

2 Variablen

Variablen sollten immer mit `let` oder früher mit `var` erstellt werden. Es funktioniert aus historischen Gründen auch ohne, wird auch vom Browser unterstützt, wie möchten aber aktuelles valides JavaScript benutzen.

var, let oder const?

`Var` ist eine veraltete Möglichkeit Variablen zu definieren und sollte nicht mehr verwendet werden. Deshalb wird `let` benutzt.

`Let` hat einen anderen Gültigkeitsbereich.

„let allows you to declare variables that are limited to the scope of a block statement, or expression on which it is used, unlike the var keyword, which declares a variable globally, or locally to an entire function regardless of block scope. The other difference between var and let is that the latter can only be accessed after its declaration is reached (see temporal dead zone).“ (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let?retiredLocale=de> aufgerufen am 9.2.2023)

`const` definiert konstante Variablen, deren Wert nicht verändert werden kann. Mit `const` definierte Variablen müssen einen Wert haben.

→ Variable erstellen und ausgeben.

```
// Variablen definieren
let greeting;
greeting = "Hallo Welt!!";
console.log(greeting);
// gleich Wert zuweisen
let greeting2 = "Wie geht's?"
```

Regeln

- Variablen dürfen nicht mit einer Zahl anfangen.
- Variablen dürfen keinen Punkt enthalten.
- Unterstrich geht
- Minus ist z. B. nicht erlaubt
- Editor unterstützt, Fehler werden angezeigt

→ Vergleichen Sie folgende drei Programme, lassen Sie sie laufen.

→ Variablen ohne `let` können benutzt werden SOLLTEN aber nicht mehr benutzt werden.

→ Deshalb benutzen wir immer „`use strict`“

```
1)
greeting = "Hallöchen";
console.log(greeting);
```

```
2)
"use strict"
greeting = "Hallöchen";
console.log(greeting);
```

Ältere JavaScript Interpreter ignorieren die Angabe „`use strict`“ einfach.

```
3)
"use strict"
let greeting = "Hallöchen";
console.log(greeting);
```

3 Datentypen

Sie kennen von anderen Programmiersprachen schon Datentypen. Testen Sie einfach oder - vollziehen Sie einfach nach.

```
"use strict"
```

```
// String
let a = 'Hallo Welt!'
console.log(typeof a)
console.log(a)

// Zahlen
let b = 42.5
console.log(typeof b)
console.log(b)

// Booleans (true / false)
let c = true
console.log(typeof c)
console.log(c)
```

3.1 Datentypen und Plus

Wahrscheinlich sind Ihnen alle Ausgaben rechts unmittelbar klar. Mit Zahlen wird gerechnet, Strings werden aneinandergesetzt usw.

```
"use strict"
```

```
//console.log(5 + 6)
// console.log("5" + "6")
```

Eine häufige Fehlerquelle ist aber die Angabe rechts. Testen Sie, und versuchen Sie zu erklären.

```
// Welches Problem entsteht hier?
let water = "1.9"
let amount = 10 + water
console.log(amount)
```

(JavaScript interpretiert plötzlich beide Zahlen (10 und 1.9) als Strings und verkettet sie, so dass das Ergebnis 101.9 ergibt.

```
// console.log("Hallo, " + "JavaScript")

// console.log(true + true)
```

Plus kann für mathematische Rechnungen als auch für die Verkettung von Strings stehen.

→ **Übungsaufgabe: Gruppe1_Aufgabe_Plus_Operator.js**