

41.5 LUA Grundlagen - spezielle Operatoren - das Prozent Zeichen

Autor:Goetz

Quelle: Mein EEP-Forum

In verschiedenen Threads wurde schon nach der Bedeutung des % Zeichens gefragt und darauf gab es ein paar gute, mathematische Erklärungen.

Aber Mathe ist nicht jedermanns Sache. Deshalb möchte ich ergänzend einen anderen Weg versuchen.

Stellt euch eine Uhr vor.

Eine Uhr, die nur einen Stundenzeiger hat, der von einer Stunde zur nächsten springt.



Wenn diese Uhr auf 1 steht und man 3 dazu zählt, dann steht sie anschließend auf 4.

Steht die Uhr aber auf 10 und man rechnet 3 hinzu, dann steht die Uhr anschließend nicht auf 13, sondern auf 1.

Weil sie nach 12 immer wieder von vorne beginnt.

Es ist völlig egal, wie viel Stunden vergehen - die Uhr zeigt immer nur einen Wert zwischen 1 und 12.

Und so ist der % Operator gedacht.

Nur, dass man ihm sagen kann, wie viele Stunden auf der Uhr stehen sollen

13 % 12 ergibt 1 (wie bei der Uhr)

13 % 10 ergibt 3, denn diese Uhr hat nur 10 Stunden

23 % 10 ergibt ebenfalls 3

23 % 5 ergibt ebenfalls 3. Die Uhr hat nur 5 Stunden. Aber wenn ich bei 0 Uhr anfangen und 23 Stunden dazu zähle, dann lande ich auch hier bei 3 Uhr. Ich komme nämlich vier mal rum und habe dann noch drei Stunden übrig.

Und nun komme ich doch mit Mathematik:

Wie in einem anderen Thread schon erläutert wurde, kann ich 20 sauber durch 5 teilen. Es bleibt ein Rest von 3.

Aber einen wichtigen Unterschied zur richtigen Uhr gibt es:

Oben steht nicht die 12 (beziehungsweise der Wert, den man rechts vom % Zeichen schreibt), sondern 0

10 % 10 ergibt also 0 und nicht 10

% 5 erlaubt also alle Werte von 0 bis 4.

Aus der 5 wird wieder die 0.

Aus der 6 die 1

Aus 7 wird 2

etc.

Im Skript zum ersten Lua Tutorial steht

```
route % 2
```

Das Ergebnis kann nur 0 oder 1 sein

Genauer betrachtet steht dort

```
route % 2 == 1
```

Das Ergebnis von **route** % 2 wird also mit der Zahl 1 verglichen.

Wenn das Ergebnis von **route** % 2 den Wert 1 hat, dann ist das Ergebnis des Vergleichs "wahr".

Weil die Variable in der Zeile

```
route = route + 1
```

bei jedem Durchgang weitergezählt wird, ist das Ergebnis von

```
route % 2
```

immer abwechselnd 0 oder 1. Und damit ist das Ergebnis des Vergleichs immer abwechselnd "wahr" oder "falsch"

Hinter

```
route % 2
```

verbirgt sich also ein einfacher Trick, um immer abwechselnd mal so und mal so zu entscheiden.