

Coding Konvention für TypeScript

TypeScript

```
const CONTENT = document.getElementById("main-container") as HTMLElement;

function init(): void {
    CONTENT.innerHTML += renderMain();
}

function renderMain(): string {
    return `
        <div> Hallo Welt </div>
    `;
}
```

1. Allgemein

Dateinamen in kebab-case	user-utils.js
max. 14 Zeilen pro Funktion	
Semikolons verwenden	const NAME: string = "Max";
Kein <code>any</code> verwenden (wenn möglich)	besser: ein exakter Typ oder <code>unknown</code>

2. Namensgebung

Funktionen im camelCase	function getUser() {...};
Klassen in PascalCase	class UserProfile {};
Interfaces in PascalCase	interface User {};
Konstanten in UPPER_CASE	const MAX_RETRIES = 3;
Typen in PascalCase	type UserID = string;

3. Formatierung

2 Leerzeichen für die Einrückung	return user;
imports gruppieren (Std, Dritt, Lokal)	import os\import requests\from .import tools
Typen und Rückgabewerte explizit angeben	function loadData(): Promise<Data>

4. Kommentare

TSDoc für Funktionen & Methoden	/** Gibt den user zurück. */
---------------------------------	------------------------------

5. Clean Code

Eine Aufgabe pro Funktion	function calculateTotal():number {}
Keine Magic Numbers	const MAX_USERS = 100;
HTML auslagern statt inline	
Lesbare Bedingungen	if (isUserActive) statt if (x)

6. Quellen

- [Kommentar-Standards \(TSDoc\)](#)
- [Formatierung, Einrückung, Namensgebung](#) - Im Arbeitsalltag-Kontext (Bsp. Airbnb)
- [TypeScript StyleGuide](#)
- [Clean Code](#)
- [Funktionslänge & Funktionswirkung](#)