

LU02a - Variablen

Einleitung

Variablen sind Platzhalter für Daten (Kontainer), die im Programm gespeichert und später wiederverwendet werden können. Sie ermöglichen es, Werte wie Zahlen, Texte oder Objekte mit einem Namen zu versehen, anstatt sie direkt im Code mehrfach zu verwenden. Dadurch wird der Code flexibler, besser lesbar und leichter wartbar.

Variablen dienen in JavaScript dazu, Werte zu speichern und wiederzuverwenden. Sie können Zahlen, Texte, Objekte, Funktionen und vieles mehr aufnehmen.

Keywords

Zur Deklaration von Variablen stehen drei Keywords zur Verfügung:

- var
- let
- const

var

Das älteste Schlüsselwort in JavaScript. Es ist funktion-skopiert (gültig innerhalb der Funktion) und nicht block-skopiert (gültig im gesamten Block) und wird beim Programmstart an den Anfang des Scopes verschoben.

Hinweis: Heutzutage wird var selten empfohlen, da es leicht zu unerwartetem Verhalten führt, ist also veraltet.

Beispiel

```
function demoVar() {  
    if (true) {  
        var name = "Volkan";  
    }  
    console.log(name); // funktioniert, obwohl innerhalb des if-BLOCKS  
    deklariert (scope innerhalb der Funktion  
}  
demoVar();
```

The screenshot shows a terminal window with the following content:

```
scripts > JS test.js > ...
1 function demoVar() {
2   if (true) {
3     var name = "Volkan";
4   }
5   console.log(name); // funktioniert, obwohl innerhalb des if-Blocks deklariert (scope innerhalb der Funktion
6 }
7 demoVar();
```

TERMINAL

```
volkandemir@Mac scripts % node test.js
Volkan
volkandemir@Mac scripts %
```

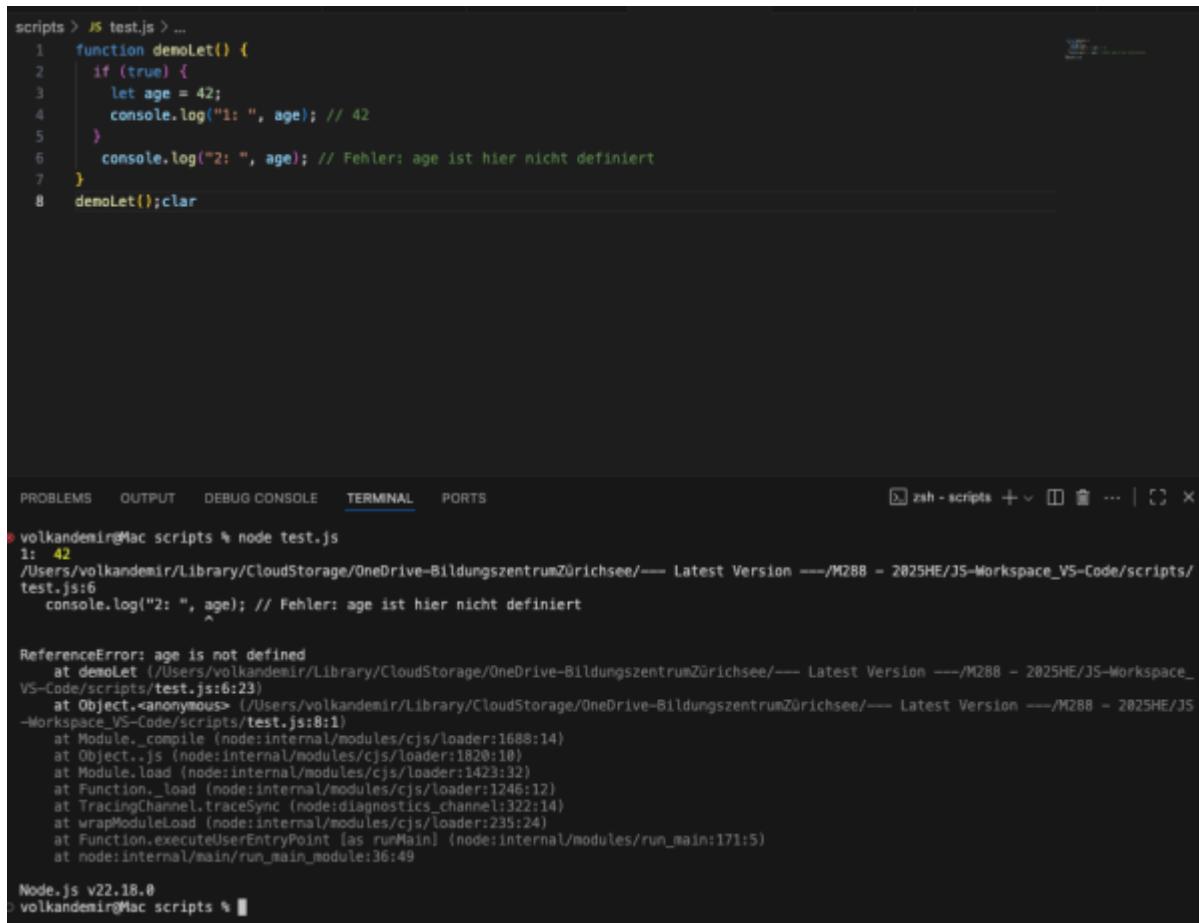
Wie in der Abbildung zu sehen ist, ist die Ausgabe der Variable *name* fehlerfrei, obwohl diese ausserhalb des if-Blocks definiert wurde. Der Scope (Gültigkeitsbereich) ist bei var die Funktion.

