Zahlen, Strings und Booleans

Drei wichtige Datentypen

Variablen speichern Werte. In JavaScript kann ein Wert unterschiedliche Arten haben. Drei besonders wichtige Datentypen sind:

- **Zahlen** für Mengen, Rechnungen oder Messwerte
- **Strings** für Texte
- **Booleans** für Wahrheitswerte: wahr oder falsch

```
let alter = 18;
let name = "Mira";
let istAngemeldet = true;
```

JavaScript erkennt automatisch, welche Art von Wert in einer Variablen gespeichert ist. Du musst den Datentyp normalerweise nicht vorher festlegen.

Zahlen

Zahlen werden in JavaScript ohne Anführungszeichen geschrieben.

```
let punkte = 42;
let temperatur = -3;
let preis = 19.99;
```

Mit Zahlen kannst du rechnen:

```
let a = 10;
let b = 5;
```

```
let summe = a + b;
console.log(summe); // 15
```

Auch negative Zahlen und Dezimalzahlen sind möglich:

```
let kontostand = -25;
let breite = 12.5;

console.log(kontostand);
console.log(breite);
```

> **Wichtig:** JavaScript verwendet für Dezimalzahlen einen Punkt, kein Komma.

```
let richtig = 3.14;
let falsch = 3,14;
```

Die zweite Zeile speichert nicht die Dezimalzahl `3.14`. Verwende deshalb immer einen Punkt.

Zahlen prüfen

Mit `typeof` kannst du herausfinden, welchen Datentyp ein Wert hat:

```
let anzahl = 7;

console.log(typeof anzahl); // "number"
```

Der Datentyp für Zahlen heißt in JavaScript immer `number` — sowohl für ganze Zahlen als auch für Dezimalzahlen.

Strings: Texte speichern

Ein **String** ist eine Zeichenkette, also Text. Strings stehen in Anführungszeichen.

```
let vorname = "Lea";
let stadt = "Hamburg";
let begruessung = "Guten Morgen!";
```

Du kannst für Strings doppelte oder einfache Anführungszeichen verwenden:

```
let text1 = "Hallo";
let text2 = 'Willkommen';

console.log(text1);
console.log(text2);
```

Beide Varianten funktionieren. Wichtig ist nur: Ein String muss mit derselben Art von Anführungszeichen beginnen und enden.

```
let korrekt = "Das ist ein Text.";
let auchKorrekt = 'Das ist ebenfalls ein Text.';
```

Anführungszeichen im Text

Möchtest du Anführungszeichen *innerhalb* eines Textes verwenden, kannst du außen eine andere Variante wählen:

```
let zitat = 'Mira sagt: "JavaScript macht Spaß."';

console.log(zitat);
```

Oder umgekehrt:

```
let frage = "Hast du heute schon 'Hallo' gesagt?";

console.log(frage);
```

Strings verbinden

Mit dem Pluszeichen `+` lassen sich Strings zusammenfügen. Das nennt man **Verkettung**.

```
let vorname = "Noah";
let nachname = "Schmidt";

let vollerName = vorname + " " + nachname;

console.log(vollerName); // Noah Schmidt
```

Das Leerzeichen zwischen den Namen musst du selbst als String `" "` ergänzen.

Auch feste Texte und Variablen können verbunden werden:

```
let name = "Aylin";
let begruessung = "Hallo, " + name + "!";

console.log(begruessung); // Hallo, Aylin!
```

Moderne Schreibweise mit Template Strings

Template Strings sind besonders praktisch, wenn du Variablen in einen Text einsetzen möchtest. Sie verwenden umgekehrte Anführungszeichen, sogenannte Backticks:

```
let name = "Aylin";
let alter = 16;

let text = `Hallo, ich heiße ${name} und bin ${alter} Jahre alt.`;

console.log(text);
```

Innerhalb von `${...}` kannst du Variablen und sogar einfache Berechnungen einsetzen:

```
let punkte = 8;

console.log(`Du hast ${punkte + 2} Punkte erreicht.`);
```

Für Anfänger reicht zunächst das Verbinden mit `+`. Template Strings wirst du aber in modernem JavaScript sehr häufig sehen.

String oder Zahl?

Die Anführungszeichen machen einen entscheidenden Unterschied:

```
let zahl = 12;
let text = "12";

console.log(typeof zahl); // "number"
```

```
console.log(typeof text); // "string"
```

`zahl` ist eine echte Zahl. Mit ihr kannst du rechnen. `text` besteht nur aus den Zeichen `1` und `2`.

```
let ergebnis1 = 12 + 3;
let ergebnis2 = "12" + "3";

console.log(ergebnis1); // 15
console.log(ergebnis2); // 123
```

JavaScript verbindet zwei Strings, statt sie zu addieren.

