



Programmieren
Starten

Zusammenfassung

Wichtige Grundbausteine von JavaScript

JavaScript - Dynamische Webseiten entwickeln



EINLEITUNG VARIABLEN

Variablen

Eine Variable wird verwendet, um einen Wert zu speichern. Sie besteht bei der **Initialisierung**, also der Erstellung, aus drei Teilen. Einem **Keyword**, das die Variable definiert, einem **Variablennamen**, den man selbst vergeben kann und einem dazugehörigen **Wert**, der über das =-Zeichen zugewiesen wird.

Datentypen

Der **Datentyp** einer Variable wird je nach Wert zugewiesen. Beispiele für Datentypen sind String, Integer und Boolean.

Beispiel einer Variable, die einen String (= Zeichenkette) als Wert enthält:

```
let name = "Jannick";
```

Beispiel einer Variable, die einen Integer (= Ganzzahl) als Wert enthält:

```
let age = 18;
```

Beispiel einer Variable, die einen Float (= Gleitkommazahl) als Wert enthält:

```
let price = 3.99;
```

Beispiel einer Variable, die einen Boolean (= Wahrheitswert) als Wert enthält:

```
let isWaiting = false;
```

ARBEITEN MIT VARIABLEN

Wenn man eine Variable definiert, kann man im Nachhinein noch mehrmals ihren Wert ändern.

Beispiel:

```
let x = 5;  
x = 10;
```

// Die Variable x enthält nun den Wert 10.

Man kann einer Variable auch den Wert einer anderen Variable zuweisen.

Beispiel:

```
let x = 5;  
let y = x;
```

// Die Variable y enthält nun den Wert 5.

WICHTIG

Der Code wird von oben nach unten ausgeführt.

RECHNEN MIT VARIABLEN

Mathematische Operatoren

Mithilfe mathematischer Operatoren (+, -, /, etc.) kann im Code gerechnet werden.

Beispiel:

```
let x = 5;  
let y = 10;  
  
let sum = x + y;           //addieren  
let sub = x - y;           //subtrahieren  
let div = x / y;           //dividieren  
let multiply = x * y;      //multiplizieren
```

WICHTIG

Es gilt dabei die Punkt-vor-Strich-Regelung.

Beispiel:

```
let func = x + y * 5;      //Ergibt 55  
let func = (x + y) * 5;    //Ergibt 75
```

OPERATOREN FÜR EINE VERKÜRZTE SCHREIBWEISE

Bestimmte Operatoren können dazu verwendet werden, um **Werte** anhand einer **verkürzten Schreibweise** zu **verändern**. Im folgenden Beispiel soll der Wert der Variable x um 10 erhöht werden.

Beispiel:

```
let x = 20;
```

```
x = x + 10;           //unverkürzte Schreibweise  
x += 10;             //verkürzte Schreibweise => ergibt 30
```

Mithilfe der verkürzten Schreibweise können genauso gut andere Operationen vorgenommen werden:

```
x -= 10;   //x = x - 10; => ergibt 10  
x *= 10;   //x = x * 10; => ergibt 200  
x /= 10;   //x = x / 10; => ergibt 2
```

Auf diese Art und Weise wird zuerst die Operation durchgeführt und anschließend der neue Wert, der dabei herauskommt, der Variable zugewiesen.

Erhöhen um 1

Mit dem sogenannten **Inkrementoperator (++)** kann ein Wert um 1 erhöht werden.

Beispiel:

```
let x = 4;  
x++;   //Erhöht x um 1, d. h. 4 + 1 = 5
```

Wenn man x auf die längere Schreibweise um 1 erhöhen würde, sähe das Ganze so aus:

```
x = x + 1;
```

Vermindern um 1

Mit dem sogenannten **Dekrementoperator (--)** kann ein Wert um 1 vermindert werden.

Beispiel:

```
let x = 4;  
x--;      //Vermindert x um 1, d. h. 4 - 1 = 3
```

Wenn man x auf die längere Schreibweise um 1 vermindern würde, sähe das Ganze so aus:

```
x = x - 1;
```

VERGLEICHOPERATOREN

Die Vergleichsoperatoren ermöglichen es, Werte miteinander zu vergleichen. Damit kann festgestellt werden, ob eine Bedingung wahr oder falsch ist.

Praxisbeispiel:

Der Türsteher vor einem Club überprüft auf dem Personalausweis, ob man schon über 18 Jahre alt ist. Die Antwort darauf kann entweder „Ja“ oder „Nein“ lauten. Es wird also das Alter 18 als Wert genommen und mit dem Alter auf dem Personalausweis verglichen.

Genauso kann in JavaScript das Ergebnis eines Vergleichs entweder „true“ oder „false“ sein. Dabei handelt es sich um sogenannte Wahrheitswerte. Der Datentyp für Wahrheitswerte wird **Boolean** genannt.

WICHTIG

Ein **=-Zeichen (=)** weist einen Wert zu.

Zwei **=-Zeichen (==)** vergleichen zwei Werte miteinander.

==-Operator (Gleich)

Dieser Operator überprüft, ob zwei Werte **gleich** sind. Wenn das der Fall ist, wird der Wert true zurückgegeben, andernfalls der Wert false.

Beispiel 1:

```
let x = 18;  
console.log(x == 18);    //Diese Bedingung ist ein „boolescher Ausdruck“,  
                           der „true“ ergibt
```

Beispiel 2:

```
let name = "jannick";  
console.log(name == "jannick");    //Auch diese Bedingung ergibt „true“
```

!=-Operator (Ungleich)

Dieser Operator überprüft, ob zwei Werte **ungleich** sind. Wenn sie also nicht gleich sind, wird der Wert `true` zurückgegeben. Sind sie allerdings gleich, wird `false` zurückgegeben.

Beispiel 1:

```
let x = 18;
console.log(x != 18);    //Diese Bedingung ergibt „false“
```

Beispiel 2:

```
let name = "jannick";
console.log(name != "jannick");    //Auch diese Bedingung ergibt „false“
```

Beispiel 3:

```
let name = "karl";
console.log(name != "jannick");    //Diese Bedingung ergibt „true“
```

>-Operator und <-Operator (größer als, kleiner als)

Mithilfe dieser Operatoren wird überprüft, ob ein Wert **größer** bzw. **kleiner** als der gegenüberstehende Wert ist.

Beispiel:

```
let x = 10;
let y = 20;

console.log(x > y);    //false
console.log(x < y);    //true
console.log(y > x);    //true
console.log(y < x);    //false
```

>=-Operator und <=-Operator (größer oder gleich, kleiner oder gleich)

Mit diesen Operatoren kann überprüft werden, ob ein Wert **größer** bzw. **kleiner** als der gegenüberstehende Wert ist **ODER** er diesem **gleich**.

Beispiel:

```
var z = 18;
console.log(z >= 18);    //true
```

```
console.log(z <= 18);    //true  
console.log(z <= 16);    //false
```