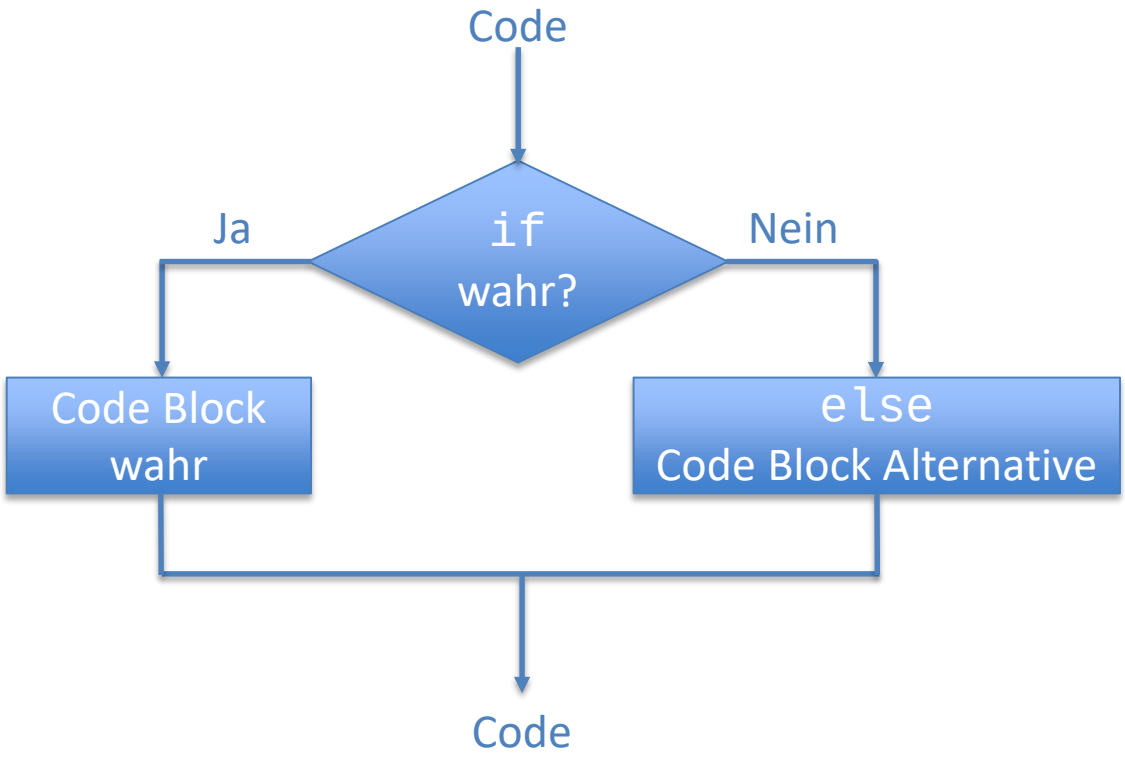
Funktionen nur wenn…dann…

Funktionen können auch nur in bestimmten Fällen angewendet werden.

Code wird nur in einem bestimmten Fall ausgeführt



If… else..

```
if (Bedingung) {
  //Was unter dieser Bedingung getan werden soll
} else {
// Was ansonsten getan werden soll
}
```

Die geschweiften Klammern haben die gleiche Aufgabe wie bei der Funktions-definition. Sie fassen Code zusammen, sodass der Interpreter weiß, dass er zusammengehört.

Die Bedingung ermöglicht die Entscheidung, welcher Block ausgeführt wird.

Der `if`-Block wird ausgeführt, wenn die Bedingung erfüllt ist.

Ist die Bedingung nicht erfüllt, wird der `else`-Block ausgeführt.

If… else.. – die Bedingung

```
if (Bedingung) {
 //Was unter dieser Bedingung getan werden soll
} else {
// Was ansonsten getan werden soll
}
```

Eine gültige Bedingung ist immer dann gegeben, wenn der Computer die Entscheidung: **wahr oder falsch** treffen kann.

```
2 == "2"     wahr
2 === "2"    falsch
```

Zeichen	Bedeutung
>	größer als
>=	größer gleich
==	gleich
<=	kleiner gleich
<	kleiner
!=	ungleich
===	gleich ohne Typkonvertierung
!==	ungleich ohne Typkonvertierung

If… else.. – ein Beispiel

```
if (raumtemperatur > 20) {
    heizungAusschalten();
} else {
    heizungEinschalten();
}
```

Die Bedingung ist dann **wahr**, wenn die Raumtemperatur **größer** ist als 20.

Wenn die Bedingung wahr ist, wird (hier) eine zuvor spezifizierte Funktion ausgeführt.

