

Lab03

Styling & Fluent-UI

1. ToDo-Liste stylen mit Inline-Styles (index.css nutzen und inline-styles)

z.B.

index.css

```
h1 {  
  font-family: 'Courier New', Courier,  
  monospace;  
}
```

```
<h1 style={{backgroundColor: 'red'}}>
```

Oder Styled Components

```
npm install styled-components
```

```
import styled from 'styled-components';
```

```
const Title = styled.h1`  
  font-size: 1.5em;  
  text-align: center;  
  color: palevioletred;  
`;  
;
```

2. Fluent UI installieren

Zu vorhandenem Projekt hinzufügen:

```
npm install @fluentui/react
```

Oder neues Projekt mit Fluent-UI Template erstellen (hier das Beispiel für TypeScript):

```
npx create-react-app my-app --template @fluentui/cra-template
```

(Stand Oktober 2022: Achtung! Leider unterstützt unser aktuelles React nicht Fluent UI. Wir können entweder die React-Version downgraden (empfehle ich nicht, weil andere Abhängigkeiten entstehen können) oder mit - - force die Installation erzwingen)

```
npm install @fluentui/react - force
```

3. Nun haben wir die Möglichkeit, Fluent UI zu nutzen

```
import { PrimaryButton } from  
'@fluentui/react/lib/Button';
```

```
<PrimaryButton>Microsoft Office Style  
</PrimaryButton>
```

4. Doku:

<https://developer.microsoft.com/en-us/fluentui#/controls/web>

5. Custom Styles können mit Inline-Styles hinzugefügt werden, zwei Wege:

```
const styles = {color: 'red'}  
  <TextField style={{backgroundColor:  
'red', padding: '20px'}}></TextField>  
    <PrimaryButton  
style={styles}>Click Me</PrimaryButton>
```

6. Aufgabe: Erstelle eine kleine Office-App, die mindestens ein Textfeld und Buttons enthält. Anfang:

```
import { TextField } from
 '@fluentui/react';
import { PrimaryButton } from
 '@fluentui/react/lib/Button';

const styles = {color: 'red'}

export function MyOfficeApp(){
  return(
    <div>
      <TextField
style={{backgroundColor: 'red', padding:
'20px'}}></TextField>
      <PrimaryButton>Click
Me</PrimaryButton>
    </div>
  )
}
```