

# Compiler und Buildsystem in TypeScript

---

## TypeScript Compiler (tsc)

Der TypeScript Compiler (**tsc**) übersetzt TypeScript-Code (**.ts**-Dateien) in JavaScript.

### Installation

```
npm install -g typescript
```

### Verwendung

- Kompilieren einer einzelnen Datei: **tsc index.ts**
- Automatische Kompilierung bei Änderungen: **tsc index.ts --watch**

### Nützliche CLI-Optionen

- **tsc --init**: Erstellt eine Standard-**tsconfig.json**.
- **tsc --strict**: Aktiviert strikte Typprüfung.
- **tsc --target**: JavaScript-Zielversion festlegen.
- **tsc --outDir**: Ausgabeordner definieren.
- **tsc --watch**: Automatische Kompilierung bei Änderungen.

## tsconfig.json

Die **tsconfig.json** ist die zentrale Konfigurationsdatei für TypeScript-Projekte.

### Wichtige CompilerOptions

- **target**: JavaScript-Zielversion (z.B. **ES6**, **ES2020**, **ESNext**).
- **module**: Modulsystem (z.B. **CommonJS**, **ESNext**).
- **outDir**: Ausgabeverzeichnis für kompilierte Dateien.
- **include**: Eingeschlossene Dateien.
- **exclude**: Ausgeschlossene Dateien.
- **baseUrl**: Basisverzeichnis für Imports.
- **paths**: Alias für Module.

## Beispielsetup für ein Webprojekt (React mit Vite)

Vite ist ein schnellerer und effizienterer Build-Tool-Ersatz für Create React App.

1. Erstelle ein neues React-Projekt mit Vite:

```
npm create vite@latest my-react-app --template react-ts
cd my-react-app
npm install
```

2. Starte das Projekt im Entwicklungsmodus:

```
npm run dev
```

## Debugging von TypeScript-Code

TypeScript erzeugt Source Maps, um den ursprünglichen TypeScript-Code im Browser debuggen zu können.

1. Öffne die Developer Tools im Browser.
2. Wechsle in den Tab Sources.
3. Setze Breakpoints in den `.tsx`-Dateien und überprüfe den Code während der Laufzeit.

## Linting

ESLint ist das Standard-Tool für TypeScript-Projekte, um Code-Qualität zu verbessern, Fehler frühzeitig zu erkennen und Konsistenz sicherzustellen. Für React-Projekte kann `eslint-plugin-react` verwendet werden, um React-spezifische Regeln zu aktivieren.

## Polyfills

Polyfills sind JavaScript-Codefragmente, die fehlende Funktionen in älteren Umgebungen bereitstellen. `core-js` ist eine weit verbreitete Bibliothek für Polyfills.

```
npm install core-js
```

Verwendung in TypeScript:

```
import 'core-js/stable';
import 'regenerator-runtime/runtime';
```