

LU01c - JSON-Datei einlesen und ausgeben

Was ist JSON?

JSON (JavaScript Object Notation) ist ein textbasiertes Datenformat für Datenaustausch, Konfigurationen und API-Antworten. In JavaScript ist JSON besonders bequem, weil es der Objekt-Notation sehr ähnelt.

Typische JSON-Struktur (Datei person.json):

```
{
  "name": "Anna",
  "alter": 30,
  "ist_aktiv": true,
  "hobbys": ["Lesen", "Radfahren"],
  "adresse": {
    "stadt": "Berlin",
    "plz": "10115"
  }
}
```

Das fs-Modul

Grundsätzlich benötigen wir das Modul *fs* (File System) um Daten aus Files ins JavaScript einlesen zu können. Damit müssen wir die Funktionen zum Einlesen der Daten aus der JSON-Datei nicht selbst schreiben. Das *fs-Modul* bietet uns verschiedene, sehr praktische Methoden wie *readFileSync*, die alle Werte aus der JSON-Datei in ein entsprechendes JS-Objekt einliest.

Funktion	Zweck
<code>fs.readFile()</code>	Datei asynchron lesen
<code>fs.readFileSync()</code>	Datei synchron lesen
<code>fs.writeFile()</code>	Datei speichern/anlegen
<code>fs.mkdir()</code>	Ordner erstellen
<code>fs.unlink()</code>	Datei löschen
<code>fs.existsSync()</code>	Prüfen, ob eine Datei existiert

Synchrones Einlesen

einlesenSync.js

```
const fs = require("fs"); // nicht vergessen
//
// Datei einlesen
const jsonText = fs.readFileSync("person.json", "utf8");
```

```
//  
// In Objekt umwandeln  
const daten = JSON.parse(jsonText);  
//  
console.log(typeof daten); // object  
console.log(daten);  
//  
// Einzelfelder ausgeben  
console.log("Name:", daten.name);  
console.log("Alter:", daten.alter);  
console.log("Aktiv:", daten.ist_aktiv);  
//  
// Hobbys  
console.log("Hobbys:");  
daten.hobbys.forEach(hobby => console.log(" -", hobby));  
//  
// Adresse  
console.log("Adresse:");  
console.log("  Stadt:", daten.adresse.stadt);  
console.log("  PLZ:", daten.adresse.plz);
```

Asynchrones Einlesen

einlesenAsync.js

```
const fs = require("fs");  
//  
// Asynchrones Einlesen  
fs.readFile("person.json", "utf8", (err, data) => {  
  if (err) {  
    console.error("Fehler beim Lesen:", err);  
    return;  
  }  
  //  
  // JSON umwandeln  
  const daten = JSON.parse(data);  
  //  
  console.log(typeof daten); // object  
  console.log(daten);  
  //  
  console.log("Name:", daten.name);  
  console.log("Alter:", daten.alter);  
  console.log("Aktiv:", daten.ist_aktiv);  
  //  
  console.log("Hobbys:");  
  daten.hobbys.forEach(hobby => console.log(" -", hobby));  
  //  
  console.log("Adresse:");
```

```
  console.log("  Stadt:", daten.adresse.stadt);
  console.log("  PLZ:", daten.adresse.plz);
});
```

Auf einzelne Werte zugreifen

```
console.log(daten.name);           // Anna
console.log(daten.alter);          // 30
console.log(daten.hobbys[0]);       // Lesen
console.log(daten.adresse.stadt);   // Berlin
//
// optionaler Zugriff mit optional chaining
console.log(daten.adresse?.stadt ?? "unbekannt");
```

JSON formatiert ausgeben

Rohes Objekt:

```
console.log(daten);
```

Formatiert als JSON-String:

```
const jsonString = JSON.stringify(daten, null, 4);
console.log(jsonString);
```

Parameter bei `JSON.stringify`:

1. Argument: das Objekt
2. Argument: Replacer (meist null)
3. Argument: Anzahl Leerzeichen fuer Einrueckung (4 = gut lesbar, nicht komplett uebertrieben)

einlesenAsyncFormatiert.js

```
const fs = require("fs");
//
// JSON einlesen (asynchron)
fs.readFile("person.json", "utf8", (err, data) => {
  if (err) {
    console.error("Fehler beim Lesen:", err);
    return;
  }
  //
  const daten = JSON.parse(data);
  //
  console.log("Original-Objekt:", daten);
  //
  // Objekt zurück in JSON-String wandeln
```

```
const jsonString = JSON.stringify(daten, null, 2); // null,2 = schön
formatiert
//
console.log("\nAls JSON-String:");
console.log(jsonString);
//
// Optional: neue Datei schreiben
fs.writeFile("person_copy.json", jsonString, (err) => {
  if (err) {
    console.error("Fehler beim Schreiben:", err);
    return;
  }
  console.log("\nDatei 'person_copy.json' wurde erstellt.");
});
});
```



Volkan Demir

From:

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

Permanent link:

<https://wiki.bzz.ch/de/modul/m307/learningunits/lu01/03?rev=1765351574>

Last update: **2025/12/10 08:26**

