41.1 LUA Grundlagen - Variablen

Autor: Goetz
Quelle: Mein EEP-Forum

In einem Programm muss man sich oft etwas merken.
Beispielsweise eine Zahl, die man ausgerechnet hat.
Um diese Zahl später wieder zu benutzen.

Für diesen Zweck gibt es Variablen.
Diese Variablen können einen Namen bekommen, den man selber festlegt.
Man kann sie zum Beispiel **John** nennen.
Oder **Paul**
Oder **George**
Oder **Ringo**

Genauso gut gehen
Gleis
Zug
Besetzt
oder auch **Vergissmeinnicht**

Aber der Name muss immer mit einem Buchstaben beginnen!
Und er darf nur normale Buchstaben, Ziffern und den Unterstrich enthalten. **Sonst nichts!**
Denn alle Sonderzeichen haben in Lua Aufgaben.
Beispielsweise der Bindestrich. Der wird als Minus benutzt

Gibt man einer Variablen den Namen **John-Paul**, dann zieht Lua **Paul** von **John** ab.
Ob bei dieser Subtraktion etwas sinnvolles entsteht ist Lua völlig egal.
Da steht ein Minus in der Zeile, also wird gerechnet. Punkt!

Wenn ich am Anfang des Lua Skripts schreibe:
John = 3
dann bedeutet das für Lua:
„Immer, wenn der Götz **John** schreibt, dann meint er 3“

Steht also weiter hinten im Skript der Befehl
EEPSetSwitch(**John**, 2)
dann wird die Weiche 3 in die Stellung 2 (also Abzweig) gebracht.

Apropos:

Ich kann auch am Anfang des Skripts schreiben:
Fahrt = 1
Abzweig = 2

Dann kann ich später im Skript diese Worte für die Weichenstellungen benutzen und so mein Skript etwas lesbarer machen.

Ich kann also später schreiben:
EEPSetSwitch(**John**, **Fahrt**)
und Lua weiß, dass damit gemeint ist:
Weiche 3 in Stellung 1 bringen.

Weil EEP sich am Anfang gemerkt hat, dass **John** 3 bedeutet und **Fahrt** 1.

Die meisten Programmierer machen sich solche Mühe nicht. Weil es umständlich und unnötig ist.
Aber es zeigt, was eine Variable überhaupt ist.

Weil in großen Programmen viele Variablen benötigt werden und man deshalb viele Namen benötigt, wird jeder Unterschied im Namen als eine andere Variable angesehen

John ist also eine andere Variable als **john** oder **JOHN** oder **jOhn**

Ein einzelner Buchstabe reicht als Variablenname.
Ich kann sie also beispielsweise
Z
nennen.

Und **Z** ist eine andere Variable als **Z_BUE**.
Die zwei haben nichts miteinander zu tun.
Ich kann mir in **Z** eine Zahl merken und in **Z_BUE** eine andere.

Der Wert, der in einer Variablen gespeichert wird, ist veränderbar.
Deshalb nennt man diese kleinen Speicherplätze Variable. Weil sie genau das sind: variabel.

Wenn am Anfang des Skripts stand
Gleis = 1
und ich etwas später im Programm schreibe
Gleis = 2
dann ist ab jetzt mit **Gleis** nicht mehr die Zahl 1, sondern 2 gemeint.
Bis ich den Wert erneut überschreibe.

Mit Variablen kann ich auch rechnen, als ob es Zahlen wären.
Ich kann beispielsweise
Gleis + 1 ausrechnen
Wenn **Gleis** den Wert 2 gespeichert hat, dann bedeutet die Rechenaufgabe oben
2+1
und das Ergebnis ist 3

Wenn ich dieses Ergebnis anschließend benutzen will, dann muss ich es wieder in einer Variablen speichern.
John = **Gleis** + 1
heißt nimm den Wert, der in **Gleis** gespeichert ist, addiere Eins hinzu und speicher das Ergebnis in der Variablen **John**.

Ich kann auch schreiben
Gleis = **Gleis** + 1
was bedeutet:
Nimm den alten Wert, der in **Gleis** gespeichert war, addiere 1 hinzu und Speicher das Ergebnis wieder in der Variablen **Gleis** ab.

Zuletzt noch eine sehr **wichtige Information**, die ich beinahe vergessen hätte:

In einer Programmier- oder Skriptsprache gibt es viele Befehle.
Und das sind auch nur Buchstaben.

Wie kann Lua also unterscheiden, ob ein Befehl gemeint ist oder eine Variable?
Ganz einfach:
Sämtliche Befehle, die es in Lua gibt, sind als Name für eine Variable streng verboten!
Da alle Befehle (meines Wissens) nur aus Kleinbuchstaben bestehen ist man auf der sicheren Seite, wenn der Name einer Variablen mindesten einen Großbuchstaben enthält.

Damit ist die erste Erläuterung zu Variablen beendet. Es gäbe zwar noch mehr zum Thema zu sagen. Aber das hebe ich mir aus gutem Grunde für später auf.

Im nächsten Kapitel werde ich Funktionen erläutern.