

## WBE-Praktikum 11

# UI-Bibliothek (Teil 1)

## Aufgabe 1: SJDON ins DOM abbilden

Entwickeln Sie eine JavaScript-Funktion *renderSJDON*, welche die im Unterricht eingeführte – SJDON genannte – Notation ins DOM überträgt:

```
const element =  
  ["div", {style: "background: salmon"},  
   ["h1", "Hello World"],  
   ["h2", {style: "text-align:right"}, "from our library"] ]
```

```
let appRoot = document.getElementById("app")  
renderSJDON(element, appRoot)
```

Der Aufruf sollte die in *element* referenzierte Baumstruktur im Browser anzeigen.

## Abgabe

Geben Sie die Funktion *renderSJDON* ab. Um keine unerwünschten Interaktionen mit dem DOM der Abgabeseite zu bekommen, wird der Code auf dem Server nicht ausgeführt.

Abgabeserver: <https://radar.zhaw.ch/python/UploadAndCheck.html>  
Name Praktikum: WBE10  
Dateiname: render-sjdon.js  
Funktionsname: renderSJDON  
Export im Script: nicht nötig

## Aufgabe 2: «Vier gewinnt» verbessert (Miniprojekt)

An dieser Stelle sollen noch ein paar Anpassungen am Spiel vorgenommen werden:

- Verwenden Sie Ihre *render*-Funktion aus Aufgabe 1, um das Spielfeld für das Spiel aus dem letzten Praktikum anzulegen.
- Anstelle des Ladens und Speicherns des Spielzustands von/auf einem Server soll der Spielzustand nun bei Bedarf im LocalStorage des Browsers gespeichert werden können, so dass das Spiel nicht von der Verfügbarkeit eines Servers abhängig ist.

### Hinweis zur Umsetzung

Wie Sie den zweiten Punkt umsetzen bleibt Ihnen überlassen. Eine einfache Variante ist, die Buttons «Laden» und «Speichern» auf der Seite zu lassen und ihre Handler so zu ersetzen, dass die Daten im LocalStorage persistiert werden. Eine komfortablere Version ist, die bestehenden Buttons umzubenennen in «Laden vom Server» und «Speichern auf Server» (allenfalls deaktiviert, wenn kein Service erreichbar ist) und neue Buttons «Laden» und «Speichern» für den Browserspeicher anzulegen.