

Syntax und Typsystem in TypeScript

Syntax

TypeScript bietet verschiedene Schlüsselwörter zur Deklaration von Variablen:

- `const`: Für unveränderliche Werte (Block-Scope)
- `let`: Für veränderliche Werte (Block-Scope)
- `var`: Für Variablen mit Funktions-Scope (sollte vermieden werden)

Funktionen können klar typisiert werden:

```
function greet(name: string): string {
  return `Hallo, ${name}!`;
}
```

TypeScript unterstützt bekannte Kontrollstrukturen wie `if`, `switch`, `for`, `while`.

Typsystem

TypeScript bietet folgende primitive Typen:

- `number`: Für Zahlen
- `string`: Für Zeichenketten
- `boolean`: Für Wahrheitswerte
- `any`: Für beliebige Typen (sollte vermieden werden)
- `unknown`: Für unbekannte Typen (sicherere Alternative zu `any`)
- `null`: Für den Wert `null`
- `undefined`: Für den Wert `undefined`
- `never`: Für Funktionen, die niemals einen Wert zurückgeben

TypeScript kann Typen automatisch erkennen (Typinferenz).

Der Strict-Mode aktiviert striktere Regeln für besseren Code (z.B. `noImplicitAny`, `strictNullChecks`, `strictFunctionTypes`).

Coding Kata

1. Erstelle eine Funktion `createUserProfile(name, age, isActive)`, die ein Nutzerobjekt zurückgibt.
2. Implementiere `getUserStatus(age)`, um das Alter in Kategorien einzuteilen.
3. Schreibe `calculateBirthYear(age)`, um das Geburtsjahr zu berechnen.
4. Implementiere `logUserInfo(name, age)`, das `age` als Zahl oder String akzeptiert.