Ein anderer Code wird ausgeführt, wenn die Bedingung **falsch** ist.
Der *else*-Code kann auch weggelassen werden, wenn nichts gemacht werden soll.

If… else.. – Geschachtelt in einem else-Code

```
if (raumtemperatur > 20) {
    heizungAusschalten();
} else if (Raumtemperatur < 17) {
    heizungEinschaltenLevel5();
} else if (Raumtemperatur < 18){
    heizungEinschaltenLevel4();
} else
    heizungEinschaltenLevel3();
}
```

else if ist die einfachste Möglichkeit, mehr als zwei Fälle zu unterscheiden. Hier kann beliebig häufig geschachtelt werden.

Die Reihenfolge ist relevant, es wird von oben nach unten gelesen. Das zuerst zutreffende wird ausgeführt.

If… else.. – Bedingungen kombinieren

```
if (raumtemperatur > 20 && jahreszeit == "Winter") {
    heizungAusschalten();
} else if (Raumtemperatur < 17) {
    heizungEinschaltenLevel5();
} else if (Raumtemperatur < 18){
    heizungEinschaltenLevel4();
} else
    heizungEinschaltenLevel3();
}
```

Das doppelte ampersand && verknüpft zwei Bedingungen. Die gesamte Bedingung ist nur dann wahr, wenn beide Bedingungen einzeln wahr sind.

If… else.. & Funktionen – Übungsaufgabe 12

Ziel: Bau eines Eingabeformulars, welches die Korrektheit der Inhalte prüft.

1. Baue ein Formular mit 3 Eingabefeldern (Vorname, Nachname, E-Mail)
2. Erstelle einen Button, der die Formulareingaben überprüft.
3. Gebe eine Fehlermeldung aus, wenn JavaScript im Browser nicht aktiv ist.
4. Schreibe je eine leere Funktion zur Überprüfung eines Formularfeldes (function pruefeVorname(){})
5. Schreibe eine Funktion, die die zuvor erstellen Funktionen ausführt und mit dem Button verknüpft.
6. Schreibe nun, in die in 4. erstellte Funktion die Prüfungen ein:
   a. Prüfe ob ein Vorname eingetragen wurde
   b. Prüfe ob ein Nachname eingeben wurde, der länger ist als ein Buchstabe.
   c. Prüfe ob eine E-Mail-Adresse eingegeben wurde (enthält die Eingabe ein `@?.indexOf("@")<0`)

If… else.. & Funktionen – Übungsaufgabe 12

7. Erstelle in 6. zunächst die relevanten Variablen. Dann erstelle eine If…else…Abfrage, um zu prüfen, dass das Feld gefüllt ist.
   Bei leerem Feld (vorname== "") soll eine Fehlermeldung an die Funktion schreibeFehler() übergeben werden.
8. schreibeFehler-Funktion lautet wie folgt und steht zuoberst im Skript:

```
function schreibeFehler(meldung) {
    var fehlerContainer = document.getElementById("fehler");
    fehlerContainer.innerHTML += "<p>" + meldung + "</p>";
}
```

9. Gebe der Fehlermeldung mit Hilfe von `<div id=„fehler“></div>` einen Ort im <body>.

If… else.. & Funktionen – Übungsaufgabe 12

10. Und jetzt das letzte noch was zum knobeln:

Damit die Fehlermeldungen bei erneutem Buttonklick nicht immer wieder untereinander angezeigt werden, muss eine weitere Funktion geschrieben werden, welche die bestehenden Ausgaben überschreibt. Beachte dabei die Position der Funktion.

If… else.. – Oder-Bedingungen

```
if (raumtemperatur > 20 || jahreszeit == „Sommer") {
    heizungAusschalten();
} else if (Raumtemperatur < 17) {
    heizungEinschaltenLevel5();
} else if (Raumtemperatur < 18){
    heizungEinschaltenLevel4();
} else
    heizungEinschaltenLevel3();
}
```

Der doppelte senkrechte Strich || verknüpft zwei Bedingungen. Die gesamte Bedingung ist dann wahr, wenn eine der beiden Bedingungen wahr ist.

If… else.. – Negation-Bedingung

```
if (!raumtemperatur) {
    Fehlermeldung(Es kann keine Temperatur gemessen
    werden.);
}
```

Das Ausrufezeichen vor einer Bedingung dreht die Prüfung von Richtig auf Falsch. Das Beispiel prüft den Fall, dass es keine Raumtemperatur gibt.

