



Programmieren
Starten

Zusammenfassung

Verzweigungen

JavaScript - Dynamische Webseiten entwickeln



EINLEITUNG IF ELSE STATEMENTS

Ein if-Statement ist eine **bedingte Anweisung**. Das bedeutet, dass ein bestimmter Codeabschnitt nur unter einer bestimmten **Bedingung** ausgeführt wird. Im folgenden Beispiel wird anhand eines if-Statements überprüft, ob eine Person 18 Jahre oder älter ist. Ist das der Fall, wird ihr der Eintritt gewährt. Andernfalls soll vorerst nichts passieren:

```
var age = 20;

if (age >= 18) {
    console.log("Eintritt gewährt");
}
```

WICHTIG

Der Codeblock innerhalb der geschweiften Klammern wird nur ausgeführt, wenn die Bedingung erfüllt wird. Sprich: Wenn „age“ 18 oder größer ist.

else

Das if-Statement kann um einen **else-Block** erweitert werden. Dieser wird dann ausgeführt, wenn die Bedingung des if-Statements **nicht** erfüllt wird.

Beispiel:

```
if (age >= 18) {
    console.log("Eintritt gewährt"); //Wird ausgeführt, wenn Bedingung
}                                     wahr wird
else {
    console.log("Nicht alt genug"); //Wird in jedem anderen Fall ausgeführt
}
```

prompt

Dabei handelt es sich um ein Fenster, das sich im Browser öffnet, in das man einen Wert eintragen kann. Im folgenden Beispiel wird der Wert, der in dieses Fenster eingegeben wird, in der Variable `age` gespeichert. Anschließend wird der eingegebene Wert im nachfolgenden `if`-Statement ausgewertet.

Beispiel:

```
var age = prompt("Trage dein Alter ein:");

if (age < 18) {
    console.log("Nicht alt genug");
}
else {
    console.log("Eintritt zugelassen");
}
```

BEDINGUNGEN UMDREHEN (!-OPERATOR)

Der !-Operator

Durch den `!`-Operator kann eine Bedingung umgedreht werden. Das bedeutet beispielsweise in einem `if`-Statement, dass nicht überprüft wird, ob die zwischen Klammern stehende Bedingung wahr (`true`) ist, sondern ob sie **falsch** (**`false`**) ist.

Beispiel:

```
if (!(age > 18)) {
    console.log("Nicht alt genug");    //Wird ausgeführt, wenn age
}                                     NICHT größer ist als 18
else {
    console.log("Eintritt zugelassen"); //Wird in jedem anderen Fall
}                                     ausgeführt
```

ELSE IF STATEMENTS

Mit **`else if`** können weitere Bedingungen zu einem `if`-Statement hinzugefügt werden, die ausgewertet werden sollen, wenn die ursprüngliche Bedingung nicht `true` ist.

Beispiel:

```

var color = prompt("Trage eine Ampel-Farbe ein:");
var lowerColorName = color.toLowerCase();

if (lowerColorName == "grün") {
    alert("Du darfst fahren");
} else if (lowerColorName == "orange") {
    alert("Du darfst anfahren oder musst bremsen");
} else if (lowerColorName == "rot") {
    alert("Du musst halten");
}

```

toLowerCase()

Diese Methode wandelt alle Buchstaben eines Strings in **Kleinbuchstaben** um.

ODER-OPERATOR UND UND-OPERATOR

| | | |
|--------------------------------------|---|---|
| Logischer ODER -Operator | Verknüpft zwei oder mehrere Bedingungen | Eine der Bedingungen muss erfüllt sein |
| Logischer UND -Operator && | Verknüpft zwei oder mehrere Bedingungen | Beide bzw. alle Bedingungen müssen erfüllt sein |

Beispiel ODER-Operator

Im folgenden Beispiel wird der Code innerhalb des if-Statements dann ausgeführt, wenn das eingegebene Alter entweder kleiner oder gleich 6 ist **ODER** größer oder gleich 60.

```

if (age <= 6 || age >= 60){
    alert("Preis beträgt 4€")
}

```

Beispiel UND-Operator

Im folgenden Beispiel wird der Code innerhalb des if-Statements dann ausgeführt, wenn die Variable speed größer als 70 ist **UND** die Variable isFacing true ergibt.

```

if (speed > 70 && isFacing == true) {
    alert("Du wurdest geblitzt");
}

```

NESTED IF STATEMENTS

Ein if-Statement innerhalb eines anderen if-Statements wird als **nested if-Statement** bezeichnet.

Beispiel:

```
if (vehicle == "car") {  
  
    if (isTruck == true) {  
        //Reparatur LKW  
    } else {  
        //Reparatur PKW  
    }  
} else if (vehicle == "motorcycle") {  
    //Reparatur Motorrad  
}
```

SWITCH-STATEMENT

Bei einem Switch-Statement wird der Wert der **Bedingung** mit dem Wert, der neben jedem **case** steht, verglichen. Das **default-Statement** am Ende wird ausgeführt, wenn keiner der Fälle (cases) mit dem Wert der Bedingung übereingestimmt hat. Das Schlüsselwort „**break**“ beendet ein Switch-Statement.

Beispiel:

```
var product = "Mehl";  
  
switch (product) {  
  
    case "Mehl":  
        console.log("Mehl kostet 1€");  
        break;  
    case "Milch":  
        console.log("Milch kostet 2€");  
        break;  
    case "Apfel":  
    case "Birne":  
        console.log("Früchte kosten 1€");  
        break;  
    default:  
        console.log("Kein Preis gefunden");  
  
}    //Ausgabe: Mehl kostet 1€
```