

# Conditional Statements - Bedingungen

Ein Conditional Statement ist eine Bedingung. Nur wenn diese Bedingung zutrifft, dann wird der Code ausgeführt.

Es gibt zwei verschiedene Arten von Conditional Statement, die Switch-Bedingung und die häufiger verwendete If-Bedingung.

## If-Bedingung vs Switch-Bedingung

Bei der Switch-Bedingung muss der ausführende Code für jeden Fall neu definiert werden, wobei nur genau dieser Fall eintreten kann (also nur wenn z.B. `var a = 1`).

Bei der If-Bedingung wird der Code einmal definiert, er kann für mehrere Fälle ausgeführt werden (also z.B. `var a >= 0`, also für alle Zahlen ab 0 [0,1,2,3,4,5,6,...]).

Außerdem können bei der If-Bedingung mehrere Bedingungen kombiniert werden und mit dem Und-Operator `&&` (alle Bedingungen müssen zutreffen, damit der Code ausgeführt wird) oder dem Oder-Operator `||` (eine Bedingung muss zutreffen, damit der Code ausgeführt wird) verbunden werden.

## If-Bedingung

Syntax:

`if (Bedingung) {Code}`

```
var ifVar = 1;
if (ifVar > 0) {
  console.log(`ifVar ist größer als 0`);
}
```

// prüft ob `ifVar > 0` ist, was in diesem Fall zutrifft, und gibt dann auf der Console aus:  
`ifVar ist größer als 0`

Bei der Prüfung ob die Bedingung gleich ist, muss in der Klammer mehr als ein Gleichheitszeichen eingesetzt werden, ansonsten wird der Variablen `ifVar` ein neuer Wert zugeschrieben!!!

```
var ifVar2 = 1;
if (ifVar2 = 0) {
  console.log(`Falscher Syntax!!!`);
}
```

// gibt auf der Console aus: `Falscher Syntax!!!`

**// falscher Syntax, hier wird `ifVar` gleich 0 gesetzt und nur überprüft, ob die Überschreibung funktioniert hat. Wenn der Prozessor die Überschreibung richtig ausgeführt hat, dann wird die Bedingung gleich wahr gesetzt und der Code ausgeführt!!!**

Mathematische Operatoren für den Vergleich:

```
> // größer als
>= // größer oder gleich als
< // kleiner als
<= // kleiner oder gleich als
== // gleich    Es wird nur der Inhalt aber nicht der Datentyp geprüft,
                somit ist 7 = "7" wahr!
=== // gleich   Es wird der Inhalt und der Datentyp geprüft, somit ist 7 = "7" falsch!
!= // ungleich  Es wird nur der Inhalt aber nicht der Datentyp geprüft,
                somit ist 7 != "8" wahr! weil der Wert nicht übereinstimmt!
!== //         Es wird der Inhalt und der Datentyp geprüft, somit ist 7 != "8" ebenso
                wahr! weil weder der Wert und der Datentyp nicht übereinstimmt.
```

Mehrere Vergleiche können in einer Bedingung mithilfe des Und-Operators && oder des Oder-Operators || in eine Bedingung gesetzt werden.

Syntax:

If (Bedingung && Bedingung) {Code}

```
var ifVar2 = 10;
```

```
if (ifVar2 >= 0 ifVar2 && <= 100) {
  console.log(`ifVar2 ist ${ifVar2} und somit zwischen 0 und 100.`)
}
```

// prüft ob ifVar2 >= 0 **und** ob ifVar2 <= 100 ist, was in diesem Fall zutrifft, und gibt dann auf der Console aus:

ifVar2 ist 10 und somit zwischen 0 und 100.

Bei dem && Operator müssen beide Bedingungen wahr sein.

Syntax:

If (Bedingung || Bedingung) {Code}

```
var ifVar3 = 100;
var ifVar4 = 200;
```

```
if (ifVar3 >= 0 || ifVar4 <= 100) {
  console.log(`ifVar3 ist ${ifVar3} und somit größer als Null.`)
}
```

// prüft ob ifVar3 >= 0 **oder** ob ifVar4 <= 100 ist, die erste Bedingung trifft zu, somit wird der Code ausgeführt und auf der Console wird ausgegeben:

ifVar3 ist 100 und somit größer als Null.

Bei dem || Operator muss eine Bedingung wahr sein.

## If-Else-Bedingung

Syntax:

```
if (Bedingung) {Code}
    else {Code}
```

```
var ifVar5 = 99;
```

```
if (ifVar5 === 50) {
  console.log(`ifVar5 ist gleich 50`);
} else {
  console.log(`ifVar5 ist ungleich 50`);
}
```

// prüft, ob ifVar5 gleich 50 ist. Da diese Bedingung unwahr ist, wird das else ausgeführt.  
Auf der Console wird ausgegeben: ifVar5 ist ungleich 50

## Else-If-Bedingung

Anstatt des einfachen Codes nach dem Else kann hier auch eine erneute If-Bedingung stehen.

Syntax:

```
if (Bedingung) {Code}
else if ((Bedingung) ) {Code}
else {Code}                // else ist hier optional
```

```
var ifVar6 = 15;
```

```
if (ifVar6 === 50) {
  console.log(`ifVar6 ist gleich 50`);
} else if (ifVar6 === 100) {
  console.log(`ifVar6 ist gleich 100`);
} else {
  console.log(`ifVar6 ist weder 50 noch 100`);
}
```

// prüft, ob ifVar6 gleich 50 ist. Da diese Bedingung unwahr ist, wird das else if ausgeführt und geprüft ob ifVar6 gleich 100 ist. Da diese Bedingung wieder unwahr ist, wird das else ausgeführt.

Auf der Console wird ausgegeben: ifVar6 ist weder 50 noch 100

## Switch-Bedingung

In der Switch-Bedingung wird geprüft, ob genau der Fall eintritt.

Falls kein Fall eintritt, kann ein Default festgelegt werden, welches dann ausgeführt wird.

```
switch (Prüfwert) {  
    case "Wert3":  
        Code;  
        break;  
    case "Wert13":  
        Code;  
        break;  
    default:  
        Code;  
}
```

Wichtig nach jedem Fall (case) ist das break, mit dem break mit der Fal (case) beendet.

```
var switchVar = `GENAU`;  
  
switch (switchVar) {  
    case `ungenau`:  
        console.log(`ungenau`);  
        break;  
    case `fastgenau`:  
        console.log(`fastgenau`);  
        break;  
    case `genau`:  
        console.log(`genau`);  
        break;  
    default:  
        console.log(`Kein Fall tritt ein`);  
}
```

// prüft nacheinander welcher Fall eintritt. Da die Überprüfung case-sensitive erfolgt (also auch auf Groß- und Kleinschreibung überprüft wird), tritt keiner der definierten Fälle ein und der Default-Code wird ausgeführt.

Auf der Console wird ausgegeben: Kein Fall tritt ein

**Questions? You're welcome!**

holger.zerbe@web.de