`True` und `false` sind Boolean-Werte in JavaScript. Boolean-Werte sind Datentypen, wie String und Number.
! Liefert auch einen Boolean-Wert.

```
var = raumtemperatur = false;
if (!raumtemperatur) {
    Meldung(Ich kann eine Temperatur lesen.);
}
```

If… else.. – Was ist die Wahrheit?

Werte, die von JavaScript als `false` behandelt werden, heißen falsy (falsch-artig).

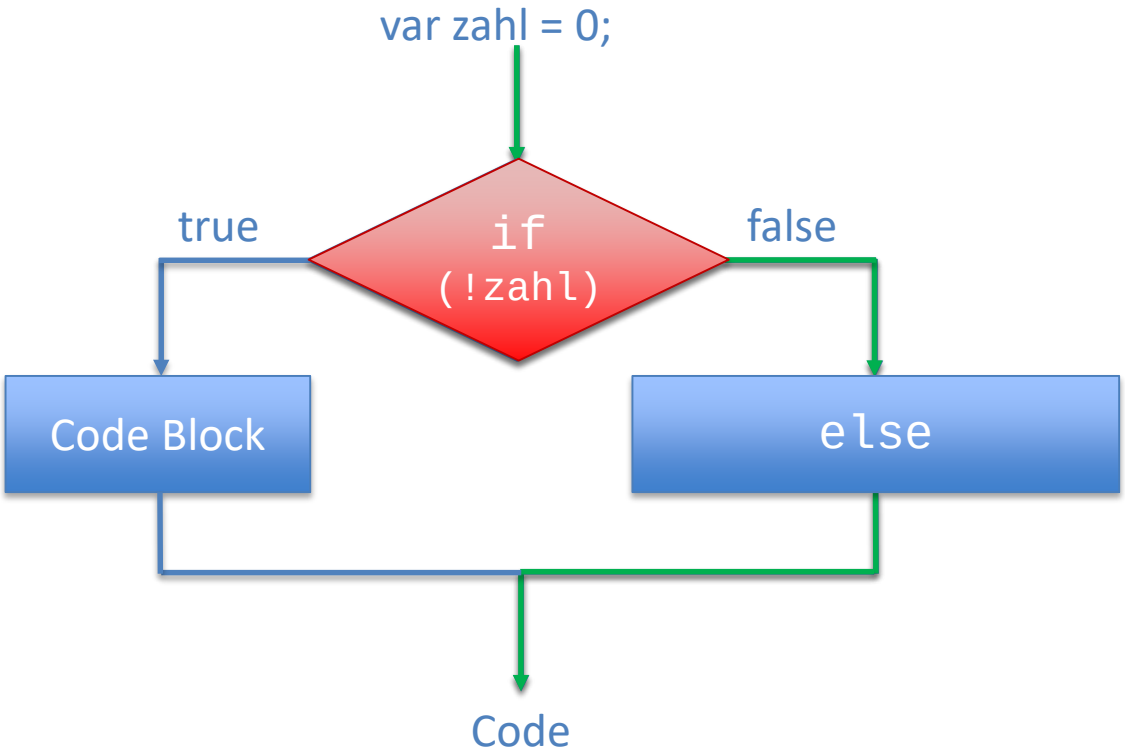
Werte, die von JavaScript als `true` behandelt werden, heißen truthy *(richtig-artig)*.

falsy-Werte:

- false
- 0 *(Achtung "0" ist truly)*
- ""
- null
- NaN (not a Number)
- undefined

If… else.. – Was ist die Wahrheit? Ein Beispiel

Welcher Code wird ausgeführt?



If… else.. – Was ist die Wahrheit? Übung 13

Welcher Code wird ausgeführt?

```
var variable1 =3;
var variable2 =7;
var variable3 =0;
var variable4 ="0";
var variable5 ="";
var variable6 =true;
if (variable1 > variable2) {…} else {…}        //1
if (variable1 != variable2) {…} else {…}       //2
if (variable1) {…} else {…}                    //3
if (variable1 === "3") {…} else {…}            //4
if (!variable4) {…} else {…}                   //5
if (variable3 || variable 2) {…} else {…}      //6
if (variable1 && variable 4) {…} else {…}      //7
if (!variable6) {…} else {…}                   //8
if (variable4.indexof("3")) {…} else {…}       //9
if (variable3 == variable4) ==
(variable3 === variable4) {…} else {…}         //10
```

Lessons Learned – Java Script

Primitive Datentypen:

- String – "Hallo", "23"
- Number – 1, 2, 3, …
- Boolean – true, false

Objekt-Typen

- Function – Aneinanderreihungen von Anweisungen, die man in Variablen speichern kann, man kann sie als Parameter übergeben und ihnen Eigenschaften zuweisen.