To 15 ; für n ist eine Typangabe überflüssig
37 PrintN(Str(n) + ".te Potenz " + Str(Pow(zahl,n)))
38 Next n
39 Input( )

```

```

1  OpenConsole( )
2  summe.l = 0
3  For n = 1 To 10
4  summe = summe + n*n
5  Next n
6  PrintN(Str(summe))
7  Input( )

```

```

8  For n = Anfangswert To Endwert      ; n ist die Laufvariable
9  ...
10 Next n

```

```

11 For n = 1 To 600 Step 20           ; die Schrittweite kann vorgegeben werden
12 ...
13 Next n

```

14 3. If-Abfrage

```

=====
15 If a < 5 And b >= 0
16 PrintN("...")
17 EndIf

```

```

18 If a<5 Or c>=2
19 ...
20 Else
21 ...
22 EndIf

```

23 4. Array

```

24 =====

```

```

25 Dim Feld.l(10)
26 ; hiermit werden 11 Variablen Feld(0), Feld(1), Feld(2), ..., Feld(10) vom Typ Long erstellt
27
28 Feld(0) = 1      ; Wertzuweisung
29 Feld(20) = 2

```

```

1           ; mit einer For-Schleife kann der Array oder ein Teil davon durchlaufen werden
2   OpenConsole( )
3       Dim Feld.l(10)
4
5       For n = 1 To 10
6           Feld(n) =Random(100)
7           Print( Str( Feld(n) )+" " )
8       Next n
9       Input( )

```

10 5. Prozeduren

```

11 =====
12 Procedure.f name( )
13     a.f = ...
14     ...
15     ProcedureReturn a
16 EndProcedure
17
18 Aufruf mit c = name( )

```

```

19 Procedure.l Maximum(a.l, b.l)
20     If a > b
21         Ergebnis.l = a
22     Else
23         Ergebnis = b
24     EndIf
25     ProcedureReturn Ergebnis
26 EndProcedure

```

```

27
28 Aufruf mit c = Maximum (d, e)

```

```

29
30 Global c.l für Variablen, die im ganzen Programm gültig sein sollen.

```

Grafik

```
1 Width = 800
2 Height = 600

3 OpenWindow(0, 0, 0, Width, Height, "Grafik1", #PB_Window_SystemMenu |
4                                           #PB_Window_ScreenCentered)
5
6 StartDrawing(WindowOutput(0))

7     LineXY(10,20,450,300,RGB(0, 0, 0))
8     Box(150,300,20,50, RGB(0, 0, 255))
9     radius=40
10    Circle(200, 100, radius, RGB(Random(255),Random(255),Random(255)))

11
12 Repeat
13 Until WaitWindowEvent( ) = #PB_Event_CloseWindow
14 StopDrawing( )
```

- 15 Die Zeilen 1 bis 6 und 12 bis 14 sind für unsere Grafik-Programme stets gleich.
16 Mit diesen Anweisungen wird ein Grafik-Fenster geöffnet, das dann auch geschlossen werden kann.

17	LineXY(x1,y1,x2,y2, Farbe)	zeichnet eine Linie von (x1,y1) nach (x2,y2)
18	RGB(r,g,b)	Farbe, r_rot, g_rün, b_lau, Farbwerte aus dem Bereich 0 ... 255
19	Random(100) (z.B.)	erzeugt eine Zufallszahl aus dem Bereich 0 ... 100
20	Box(x,y,Breite,Höhe,Farbe)	erzeugt ein Rechteck, linke obere Ecke: (x,y)
21	Circle(x,y, Radius,Farbe)	zeichnet einen Kreis mit dem Mittelpunkt (x,y)
22	Delay(40) (z.B.)	verzögert die Programmausführung