JavaScript

Inhalt

Kommentare

Nebst dem "normalen" Code, können auch Kommentare geschrieben werden. Es gibt ein- und mehrzeilige Kommentare. Code, welcher auskommentiert ist, wird nicht als Code behandelt und wird deshalb auch nicht durchlaufen.

Einzeiliger Kommentar

```
// Ich bin ein einzeiliger Kommentar
```

Mehrzeiliger Kommentar

```
/*
    Ich bin
    ein mehrzeiliger Kommentar
*/
```

HTML Inhalte mit JavaScript ändern

Mit der Methode getElementById() kann ein HTML-Element via seiner Id selektiert werden und dessen Inhalt verändert werden.

```
document.getElementById("demo").innerText = "Hello JavaScript";
```

Output Optionen

1. Ins HTML schreiben mit innerText.
2. Ins HTML schreiben mit document.write().
3. In eine Alertbox schrieben mit window.alert().
4. In die Konsole des Browsers schreiben mit console.log().

Variablen

Was sind Variablen?

Eine Variable ist ein Platzhalter / eine Leerstelle in einem logischen oder mathematischen Ausdruck. Der Variable kann ein beliebiger Wert zugewiesen werden.

Deklaration

Die Variable x erstellen:

```
let x;
```

Initialisierung

Der eben erstellten Variable x den Wert 5 zuweisen:

```
x = 5;
```

Deklaration & Initialisierung

Es ist auch möglich eine Variable zu erstellen und ihr direkt einen Wert zuzuweisen (meist wird das auch ganeu so gemacht).

```
let x = 5;
```

Datentypen

In JavaScript gibt es unter anderem die Datentypen string, number und boolean. Wenn einer Variable nun ein bestimmter Wert zugewiesen wird, nimmt diese automatisch den Datentypen des Wertes an.

Number

Der Datentyp number kann sowohl eine ganze Zahl (10), als auch eine Fliesskommazahl (1.5) sein. Wenn also der Variable x der Wert 5 vom Type number zugewiesen wird, ist nun die Variable x ebenfalls vom Typ number.

```
let x = 5; // Der Datentyp der Variable x ist number
```

String

Der Datentyp string ist eine Aneinanderreihung von Zeichen ("Die Stadt ist 100 Jahre alt.") und kann sowohl Buchstaben, als auch Zahlen enthalten.

```
let z = "Die Erde"; // Der Datentyp der Variable z ist string
```

Boolean

Der Datentyp boolean ist binär, das bedeutet er kann entweder "true" oder "false" sein.

```
let y = true; // Der Datentyp der Variable y ist boolean
```

Arrays

Ein Array ist eine spezielle Variable, welche mehrere Elemente beinhalten kann. Die Drei Car Variablen (car1, car2 und car3) können entweder wie unten gezeigt einzeln erstellt werden oder sie können zusammen gruppiert in einem Array erstellt werden.

```
let car1 = "Saab";
let car2 = "Volvo";
let car3 = "BMW";
```

```
const cars = ["Saab", "Volvo", "BMW"]; // Array
```

Verwendung eines Arrays

Mit dem Index kann auf ein Element eines Arrays zugegriffen werden. Der Index beginnt bei 0, deshalb hat das erste Element eines Arrays den Index 0.

```
const cars = ["Saab", "Volvo", "BMW"];

// Das erste Element "Saab" des Arrays in die Konsole schreiben
conosle.log(cars[0]);

// Element am Ende des bestehenden Arrays hinzufügen
let car4 = "Ford";
cars.push(car4);

// Das letzte Element vom bestehenden Array löschen
cars.pop();

// Das erste Element im bestehenden Array überschreiben
cars[0] = "Audi";

// Länge eines Arrays herausfinden und in der Variable size speichern
let size = cars.length;
```

Operatoren

Zuweisungsoperator

Mit dem Zuweisungsoperator (=) kann einer Variable ein Wert zugeordnet werden.

```
let z = "Die Erde";
```

Vergleichsoperatoren

Mit den Vergleichsoperatoren (<, <=, > >=, ==, ===) können zwei Variablen verglichen werden. Der Operator == überprüft zwei Variabeln auf den selben Wert. Der Operator === überprüft zwei Variablen auf den selben Wert und den selben Datentypen.

```
let x = 5;
let y = 10;
let isTrue = x == y; // isTrue ist false, da der Wert 5 nicht gleich dem Wert 10
ist
```

Addition

```
let x = 5; // x den Wert 5 zuweisen
let y = 2; // y den Wert 3 zuweisen
let z = x + y; // z das Resultat (7) von x + y zuweisen

// Spezialfall für strings: die beiden Variablen werden konkateniert
(aneinandergehängt)
let a = "Hello ";
let b = "World";
let c = a + b; // in c ist nun "Hello World" gespeichert
```

Subtraktion

```
let x = 5; // x den Wert 5 zuweisen
let y = 2; // y den Wert 3 zuweisen
let z = x - y; // z das Resultat (3) von x - y zuweisen
```

Multiplikation

```
let x = 5; // x den Wert 5 zuweisen
let y = 2; // y den Wert 3 zuweisen
let z = x * y; // z das Resultat (10) von x * y zuweisen
```

Division

```
let x = 6; // x den Wert 6 zuweisen
let y = 2; // y den Wert 3 zuweisen
let z = x / y; // z das Resultat (3) von x / y zuweisen
```

Negation

```
let x = true; // x den Wert true zuweisen
let y = !x; // y den Wert das Gegenteil (false) von x zuweisen
```

Und

Beim "&&" Operator ist das Ergebnis nur true, wenn beide Variablen auch true sind. Sonst ist das Ergebnis false.

```
let x = true; // x den Wert true zuweisen
let y = false; // y den Wert false zuweisen
```

```
let z = x && y; // z das Resultat (false) von x und y zuweisen
```

Oder

Beim "||" Operator ist das Ergebnis true, wenn mindestens eine der beiden Variablen true ist. Sonst ist das Ergebnis false.

```
let x = true; // x den Wert true zuweisen
let y = false; // y den Wert false zuweisen
let z = x || y; // z das Resultat (true) von x und y zuweisen
```

Inkrementieren

Der Wert einer Variable kann inkrementiert (erhöht) werden mit dem Operator +=.

```
let x = 5; // x den Wert 5 zuweisen
let y = 2; // y den Wert 2 zuweisen
let y += x; // y den Wert von y + x zuweisen
```

Dekrementieren

Der Wert einer Variable kann dekrementiert (verkleinert) werden mit dem Operator -=.

```
let x = 5; // x den Wert 5 zuweisen
let y = 2; // y den Wert 2 zuweisen
let y -= x; // y den Wert von y - x zuweisen
```

Verzweigungen

Mit der If-Else-Verzweigung kann unterschieden werden, was je nach Bedingung passieren soll.

```
let x = 5; // x den Wert 5 zuweisen

if (x == 5) { // Unter der Bedingung das x gleich 5 ist, wird x um 5 erhöht
  x += 5;
} else if (x == 0){ // Unter der Bedingung das x gleich 0 ist, wird x um 5
verkleinert
  x -= 5;
} else { // Unter der Bedingung das x gleich 1 ist, wird x um 1 erhöht
  x += 1;
}
```

Schleifen

For-Schleife

Ein for Loop wir so oft durchlaufen, bis die Variable i einen Wert angenommen hat, welcher grösser oder gleich 5 ist. Die Variable i ist zu Beginn 0 und wird nach jedem Durchlauf der Schleife mit dem Ausdruck i++ um eines inkrementiert (erhöht).

```
for (let i = 0; i < 5; i++) {
  console.log(i)
}
// Output in der Konsole:
// The number is 0
// The number is 1
// The number is 2
// The number is 3
// The number is 4
```

For-Each-Schleife

```
const persons = ["John", "Amie", "Leon"];

persons.forEach((person)=> {
  console.log(person); // Der Name der Person, an der x-ten Stelle des Arrays
wird in die Konsole geschrieben
})

// Output in der Konsole:
// John
// Amie
// Leon
```

Funktionen

Code kann in eigene Funktionen ausgelagert werde, welche anschliessend aufgerufen werden können und meist einen Rückgabewert haben.

```
// Funktion, um das Produkt von p1 und p2 zu berechnen
function myFunction(p1, p2) {
  return p1 * p2;
}

let result = myfunction(2, 4); // Funktion aufrufen und deren Resultat der
Variable result zuweisen
```