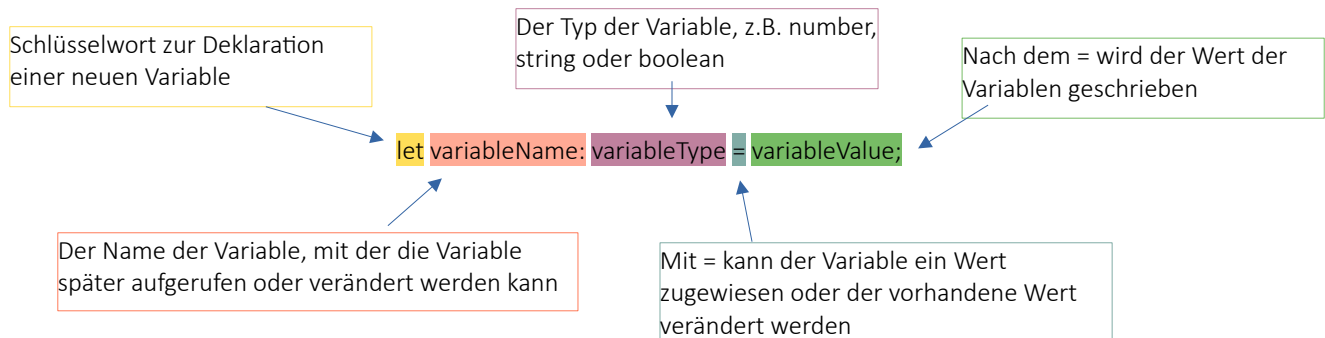


# TypeScript Cheat Sheet

## Variablen



Als Code kann das zum Beispiel so aussehen:

```
let name: string = "Hannah";
```

→ Hier wird eine Variable mit dem Namen „name“ angelegt. Da ein Name eine Zeichenkette, also ist der Variablentyp string. Als Wert wird der Variable die Zeichenkette „Hannah“ zugewiesen.

Mehr Beispiele:

```
let age: number = 23;  
let isFemale: boolean = true;  
let hobbies: string[] = ["Reading", "Swimming", "Basketball"];
```

## Variablentypen

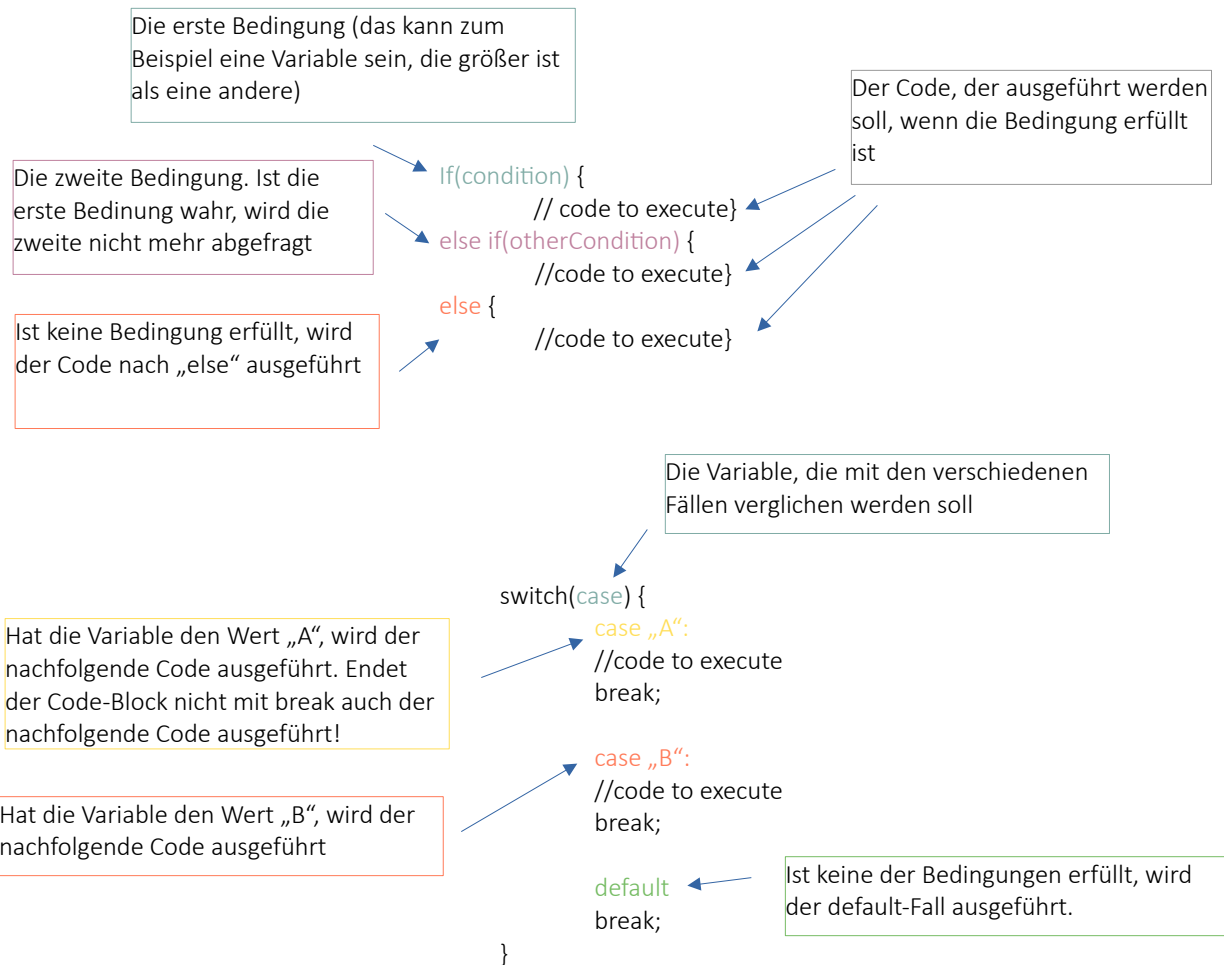
→ Variablen können verschiedene Typen haben. Die Zuweisung hilft, dass nicht aus versehen falsche Werte zugewiesen oder verarbeitet werden.

