



Schöner programmieren Teil 5 – Kürzen und kopieren

Description

Nicht nur beim Einrücken und beim Aufbau stellt man fest, dass es verschiedene Wege gibt, etwas zu programmieren. Als Anfänger neigt man dazu, manches umständlicher zu machen als es sein muss, als Fortgeschrittener neigt man dazu, zu viel zu kürzen. Ein Klassiker bei Anfängern ist folgende Abfrage:

```
if (zeug == true)
{
```

oder

```
if (zeug == false)
{
```

Einfacher geht:

```
if (zeug)
{
```

```
var RGW = Math.round((((Math.pow(rRAD, 2) * Math.PI * Breite) - (Math.pow(rRID
```

oder

```
if (!zeug)
{
```

Das spart nicht nur Arbeit und damit Zeit, sondern wird auch allgemein so gehandhabt, kann also im Prinzip jeder Programmierer lesen. Die zweite Frage die sich stellt ist die, ob man die Klammern um die Abfrage mit schreibt oder nicht, schließlich würde

```
if zeug
{
```

ebenso funktionieren. Es funktioniert sogar, wenn man mehrere Abfragen hintereinander bringt:

```
if !zeug || 8 + 2 = 10  
{
```

Allerdings erkennt man hier vielleicht schon das Problem: Je mehr Abfragen es werden, umso unübersichtlicher wird es ohne die Verwendung von Klammern, auch wenn das obere Beispiel ein wenig schwachsinnig ist.

Bei Berechnungen im Code rate ich eher dazu ab, möglichst wenig zu kürzen.

```
time = 60 * 60 * 24;
```

oder

```
time = 86400;
```

Das Ergebnis ist genau gleich, aber wer weiß noch genau, woher die Zahl 86400 kam? Wer die Meinung vertritt, dass das Spiel dadurch einen Bruchteil einer Nanosekunde schneller ist und das ganz wichtig sei, sollte dahinter wenigstens einen Kommentar lassen, damit man versteht, was sich hinter der Zahl verbirgt. Das kann auch dann nicht schaden, wenn man die Berechnung ausschreibt. Zur Veranschaulichung eine Zeile aus einem JavaScript-Code von mir: