

# JavaScript Handout

---

Kommentare werden mit zwei Schrägstrichen gekennzeichnet und nicht vom Browser ausgewertet:

```
// Dies ist ein Kommentar
```

## Variablenzuweisung

```
var nameMax = "Max Mustermann";  
  
var alter;  
alter = 25;
```

## Primitive Datentypen

```
var length = 16; // Nummer  
var lastName = "Johnson"; // String  
var isItSaturday = true; // Boolean
```

## Konsolenausgabe

```
// Ausgabe Wert direkt  
console.log("Moin"); // Ausgabe: Moin  
  
// Ausgabe Variable  
var greeting = "Hello, Develop<HER>!";  
console.log(greeting) // Ausgabe: Hello, Develop<HER>!  
  
var year = 2020;  
console.log(year) // Ausgabe: 2020
```

## Arithmetische Operationen

```
// Addition  
console.log(11 + 4); // Ausgabe: 15  
  
var addition = 11 + 4;  
console.log(addition); // Ausgabe: 15  
  
// Subtraktion
```

```

var subtraktion = 11 - 4;
console.log(addition); // Ausgabe: 7

// Multiplikation
var multiplikation = 11 * 4;
console.log(addition); // Ausgabe: 44

// Division
var division = 11 / 4;
console.log(addition); // Ausgabe: 2.75

// Inkrement um eins mit ++
var zahl = 1;
zahl++; // Dies ist das Gleiche wie zahl = zahl + 1;
console.log(zahl); // Ausgabe: 2

// Dekrement um eins mit --
var zahl = 1;
zahl--; // Dies ist das Gleiche wie zahl = zahl - 1;
console.log(zahl); // Ausgabe: 0

```

## Objekt-Datentyp

```

// Objekt-Definition
var beruf = {
  name: "Webentwicklerin", // Objekt-Eigenschaft
  arbeitgeber: "OTTO",
  wochenstunden: 37.5,
  gehalt: 4000
};

// Zugriff auf Objekt-Eigenschaften mit objektName.eigenschaft
console.log(beruf.name); // Ausgabe: Webentwicklerin

```

## Array-Datentyp

```

// Array-Definition
var cars = ["VW", "Toyota", "Tesla", "Ford"];
// Index      0      1      2      3
// Arrays starten beim Index 0!

// Zugriff auf ein Array-Element über den Index mit arrayName[Index]
console.log(cars[0]); // Ausgabe: VW

// Ein neues Element in ein Array hinzufügen
var fruits = ["Banane", "Apfel", "Orange", "Kiwi"];
fruits.push("Birne"); // Füge "Birne" am Ende des Arrays hinzu
// Resultat: fruits = ["Banane", "Apfel", "Orange", "Kiwi", "Birne"]

```

```
// Den Index eines Elements im Array auslesen mit  
arrayName.indexOf(elementImArray);  
console.log(fruits.indexOf("Apfel")); // Ausgabe: 1  
  
// Ist das Element nicht im Array, so wird -1 zurückgegeben  
console.log(fruits.indexOf("Zitrone")); // Ausgabe: -1
```

## Boolean-Datentyp

Zwei Werte: `true` oder `false`

```
var binIchInHamburg = true;  
console.log(binIchInHamburg); // Ausgabe: true  
  
// Negation von Boolean mit !  
console.log(!binIchInHamburg); // Ausgabe: false  
  
// Logisches UND mit &&:  
console.log(false && false); // Ausgabe: false  
console.log(false && true); // Ausgabe: false  
console.log(true && false); // Ausgabe: false  
console.log(true && true); // Ausgabe: true  
  
// Logisches ODER mit ||:  
console.log(false || false); // Ausgabe: false  
console.log(false || true); // Ausgabe: true  
console.log(true || false); // Ausgabe: true  
console.log(true || true); // Ausgabe: true  
  
// Vergleichsoperatoren  
// Kleiner-als-Operator: <  
console.log(1 < 2); // Ausgabe: true  
console.log(2 < 1); // Ausgabe: false  
console.log(1 < 1); // Ausgabe: false  
  
// Größer-als-Operator: >  
console.log(1 > 2); // Ausgabe: false  
console.log(2 > 1); // Ausgabe: true  
console.log(1 > 1); // Ausgabe: false  
  
// Größer-oder-gleich-Operator: <=  
console.log(2 < 1); // Ausgabe: false  
console.log(1 <= 1); // Ausgabe: true  
  
// Kleiner-oder-gleich-Operator: >=  
console.log(2 > 1); // Ausgabe: true  
console.log(1 >= 1); // Ausgabe: true  
  
// Gleichheit: ==  
console.log(1 == 1); // Ausgabe: true  
console.log(1 == 2); // Ausgabe: false
```

```

console.log(1 == "1"); // Ausgabe: true

// Ungleichheit: !=
console.log(1 != 1); // Ausgabe: false
console.log(1 != 2); // Ausgabe: true
console.log(1 != "1"); // Ausgabe: false

// Strikte Gleichheit: ===
console.log(1 === 1); // Ausgabe: true
console.log(1 === 2); // Ausgabe: false
console.log(1 === "1"); // Ausgabe: false

// Strikte Ungleichheit: !==
console.log(1 !== 1); // Ausgabe: false
console.log(1 !== 2); // Ausgabe: true
console.log(1 !== "1"); // Ausgabe: true

```

## Fallunterscheidung mit `if`, `else`

```

if (bedingung) {
    // Block von Code, der ausgeführt werden soll, wenn die Bedingung
    "bedingung" true ist
} else {
    // Block von Code, der ausgeführt werden soll, wenn die Bedingung
    "bedingung" false ist
}

```

`bedingung` ist hierbei ein Boolean

## For-Schleife

Schleifen können einen Codeblock mehrere Male ausführen.

```

for (var i = 0; i < 5; i++) {
    console.log(i);
}
// Ausgabe:
// 0
// 1
// 2
// 3
// 4

```

`var i = 0` setzt eine Variable, bevor die Schleife beginnt.

`i < 5` definiert die Bedingung für die Ausführung der Schleife

`i++` erhöht den Wert von `i` jedes Mal um eins, wenn der Codeblock in der Schleife ausgeführt wurde.

## Function

```
function functionName() {  
    // auszuführender Code  
}  
  
// Aufruf der Function  
functionName();  
  
// Function mit Rückgabewert return  
function addiere(zahl1, zahl2) {  
    return zahl1 + zahl2;  
}  
var summe = addiere(5, 7);  
console.log(summe); // Ausgabee: 12
```

## Document

```
// Gibt eine Referenz zu einem Element anhand seiner ID zurück.  
var element = document.getElementById(elementID);  
  
// Erstellt ein neues HTML-Element  
var newElement = document.createElement(htmlTag);  
  
// Ruft das im Element enthaltene HTML ab oder legt dieses fest.  
var content = element.innerHTML;  
element.innerHTML = "<h1> Neues HTML </h1>";
```

## Hilfreiche Seiten

---

- [w3schools](#)
- [MDN](#)
- [selfhtml](#)