

Fortgeschrittenes Typsystem in TypeScript

Union Types

Union Types ermöglichen, dass eine Variable mehrere Typen annehmen kann.

Intersection Types

Intersection Types kombinieren mehrere Typdefinitionen zu einem neuen Typ.

Discriminated Unions

Eine Discriminated Union ist ein spezieller Union Type, bei dem jeder Typ durch eine eindeutige Eigenschaft (den sogenannten Diskriminator) identifiziert wird.

Tuple Types

Tuple Types ermöglichen es, Arrays mit einer festen Anzahl und Reihenfolge von Typen zu definieren.

Mapped Types

Mapped Types ermöglichen es, bestehende Typen zu transformieren.

Conditional Types

Conditional Types sind typsichere Bedingungen, die zur dynamischen Typanpassung genutzt werden können.

Type Queries

Type Queries (`typeof` und `instanceof`) ermöglichen es, Typen zur Laufzeit zu prüfen und abzuleiten.

- `typeof`: Ermittelt den Typ einer Variable zur Laufzeit.
- `instanceof`: Prüft, ob ein Objekt von einer bestimmten Klasse stammt.

Type Guards

Type Guards sind spezielle Typprüfungen, die TypeScript helfen, den genauen Typ einer Variablen zu erkennen.

- `typeof` Type Guard
- `instanceof` Type Guard
- Benutzerdefinierte Type Guards
- `in` Type Guard