

# Web Engineering 2

## Styleguide

### Dokumentation

im Studiengang Informatik  
mit Schwerpunkt Mobile Informatik

an der DHBW Ravensburg  
Campus Friedrichshafen

von

Tobias Welti, Luca Kaiser, Tobias Jung, Danny Seidel, Andreas Holzapfel, Sandra  
Knefel, Thomas Schmitz

**Bearbeitungszeitraum:** 26.04.22 - 12.07.22

**Dozent:** Friedhelm Koch

# Inhaltsverzeichnis

<b>1</b>	<b>Dokumentation im Code</b>	<b>1</b>
1.1	Intention . . . . .	1
1.2	Kommentare . . . . .	1
1.3	Beispiel . . . . .	1
<b>2</b>	<b>Linterkonfiguration</b>	<b>2</b>
2.1	Intention der Linterkonfiguration . . . . .	2
2.2	ESLint . . . . .	2
2.3	Stylelint . . . . .	2
2.4	Htmlhint . . . . .	2

# 1 Dokumentation im Code

## 1.1 Intention

Um ein einheitliches Erscheinungsbild des Codes zu gewährleisten und um später so gut wie nahtlos an das Projekt anbauen zu können, werden hier einige Punkte für die Gestaltung der Kommentare und des Codes aufgelistet, dass ein externer Entwickler den Code verstehen kann.

## 1.2 Kommentare

Annotationen:

// einfacher Kommentar

{/\* \*/} Kommentar in einem Renderblock

Gewünschte Verwendung:

Als Grundkomponente für die Dokumentation des Codes werden JSDoc tags verwendet. Als Erstes bitte verständlich und in einfachen Worten für Dritte schreiben, was die Funktion macht. Hierfür werden übergebene Parameter mit @param und Datentyp angegeben. Bitte auch die Funktion angeben und gegebenenfalls Rückgabe (werte) angeben.

## 1.3 Beispiel

```
1 class App extends Component {
2     /* Description of function
3     @parameter {datatype/element}
4     @return */
5     }
6
7     render(){
8         return(
9             <div>
10                <h></h>
11            </div>
12        )
13    };
14 }
```

## 2 Linterkonfiguration

### 2.1 Intention der Linterkonfiguration

Um die Syntax, Semantik und die Formatierung des Codes statisch zu überprüfen wurden auf GitHub Actions Linter für JavaScript, CSS und HTML konfiguriert. Nach jedem Push von neuen Changes wird der Code von diesem Linter überprüft und somit kann vermieden werden, dass sich die Dateien in der Formatierung und Syntax unterscheiden.

### 2.2 ESLint

ESLint wird zum linten von JavaScript verwendet.

```
npm install eslint
```

```
npx eslint . - -ext, js, jsx
```

### 2.3 Stylelint

Für die CSS files wird stylelint verwendet.

```
npm install stylelint
```

```
nnpx stylelint "**/*.css,scss"
```

### 2.4 Htmlhint

Wird zur Überprüfung der HTML-Syntax verwendet.

```
npm install htmlhint
```

```
npx htmlhint "**/*.html"
```