

Collections in TypeScript

TypeScript bietet verschiedene Datenstrukturen zur Speicherung und Verarbeitung von Daten:

Arrays

Arrays speichern geordnete Werte eines Typs. Sie sind die gängigste Datenstruktur zur Speicherung von Listen. Arrays bieten viele eingebaute Methoden für Verarbeitung und Manipulation:

- **map**: Elemente transformieren
- **filter**: Elemente herausfiltern
- **reduce**: Werte kombinieren
- **includes**: Prüfen, ob ein Wert enthalten ist
- **find**: Erstes Element, das einer Bedingung entspricht
- **some**: Prüfen, ob mindestens ein Element die Bedingung erfüllt
- **every**: Prüfen, ob alle Elemente die Bedingung erfüllen
- **sort**: Array sortieren
- **slice**: Ein Teil des Arrays extrahieren
- **splice**: Elemente aus dem Array entfernen oder ersetzen

Maps

Map<K, V> speichert Schlüssel-Wert-Paare. Im Gegensatz zu Objekten können Maps beliebige Schlüsseltypen haben. Maps bieten Methoden zur effizienten Verarbeitung von Schlüssel-Wert-Paaren:

- **forEach**: Iteration durch alle Einträge
- **keys**: Alle Schlüssel abrufen
- **values**: Alle Werte abrufen

Sets

Ein **Set** speichert nur einzigartige Werte. Es eignet sich für Sammlungen ohne Duplikate. Sets haben spezielle Methoden zur Verarbeitung einzigartiger Werte:

- **has**: Prüfen, ob ein Wert existiert

Vergleich: Arrays, Maps und Sets

Struktur	Geordnete Elemente?	Doppelte Werte erlaubt?	Schlüssel-Typen
Array	<input checked="" type="checkbox"/> Ja	<input checked="" type="checkbox"/> Ja	Keine Schlüssel
Map	<input checked="" type="checkbox"/> Nein	<input checked="" type="checkbox"/> Ja	Beliebige Schlüssel
Set	<input checked="" type="checkbox"/> Nein	<input checked="" type="checkbox"/> Nein	Keine Schlüssel