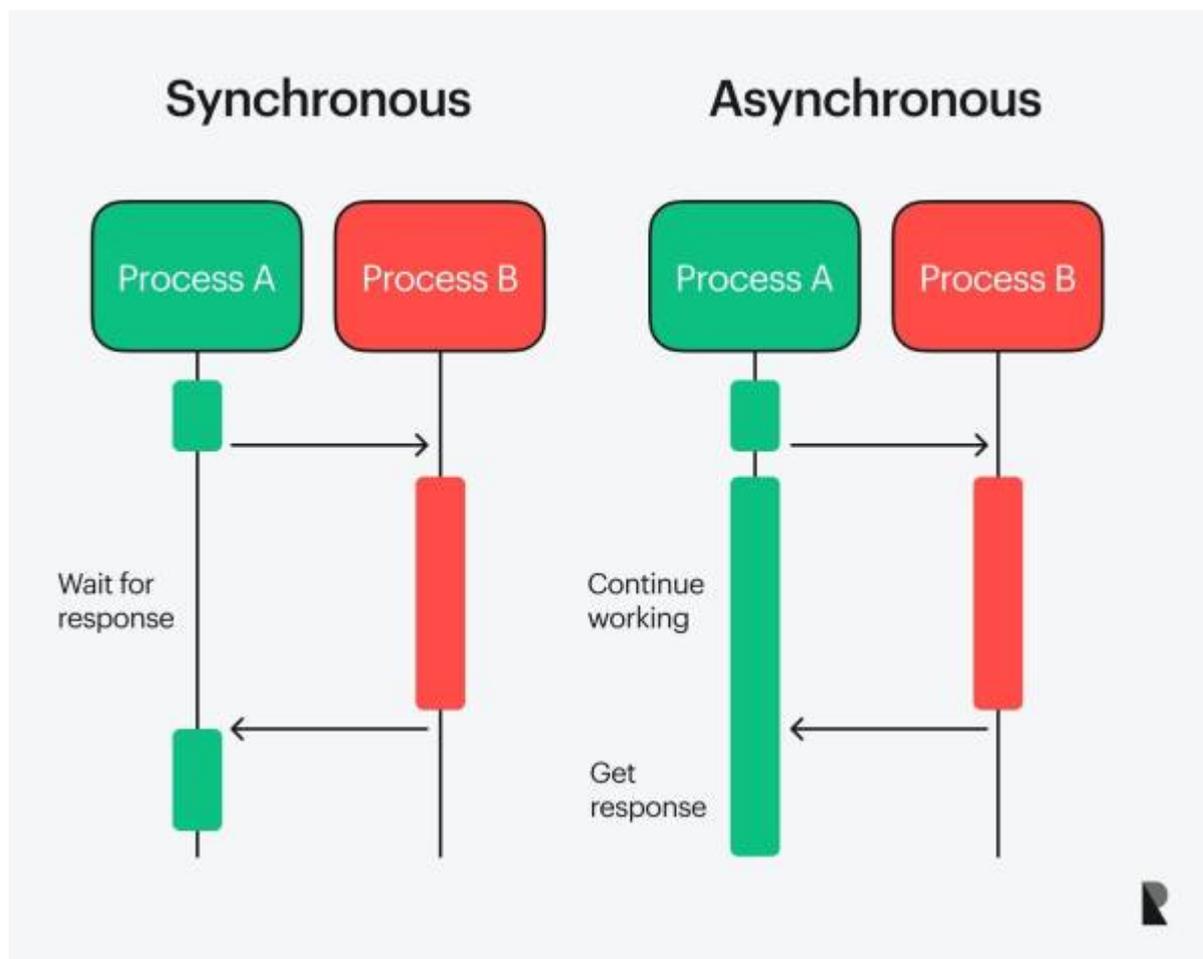


# LU01b - Synchrone VS Asynchrone Verarbeitung

Grundsätzlich gibt es zwei Varianten in JS, wie Daten eingelesen werden können:

- Synchron
- Asynchron

Beim synchronen Einlesen wartet JavaScript, bis die Datei komplett gelesen wurde - und macht in der Zeit gar nichts anderes. Beim asynchronen Einlesen liest Node.js die Datei im Hintergrund ein und kann waehrenddessen weiterarbeiten. Das eine blockiert, das andere nicht.



## Vergleich Sync VS Async

Die nachfolgende Tabelle zeigt die wesentlichen Unterschiede

Art	Verhalten	Vorteil	Nachteil
<b>synchron</b> (`readFileSync`)	Programm pausiert, bis Datei fertig eingelesen ist	einfach, gut fuer schnelle Rumprobieren	blockiert den gesamten Ablauf - bei grossen Dateien oder vielen Zugriffen fatal

Art	Verhalten	Vorteil	Nachteil
<b>asynchron</b> (`readFile`)	Datei wird im Hintergrund gelesen, Code kann weiterlaufen	performant, ideal fuer Server/mehrere Zugriffe	Rueckgabewert kommt erst spaeter per Callback/Promise

### Beispiel für Synchrones Einlesen

```
const fs = require("fs");  
const daten = fs.readFileSync("data.json", "utf8");  
console.log("Weiter gehts erst, wenn das oben fertig ist.");
```

### Beispiel für Asynchrones Einlesen

```
const fs = require("fs");  
fs.readFile("data.json", "utf8", (err, daten) => {  
  console.log("Ich komme spaeter – Datei ist jetzt fertig eingelesen.");  
});  
console.log("Dieser Text erscheint direkt danach – ohne warten.");
```

## Lernvideo

Lernvideo - 5': What is asynchronous javascript-code?	<a href="#">what_is_asynchronous_javascript_code.mp4</a>
Lernvideo - 8': Asynchronous Vs Synchronous Programming	<a href="#">asynchronous_vs_synchronous_programming.mp4</a>



Volkan Demir

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

Permanent link: <https://wiki.bzz.ch/de/modul/m307/learningunits/lu01/02?rev=1765281187>

Last update: **2025/12/09 12:53**