→ Einfachen Variablentypen sind zum Beispiel: **number** (Zahlen), **string** (Zeichenketten) oder **boolean** (true / false)

→ Alle Typen können auch in **Arrays** gespeichert werden. Also zum Beispiel `string[]` für ein Array, in dem mehrere strings durch Kommas getrennt werden

## Bedingungen

→ Manchmal soll Code nur ausgeführt werden, wenn eine bestimmte Bedingung erfüllt ist. Dazu können if-Bedingungen oder switch cases verwendet werden.



Als Code kann das zum Beispiel so aussehen:

```
let age: number = 23;
if (age == 25) {
  //do something
}
else if (age == 23) {
  //do something else
}
else {
  //do something else
}
```

```
switch (age) {
  case 25:
    //do something
    break;
  case 23:
    //do something
    break;
  default:
    // do something
    break;
}
```

## Vergleichsoperatoren

→ Diese Operatoren können verwendet werden, um zwei Variablen zu vergleichen.

Gleichheit der Werte	==
Ungleichheit der Werte	!=
Größer als	>
Kleiner als	<
Größer gleich	>=
Kleiner gleich	<=

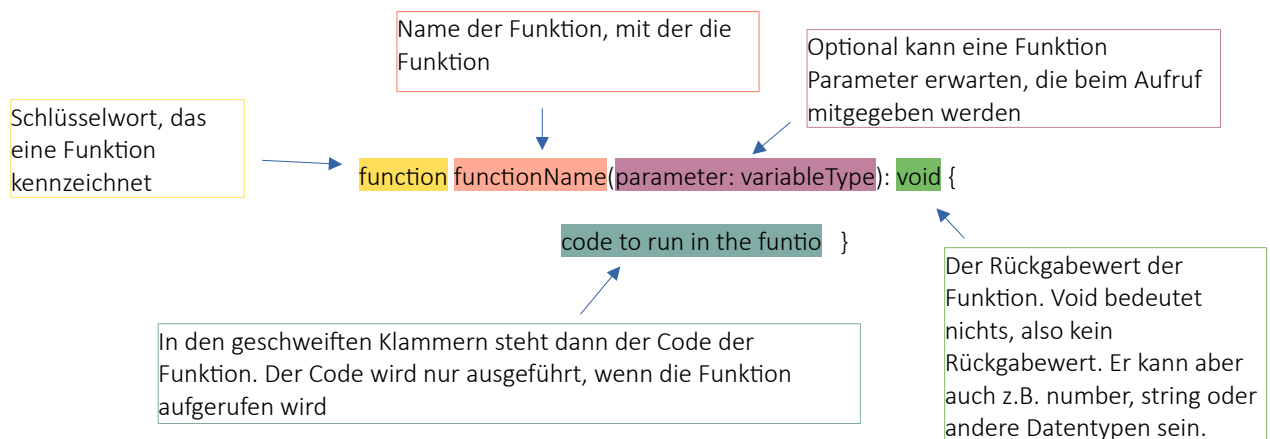
## Logische Operatoren

→ Diese Operatoren werden häufig in if-Bedingungen verwendet. So kann man zum Beispiel festlegen, dass zwei Bedingungen erfüllt werden müssen

Logisches „und“	&&	Bedingung A und Bedingung B müssen erfüllt sein
Logisches „oder“		Bedingung A oder Bedingung B müssen erfüllt sein
Logisches „nicht“	!	Bedingung ist nicht erfüllt, zum Beispiel, dass zwei Variablen nicht gleich sind

