

+ - * /

```
int x = 42;
```

```
int y = 1337;
```

```
int z = x + y;
```

```
int z = x - y;
```

```
int z = x * y;
```

```
int z = x / y; y sollte nicht 0 sein, sonst NaN
```

Es gilt natürlich zu beachten, dass kein Überlauf im Speicher entsteht

Exponenten

```
int z = x ** y; oder int z = x ^ y, oder
```

```
int z = Math.Pow(x, y);
```

Modulo

```
int x = 5;
```

```
int y = 3;
```

```
int z = x % y;
```

Nur der Rest der ganzzahl Division wird in z reingeschrieben. Wird bei Grafikprogrammierung (Sprites) verwendet

$S:3=1$ $R=2$ also ist $Z=2$

Hier kann also immer nur ein Rest von 0, 1 o. 2 bleiben

Inkrement / Dekrement

$++$; o. $y = y + 1$;

Fügt der aktuellen Variable etwas hinzu.

Es geht auch mit sämtlichen anderen arithmetischen Operatoren.

$--$; o. $y = y - 1$;

$*$; o. $y = y * 1$;

$/$; o. $y = y / 1$;

$\%$; o. $y = y \% 1$;

Wenn man nur um eins erhöhen bzw. verringern will, gibt es noch folgende Schreibweisen

$++$; u. $--$;

Es gibt auch die Schreibweise wo der Variablenname hinten steht, was bedeutet das wenn der Wert beim Zuweisen erhöht wird der erhöhte Wert zugewiesen wird und nicht der jetzige.

```
int x = 5;  
int y = 3;  
int z = y++;  
// z = 3, y = 4  
int w = ++x;  
// int x = 6, w = 6
```