

Technical
lessons

Technischer Unterricht 1





Yuliya Labudova

Team Lead/Frontend Entwicklerin

7 Jahre Erfahrung

EPAM Systems (Adobe, Tele2, Privalia...)

LimeSoda (Meinl am Graben)

Nimbuscloud GmbH

julia.labudova@nimbuscloud.at

Bitte beachten Sie!



Die Kamera muss während der gesamten Unterrichtsstunde eingeschaltet sein.



Wenn Sie während des Unterrichts eine Frage haben, heben Sie bitte die Hand und warten Sie, bis der Lehrer den Gedanken zu Ende geführt hat und Sie fragt. Sie können auch eine Frage im Chat stellen oder wenn der Lehrer sagt, dass eine Fragerunde begonnen hat.



Organisatorische Fragen zur Ausbildung werden mit den Kuratoren geklärt, nicht im thematischen Unterricht.



Verhalten Sie sich respektvoll und ethisch korrekt gegenüber den anderen Kursteilnehmern.



Während des Unterrichts wird es interaktive Aufgaben geben; seien Sie bereit, das Mikrofon einzuschalten oder den Bildschirm auf Aufforderung des Lehrers zu teilen.



Praktische Arbeit



Übung 1

1. Schreiben Sie eine asynchrone Funktion `processNumbers`, die ein Array von Zahlen entgegennimmt und jede Zahl nacheinander (z.B. quadriert) mit `async/await` verarbeitet. Geben Sie ein neues Array mit den Verarbeitungsergebnissen zurück und geben Sie es in der Konsole aus.
2. Erstellen Sie eine asynchrone Funktion `getNumberIfEven`, die eine Zahl entgegennimmt und diese nur dann zurückgibt, wenn sie gerade ist. Andernfalls gibt die Funktion die Nachricht "The number is odd" zurück.
3. Schreiben Sie eine Funktion `processMultipleArrays`, die zwei Arrays von Zahlen entgegennimmt und diese parallel mit `Promise.all` verarbeitet. Kombinieren Sie die Verarbeitungsergebnisse in einem Array und geben Sie es in der Konsole aus.
4. Erstellen Sie eine asynchrone Funktion `retryProcess`, die versucht, die Verarbeitung einer Zahl bis zu dreimal durchzuführen, wenn während der Verarbeitung ein Fehler auftritt (z.B. wenn die Zahl negativ ist). Verwenden Sie eine Pause von 1 Sekunde zwischen den Versuchen. Wenn die Verarbeitung nach drei Versuchen fehlschlägt, geben Sie einen Fehler zurück.
5. Schreiben Sie eine Funktion `batchProcessNumbers`, die ein Array von Zahlen entgegennimmt und diese asynchron in Teilen (z.B. jeweils 3 Zahlen) mit einer Verzögerung zwischen den Verarbeitungen verarbeitet. Geben Sie ein Array mit den Verarbeitungsergebnissen zurück.



Haben Sie Fragen?



Übung 2

1. Schreiben Sie eine Funktion `checkUserObject`, die prüft, ob ein Objekt die Pflichtfelder (z.B. Name und Alter) enthält. Wenn ein Feld fehlt, lösen Sie einen `ValidationError` aus. Behandeln Sie den Fehler mit `try/catch` und geben Sie eine Nachricht in der Konsole aus.
2. Erstellen Sie eine asynchrone Funktion `processData`, die ein Array von Daten verarbeitet. Wenn bei der Verarbeitung ein Fehler auftritt (z.B. ungültige Daten), lösen Sie einen Fehler mit der Nachricht "Failed to process data" aus. Behandeln Sie den Fehler mit `try/catch` und wiederholen Sie den Verarbeitungsversuch.
3. Schreiben Sie eine Funktion `performOperations`, die zwei andere Funktionen aufruft: `transformData` und `validateData`. Wenn eine der Funktionen einen Fehler auslöst, leiten Sie diesen weiter. Behandeln Sie den Fehler in `performOperations` und geben Sie eine Nachricht in der Konsole aus.
4. Erstellen Sie zwei Funktionen: `loadData` und `saveData`. Die erste Funktion lädt Daten aus einem Array, die zweite speichert sie in einem anderen Array. Behandeln Sie Fehler in jeder Funktion separat mit `try/catch` und geben Sie eine Fehlermeldung in der Konsole aus.
5. Erstellen Sie eine asynchrone Funktion `processMultipleTasks`, die mehrere asynchrone Operationen parallel ausführt (z.B. die Verarbeitung mehrerer Datenarrays). Wenn eine der Operationen fehlschlägt, geben Sie die Fehlermeldung für jede fehlgeschlagene Operation zurück. Verwenden Sie `Promise.allSettled`.



Haben Sie Fragen?



Übung 3

1. Erstellen Sie eine Datei `styles.scss` und definieren Sie darin Variablen für Hauptfarben und -größen. Verwenden Sie diese Variablen für die Stildefinition des Hintergrunds, des Textes und der Abstände von Elementen auf der Webseite.
2. Erstellen Sie eine Datei `styles.scss` und verwenden Sie Verschachtelung für die Stildefinition von `.container`, die `.header` und `.footer` enthält. Überprüfen Sie, ob die Stile korrekt angewendet werden.
3. Erstellen Sie ein Mixin `border-radius`, das den Radius für abgerundete Ecken von Elementen festlegt. Wenden Sie dieses Mixin auf mehrere Elemente auf der Webseite an und stellen Sie sicher, dass die Stile korrekt angewendet werden.
4. Teilen Sie die Stile in mehrere Dateien auf (z.B. `_variables.scss`, `_mixins.scss`, `_layout.scss`). Importieren Sie diese Dateien in die Hauptdatei `styles.scss` und verwenden Sie Variablen und Mixins für die Stildefinition.
5. Erstellen Sie ein einfaches Projekt mit SASS. Richten Sie ein npm-Skript ein, um `.scss`-Dateien in `.css` zu kompilieren. Erstellen Sie Stile für mehrere Elemente auf der Webseite und stellen Sie sicher, dass die Kompilierung erfolgreich ist.



Haben Sie Fragen?



Schlusswort

