

React Development Associate

Prüfungscode IFW-REACT-200-01 · Fassung 1.0 · Stand 18.09.2026

Benutzeroberflächen mit React: Komponenten und JSX, State und Ereignisse, Effekte und Hooks sowie Listen, Performance und Refs.

Die Prüfung im Überblick

Bearbeitungszeit 1 Std. 30 Min.	Fragen 70
Bestehensgrenze 70 %	Fragetypen Einfach- und Mehrfachauswahl
Prüfungssprache Deutsch	Zertifikat gültig Unbegrenzt
Durchführung Online, zeitlich überwacht	Preis 49,00 €

Themenblöcke

Block	Thema	Gewichtung	Fragen
1	Komponenten, JSX und Rendern	27 %	19
2	State und Ereignisse	33 %	23
3	Effekte und Hooks	20 %	14
4	Listen, Performance und Refs	20 %	14
Gesamt		100 %	70

Die Gewichtung gibt an, welchen Anteil ein Block an den Prüfungsfragen hat. Die Fragen werden beim Start zufällig und anteilig je Block aus dem Fragenpool gezogen.

Komponenten, JSX und Rendern

Aufbau von Komponenten, Übergabe von Werten und die Frage, was wann gerendert wird.

- 1.1** JSX schreiben
className, Ausdrücke in geschweiften Klammern, Fragmente `<>...</>`
- 1.2** Props verwenden
Werte und Funktionen als Props, Standardwerte, children
- 1.3** Bedingt rendern
ternärer Operator sowie die Falle `0 && ...`
- 1.4** Renderverhalten verstehen
State-Änderung, Rendern der Elternkomponente, genutzter Context

State und Ereignisse

Zustand halten, unveränderlich aktualisieren und auf Eingaben reagieren.

- 2.1** State mit useState verwalten
Setter, erneutes Rendern, Lazy Initializer
- 2.2** State unveränderlich aktualisieren
neue Arrays und Objekte statt push oder direkter Zuweisung
- 2.3** Mehrere Aktualisierungen beachten
Wert aus dem aktuellen Render gegenüber der Updater-Funktion
- 2.4** Ereignisse behandeln
Funktion übergeben statt sie beim Rendern aufzurufen
- 2.5** Formularfelder binden
Controlled und Uncontrolled Components, onChange
- 2.6** Geteilten State anheben
Lifting State Up mit Werten und Callbacks als Props

Effekte und Hooks

Seiteneffekte steuern und die Regeln für Hooks einhalten.

- 3.1** useEffect einsetzen
Ausführung nach dem Rendern, Abhängigkeitsarray und seine Wirkung
- 3.2** Aufräumen
Cleanup-Funktion für Intervalle, Abos und offene Verbindungen

- 3.3** Regeln für Hooks einhalten
gleiche Reihenfolge bei jedem Render, keine Bedingungen
- 3.4** Eigene Hooks schreiben
Logik bündeln, Rückgabewerte frei wählen, kein geteilter State

BLOCK 4

20 % · 14 Fragen

Listen, Performance und Refs

Listen stabil rendern, unnötige Arbeit vermeiden und auf das DOM zugreifen.

- 4.1** Listen mit key rendern
stabile Identität; key ist kein Prop der Komponente
- 4.2** Komponenten über key zurücksetzen
ein neuer key erzeugt eine neue Instanz
- 4.3** Berechnungen und Funktionen merken
useMemo, useCallback, React.memo und der Vergleich per Referenz
- 4.4** Refs verwenden
useRef für DOM-Zugriff und Werte ohne erneutes Rendern
- 4.5** Context nutzen
createContext, Provider, useContext, Standardwert ohne Provider

Institut für Webentwicklung · unabhängiges Lernangebot ohne Verbindung zu den genannten Technologieanbietern. Maßgeblich ist die Fassung auf der Zertifizierungsseite.