Booleans: wahr oder falsch

Ein **Boolean** kann nur zwei Werte haben:

- `true` bedeutet *wahr*
- `false` bedeutet *falsch*

Booleans werden **ohne** Anführungszeichen geschrieben.

```
let istOnline = true;
let istFertig = false;
```

Diese Werte sind besonders wichtig, wenn ein Programm Entscheidungen treffen soll.

```
let hatTicket = true;

console.log(hatTicket); // true
```

Später kannst du mit `if` prüfen, ob ein Boolean wahr oder falsch ist:

```
let istAngemeldet = true;

if (istAngemeldet) {
  console.log("Willkommen!");
}
```

Die genaue Schreibweise von `if` lernst du im nächsten Kapitel. Entscheidend ist hier: Die Variable `istAngemeldet` speichert eine einfache Ja-oder-Nein-Information.

`true` und `"true"` sind nicht dasselbe

Wie bei Zahlen gilt auch hier: Anführungszeichen verändern den Datentyp.

```
let richtig = true;
let falsch = "true";

console.log(typeof richtig); // "boolean"
console.log(typeof falsch); // "string"
```

`true` ist ein Boolean. `"true"` ist ein String, also Text.

> **Merke:** Die JavaScript-Schreibweise ist immer klein: `true` und `false`. `True` oder `False` funktionieren nicht.

Datentypen mit `typeof` untersuchen

Der Operator `typeof` zeigt dir den Datentyp eines Werts oder einer Variablen an.

```
let punkte = 100;
let nachricht = "Geschafft!";
let spielLaeuft = true;

console.log(typeof punkte);       // "number"
console.log(typeof nachricht);    // "string"
console.log(typeof spielLaeuft);  // "boolean"
```

Du kannst `typeof` auch direkt mit einem Wert verwenden:

```
console.log(typeof 25);        // "number"
console.log(typeof "Hallo");   // "string"
console.log(typeof false);     // "boolean"
```

Das ist hilfreich, wenn du unsicher bist, warum eine Rechnung oder ein Vergleich nicht wie erwartet funktioniert.

Werte sinnvoll benennen

Der Name einer Variablen sollte erkennen lassen, welche Information sie enthält:

```
let produktName = "Trinkflasche";
let lagerBestand = 24;
let istAusverkauft = false;
```

Diese Namen sind verständlicher als:

```
let a = "Trinkflasche";
let b = 24;
let c = false;
```

Bei Booleans helfen Namen, die wie eine Frage gelesen werden können:

```
let istSichtbar = true;
let hatGewonnen = false;
let kannSpeichern = true;
```

So ist sofort klar, dass die Variable einen Wahrheitswert enthält.

Typischer Fehler: Zahl als Text

Daten aus Eingabefeldern oder von einer Webseite liegen häufig als String vor. Dann kann eine Rechnung überraschend aussehen:

```
let alter = "16";
let naechstesJahr = alter + 1;


console.log(naechstesJahr); // 161
```

JavaScript verbindet hier Text: `"16"` und `1` werden zu `"161"`.

Wenn du wirklich rechnen möchtest, wandelst du den String mit `Number()` in eine Zahl um:

```
let alter = "16";
let naechstesJahr = Number(alter) + 1;

console.log(naechstesJahr); // 17
```

`Number()` brauchst du noch nicht in jedem Detail zu beherrschen. Merke dir aber: Steht eine Zahl in Anführungszeichen, ist sie zunächst Text.

Zusammenfassung

Datentyp	Beispiel	Bedeutung
`number`	`42`, `3.14`, `-5`	Zahlen für Berechnungen
`string`	`"Hallo"`	Texte und Zeichen
`boolean`	`true`, `false`	Wahrheitswerte
`typeof`	`typeof name`	Prüft den Datentyp

Die Anführungszeichen sind besonders wichtig:

```
let echteZahl = 5;
let textMitZahl = "5";

let wahrheitswert = true;
let textMitTrue = "true";
```

`echteZahl` und `wahrheitswert` haben andere Datentypen als `textMitZahl` und `textMitTrue`.

Übung: Ein kleines Profil

Lege Variablen für ein Profil an:

1. einen Namen als String,
2. ein Alter als Zahl,
3. eine Information, ob die Person JavaScript lernt, als Boolean,
4. eine Begrüßung, die alle Informationen ausgibt.

Eine mögliche Lösung:

```
let name = "Samira";
let alter = 21;
let lerntJavaScript = true;

console.log(`Hallo, ich bin ${name}.`);
console.log(`Ich bin ${alter} Jahre alt.`);
console.log(`Lerne ich JavaScript? ${lerntJavaScript}`);
```

Prüfe anschließend die Datentypen in der Konsole:

```
console.log(typeof name);
console.log(typeof alter);
console.log(typeof lerntJavaScript);
```

Ändere zum Testen einmal `alter` zu `"21"` und beobachte, wie sich die Ausgabe von `typeof alter` verändert.

Revision #1
Created 2026-07-24 13:44:16 UTC by art10m
Updated 2026-07-24 13:44:16 UTC by art10m