

JavaScript

JavaScript ist eine Programmiersprache, die hauptsächlich für die Entwicklung von interaktiven Webseiten verwendet wird. Hier sind einige wichtige Konzepte in JavaScript:

- **Variablen:** In JavaScript können wir Variablen verwenden, um Werte zu speichern. Beispiel:

```
let name = 'John';
```

- **Datentypen:** JavaScript unterstützt verschiedene Datentypen wie String, Number, Booleans, Arrays, Objekte und mehr. Beispiel:

```
let age = 30;
let isStudent = true;
let hobbies = ['Singen', 'Tanzen', 'Lesen'];
console.log(hobbies[0]); // Singen
let person = {
  name: 'John',
  age: 30,
  isStudent: true
};

console.log(person.name, person.age); // John 30
```

- **Konstanten vs. Variablen:** Konstanten können sich nach der initialen Deklaration nicht mehr ändern und werden mit `const` definiert. Variablen können sich ändern und werden mit `var` oder `let` definiert

```
const volljährigAb = 18;
volljährigAb = 21; // ERROR volljährigAb ist Konstante und kann nicht neu assigned werden!

// Mit Variablen kein Problem:
var meinAlter = 27;
meinAlter = 28;

let meineHobbies = ["Javascript"];
meineHobbies = ["Linux", "Javascript"];
```

- **var vs let:** `var` ist der ältere Weg und `let` und `const` sind der neue Weg (seit ES6 2015), `let` ist nur im aktuellen Scope verfügbar und `var` in der gesamten Funktion
- Ich würde empfehlen `let` und `const` zu verwenden

```
const test = true

if (test) {
  let myLetVar = "let it be";
  var myVarVar = "Var es nicht schön"
}

console.log(myLetVar); // ERROR - Nicht definiert!
console.log(myVarVar); // Kein Problem
```

- **Funktionen:** Mit Funktionen können wir Codeblöcke definieren, die wiederverwendbar sind. Beispiel:

```
function greet(name) {
  console.log('Hallo ' + name + '!');
}
```

```
greet('John');
```

- **Bedingungen:** Mit Bedingungen können wir Code basierend auf bestimmten Kriterien ausführen. Beispiel:

```
let age = 18;

if (age >= 18) {
  console.log('Du bist volljährig.');
```

```
} else {
  console.log('Du bist minderjährig.');
```

```
}
```

- **Schleifen:** Schleifen ermöglichen es uns, Code wiederholt auszuführen. Beispiel

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}
```

```
let alter = 0;
while (alter < 5) {
  console.log(alter);
  alter++;
}
```

- **Operatoren:** Werte können auf verschiedene Arten manipuliert werden

```
// String

const str1 = "Hallo ";
const str2 = "Welt";
console.log(str1 + str2);
// Hallo Welt

// Nummern
let num1 = 10;
let num2 = 2;

// Addition
num1 = num1 + num2
// Ist das gleiche wie:
num1 += num2

// Subtraktion
num1 = num1 - num2
// Ist das gleiche wie:
num1 -= num2

// Multiplikation
num1 = num1 * num2
// Ist das gleiche wie:
num1 *= num2

// Division
num1 = num1 / num2
// Ist das gleiche wie:
num1 /= num2

// Spezialfälle

num1 = num1 + 1
// Ist das gleiche wie:
num1++

num1 = num1 - 1
// Ist das gleiche wie:
num1--
```

- **Array Funktionen:** Es gibt verschiedene Möglichkeiten den Inhalt eines Arrays (Liste) zu manipulieren
- Die Länge eines Arrays kann mit der `length` Property abgefragt werden

```
const hobbies = ["Javascript", "Linux"]
const hobbyNum = hobbies.length; // 2 -> 2 Elemente im Hobbies Array

console.log("Ich habe " + hobbies.length + " Hobbies")

// Wird auch oft verwendet um über alle Elemente des Arrays zu loopen

for (let i = 0; i < hobbies.length; i++) {
  console.log(hobbies[i]);
}
// Javascript
// Linux
```

- Um ein neues Element am **Ende** des Arrays einzufügen, benutze `array.push`

```
let myArr = [0, 1]
myArr.push(2)
console.log(myArr);
// [ 0, 1, 2]
```