let

Seit ES6 (2015) eingeführt. let ist block-skopiert, d. h. die Variable existiert nur innerhalb des Blockes { ... }, in dem sie definiert ist.

Beispiel

```
function demoLet() {
  if (true) {
    let age = 42;
    console.log(age); // 42
  }
  // console.log(age); // Fehler: age ist hier nicht definiert
}
demoLet();
```



```

scripts > js test.js > ...
1  function demoLet() {
2    if (true) {
3      let age = 42;
4      console.log("1: ", age); // 42
5    }
6    console.log("2: ", age); // Fehler: age ist hier nicht definiert
7  }
8  demoLet();clar

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

volkandemir@Mac scripts % node test.js
1: 42
/Users/volkandemir/Library/CloudStorage/OneDrive-BildungszentrumZürichsee/--- Latest Version ---/M288 - 2025HE/JS-Workspace_VS-Code/scripts/test.js:6
  console.log("2: ", age); // Fehler: age ist hier nicht definiert
^

ReferenceError: age is not defined
  at demoLet (/Users/volkandemir/Library/CloudStorage/OneDrive-BildungszentrumZürichsee/--- Latest Version ---/M288 - 2025HE/JS-Workspace_VS-Code/scripts/test.js:6:23)
  at Object.<anonymous> (/Users/volkandemir/Library/CloudStorage/OneDrive-BildungszentrumZürichsee/--- Latest Version ---/M288 - 2025HE/JS-Workspace_VS-Code/scripts/test.js:8:1)
  at Module._compile (node:internal/modules/cjs/loader:1688:14)
  at Object.js (node:internal/modules/cjs/loader:1820:10)
  at Module.load (node:internal/modules/cjs/loader:1423:32)
  at Function._load (node:internal/modules/cjs/loader:1246:12)
  at TracingChannel.traceSync (node:diagnostics_channel:322:14)
  at wrapModuleLoad (node:internal/modules/cjs/loader:235:14)
  at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
  at node:internal/main/run_main_module:36:49

Node.js v22.18.0
volkandemir@Mac scripts %

```

Wie in der Ausgabe zu sehen, wird *Markpoint 1:* fehlerfrei ausgegeben, *Markpoint 2:* liefert dann eine Fehlermeldung, weil `let` außerhalb des IF-Blocks deklariert wurde.

Zusatzmaterial

- [W3School- JavaScript Variables](#)
- [SelfHTML - JS Variablen](#)



Volkan Demir

From:

<https://wiki.bzz.ch/> - BZZ - Modulwiki

Permanent link:

<https://wiki.bzz.ch/de/modul/m288/learningunits/lu02/01?rev=1756724681>

Last update: **2025/09/01 13:04**

