

Hier sind die wesentlichen Punkte aus dem Dokument, die Sie sich merken müssen:

1. Angular und seine Nutzung

- **Was:** Ein JavaScript UI Binding Framework.
- **Nutzen:** Bindet HTML UI und JavaScript-Modelle, reduziert Code-Aufwand, unterstützt Single Page Applications (SPA) mit Routing und bietet Features wie HTTP, Dependency Injection, Input/Output.

2. Unterschied zwischen AngularJS und Angular

- **AngularJS:** JavaScript, Controller-basiert, nicht handy-kompatibel, kein CLI, kein LazyLoad, kein SEO.
- **Angular:** TypeScript, komponentenbasiert, handy-kompatibel, CLI vorhanden, unterstützt LazyLoad und SEO.

3. Direktiven und deren Typen

- **Direktiven:** Ändern das Verhalten und Aussehen von HTML.
- **Typen:**
 - **Struktur:** Verändert DOM-Struktur (*ngFor, *ngIf).
 - **Attribut:** Verändert Erscheinung oder Verhalten ohne Strukturänderung ([hidden], [ngClass]).
 - **Komponenten:** Benutzerdefinierte HTML-Elemente (@Component).

4. npm und Node_Modules

- **npm:** Node Package Manager, verwaltet Codepakete für Node.js.
- **Node_Modules:** Enthält installierte npm-Pakete und Abhängigkeiten.

5. package.json

- **Inhalt:** Metadaten, Abhängigkeiten, Skripte und Konfigurationsinformationen.
- **Wichtigkeit:** Ermöglicht konsistente und reproduzierbare Verwaltung und Teilen von Projekten.

6. TypeScript

- **Was:** Von Microsoft entwickelte Programmiersprache.
- **Nutzen:** Fügt statische Typen zu JavaScript hinzu, verbessert Fehlererkennung und Codequalität, wird in JavaScript kompiliert.

7. Angular CLI

- **Nutzen:** Vereinfacht Projektverwaltung, Code-Generierung, Builds und Tests.

8. Komponenten und Module

- **Komponente:** Baustein der UI, bestehend aus HTML, CSS und TypeScript.
- **Modul:** Containerstruktur, gruppiert verwandte Komponenten und Dienste.

9. Decorators und Annotations

- **Decorators:** Markierungen an Klassen, Methoden oder Eigenschaften, die Angular Informationen geben.
- **Annotations/Metadaten:** Zusätzliche Informationen, die Angular helfen, den Code zu verstehen.

10. Template

- **Was:** HTML-Code-Schnipsel, das Layout und Struktur einer Komponente definiert.

11. Databinding

- **Arten:**
 - **Interpolation:** Daten von Komponente zur Ansicht.
 - **Eigenschaftsbindung:** Eigenschaften eines HTML-Elements an Komponentendaten.
 - **Ereignisbindung:** Ereignisse von Ansicht zur Komponente.
 - **Zwei-Wege-Datenbindung:** Synchronisation zwischen Ansicht und Komponente.

12. Architektur von Angular

- **Elemente:** Template, Komponenten, Module, Bindings, Direktiven, Services, Dependency Injection.

13. Standalone

- **Nutzen:** Komponenten, Direktiven und Pipes können unabhängig von Modulen deklariert werden.

14. Single Page Application (SPA)

- **Was:** Webanwendung auf einer einzigen HTML-Seite, aktualisiert nur notwendige Teile dynamisch.
- **Vorteile:** Schneller, bessere Benutzererfahrung, effizienter.

15. Routing in Angular

- **Was:** Navigation zwischen verschiedenen Seiten/Ansichten.
- **Implementierung:** URLs definieren und Komponenten zuordnen.

16. Lazy Loading

- **Was:** Laden nur der benötigten Teile der Anwendung, verbessert Geschwindigkeit.
- **Implementierung:** Modulerstellung, Routing-Konfiguration für Lazy Loading.

17. Services

- **Was:** Klassen für Datenabruf, -speicherung und -verarbeitung, trennen Logik von Komponenten.

18. State Management

- **Nutzen:** Zentralisiert Daten- und UI-Zustand, verbessert Konsistenz und Wartbarkeit.

19. Dependency Injection

- **Was:** Automatische Bereitstellung von Objekten oder Services durch Angular.
- **Vorteile:** Entkopplung, erleichtert Testing und Wartung.

Nutzen Sie diese kompakten Informationen für Ihre Vorbereitung und merken Sie sich die Kernpunkte zu jedem Thema.