

Warum React.js für die Webentwicklung?

React.js ist eine beliebte JavaScript-Bibliothek, die von Facebook entwickelt wurde, um benutzerfreundliche und dynamische Benutzeroberflächen (UI) zu erstellen. Gründe, React.js zu verwenden:

1. **Komponentenbasierte Architektur:** UIs werden in wiederverwendbare, unabhängige Bausteine (Komponenten) aufgeteilt.
2. **Virtual DOM:** Verbesserte Performance durch effizientes Rendering und minimale Manipulation des tatsächlichen DOM.
3. **React-Ökosystem:** Eine riesige Community, viele Bibliotheken und Tools wie Redux, Next.js und React Router.
4. **Einfaches Lernen:** JSX vereinfacht die Entwicklung durch eine Mischung aus JavaScript und HTML.
5. **Flexibilität:** Es ist eine Library und kein Framework, wodurch es sich leichter in bestehende Projekte integrieren lässt.

6. auch Entwicklung mobiler Apps möglich (React Native)

Welche Alternativen zu React gibt es?

1. **Vue.js:** Einfach zu lernen, ähnlich wie React, aber oft als „weniger überladen“ empfunden.
2. **Angular:** Vollständiges Framework von Google, geeignet für größere Projekte.
3. **Svelte:** Compiler-basierte Technologie, die das Schreiben von schnelleren Apps vereinfacht.
4. **Preact:** Kleinere und schnellere Alternative zu React, mit ähnlicher API.
5. **Solid.js:** Ähnlich zu React, mit Fokus auf Performance und feinkörniger Reaktivität.

Was ist JSX, und welche Rolle spielt es in React?

JSX (JavaScript XML) ist eine Syntaxerweiterung für JavaScript, die es ermöglicht, HTML-ähnlichen Code direkt in JavaScript zu schreiben.

- Rolle in React:
 - Erleichtert das Erstellen von Komponenten und das Definieren der Struktur.
 - Ermöglicht es, dynamische Inhalte und Logik in einem einzigen Syntaxblock zu kombinieren.
 - JSX wird von Babel in reguläres JavaScript kompiliert, damit es vom Browser verstanden wird.

Was ist der DOM (Document Object Model)?

Das DOM ist eine Programmierschnittstelle für HTML- und XML-Dokumente, die die Struktur eines Dokuments als Baum darstellt. Jedes Element im HTML wird als Objekt (Knoten) im DOM dargestellt. Es ermöglicht Entwicklern:

- HTML-Inhalte dynamisch zu ändern,
- Attribute zu modifizieren,
- Benutzerinteraktionen zu verarbeiten.

Wie sieht die Interaktion mit dem DOM bei React aus?

React verwendet den **Virtual DOM**, eine leichte Kopie des tatsächlichen DOM. Vorteile:

1. **Effiziente Updates:** React berechnet die Unterschiede (Diffing Algorithm) zwischen der alten und neuen Virtual-DOM-Version und aktualisiert nur die veränderten Teile im echten DOM.
2. **Performance:** Reduziert die Anzahl der direkten Manipulationen am echten DOM.
3. **Abstraktion:** Entwickler müssen sich nicht mit komplexen DOM-APIs auseinandersetzen.

JavaScript vs. TypeScript – Worin unterscheiden sich die beiden Sprachen?

1. **JavaScript:** Dynamisch typisierte Skriptsprache für Webentwicklung.
2. **TypeScript:** Superset von JavaScript, das statische Typen und zusätzliche Features wie Interfaces, Generics und Typprüfungen bietet.
 - Vorteile von TypeScript:
 - **Fehlerprävention:** Durch Typen können viele Fehler schon während der Entwicklung erkannt werden.
 - **Bessere Wartbarkeit:** Typdefinitionen erleichtern das Arbeiten in größeren Teams.
 - **Integration in moderne Tools:** Unterstützt IDEs besser, bietet Autovervollständigung und Typvorschläge.

Was ist Gatsby.js, und wofür wird es genutzt?

Gatsby.js ist ein **React-basiertes Framework** für das Erstellen von **statischen Websites** und Web-Apps.

- **Nutzung:**
 - Ideal für Blogs, Dokumentationen und Landing Pages.
 - Nutzt GraphQL, um Daten aus verschiedenen Quellen (CMS, APIs) zu laden.
 - Generiert statische HTML-Seiten für schnelle Ladezeiten und SEO.

Welchen Vorteil liefert uns Gatsby.js beim Erstellen von React-Projekten?

1. **Performance:** Statische Seiten sind schnell und optimiert für den ersten Ladevorgang.
2. **SEO-freundlich:** Bessere Indexierung durch Suchmaschinen.
3. **Out-of-the-box Optimierungen:** Bildoptimierung, Code-Splitting und Prefetching.
4. **Einfacher Datenzugriff:** GraphQL erleichtert die Integration externer Datenquellen.

Was ist Yarn?

Yarn ist ein **Paketmanager** für JavaScript, ähnlich wie npm (Node Package Manager). Es wurde von Facebook entwickelt.

- **Unterschiede zu npm:**
 - Schneller: Caches Abhängigkeiten lokal.
 - Bessere Konsistenz: Lockfiles sichern dieselben Versionen für alle Teammitglieder.
 - Verbesserte Sicherheit: Führt Checksums für Abhängigkeiten durch.

Warum verwendet der Dozent Yarn lieber als npm?

- **Erfahrung mit React:** Yarn wurde von Facebook entwickelt, das auch React geschaffen hat. Der Dozent hat daher vermutlich bessere Erfahrungen mit der Integration von Yarn in React-Projekten gemacht.
- **Stabilität und Geschwindigkeit:** Besonders bei größeren Projekten und Teams bevorzugen viele Entwickler Yarn, da es konsistenter ist und schneller arbeitet.