# Funktionen

→ Funktionen sind Code-Blöcke die durch den Namen der Funktion aufgerufen werden können



Als Code kann das zum Beispiel so aussehen:

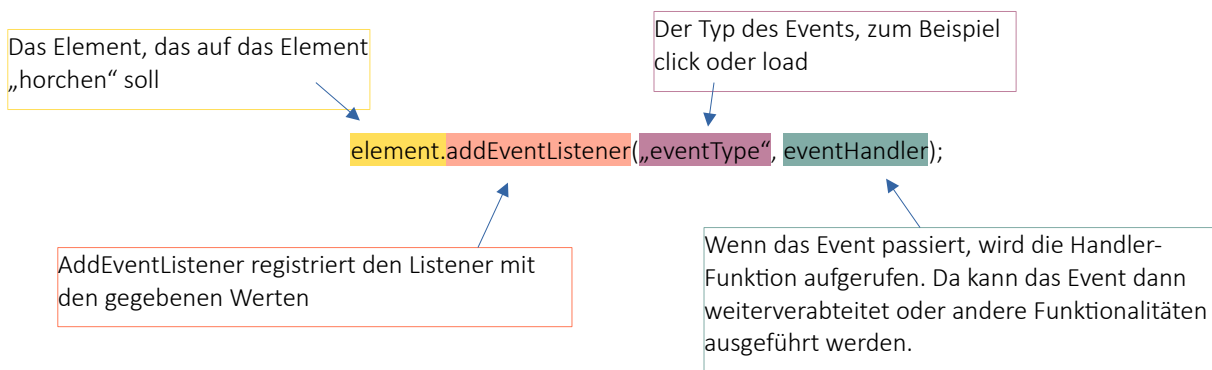
```
function addNumbers(_numberA: number, _numberB: number): number {  
  let newNumber: number = _numberA + _numberB;  
  return newNumber;  
}
```

→ Eine Funktion mit dem Namen „addNumbers“ erwartet zwei Parameter vom Typ `number` und gibt eine `number` als return-Wert zurück. In der Funktion wird eine Variable mit dem Namen `newNumber` erstellt. Ihr Wert ist das Ergebnis der Addition der beiden Parameter. Anschließend wird der Wert zurückgegeben und kann an der Stelle verwendet werden, an der die Funktion aufgerufen wurde.

## Events

→ Events sind Ereignisse, die von verschiedenen Akteuren ausgelöst werden können. Zum Beispiel „load“ nachdem der Browser die Seite fertig geladen hat, oder „click“ wenn das Nutzer die linke Maustaste drückt.

→ Events können mit EventListnern abgefangen werden.



Im Code kann das zum Beispiel so aussehen:

```
window.addEventListener("load", handleLoad);
```

→ Dem window wird ein load-Listener hinzugefügt. Wenn dann alles geladen wurde, wird das Event getriggert und die Funktion `handleLoad` aufgerufen.

## Eventtypen

→ Es gibt verschiedene Eventtypen, die von verschiedenen Akteuren ausgelöst werden können. Diese sind zum Beispiel:

Page	Mouse	Keyboard	Pointer
load	Click	Keydown	Pointerdown
	Mousedown	Keyup	Pointerup
	Mouseover (→ hover)		