

Aufgabe 1: Verständnis

Beschreibe die Antwort zu folgenden Fragen (kurz) mit eigenen Worten

Callbacks

1. Was sind Callback-Funktionen in JavaScript?
2. Wozu werden Callback-Funktionen verwendet?
3. Wie werden Callback-Funktionen in JavaScript übergeben?
4. Wie kann man eine anonyme Funktion als Callback übergeben und welche andere Möglichkeit gibt es?
5. Was sind mögliche Alternativen zu Callback-Funktionen in JavaScript?

Promises

6. Was sind Promises in JavaScript?
7. Wozu wurden Promises entwickelt?
8. Welche Zustände können Promises haben und was bedeuten sie?
9. Wie erstellt man ein Promise in JavaScript? (Kurzes Code Beispiel)
10. Wie reagiert man auf den Erfolg oder Misserfolg eines Promises?
11. Wie behandelt man Fehler in Promises?
12. Welche Vorteile bieten Promises im Vergleich zu Callback-Funktionen?

Async Await

13. Warum wurde Async Await entwickelt?
14. Was ist der Unterschied zwischen synchronem Code und Async Await?
15. Wie funktioniert Async Await in JavaScript? (Kurzes Code Beispiel)
16. Welche Schlüsselwörter werden verwendet, um in einer Funktion das await Keyword zu nutzen?
17. Wie funktioniert die sequenzielle Ausführung mehrerer asynchroner Operationen mit Async Await?
18. Welche Vorteile bietet Async Await im Vergleich zu Callback-Funktionen und Promises?