React
שיעור שמיני

React

React material

- Material-UI היא ספרייה המאפשרת לנו לייבא ולהשתמש ברכיבים שונים כדי ליצור ממשק משתמש ביישומי React שלנו. זה חוסך כמות משמעותית של זמן מכיוון שהמפתחים לא צריכים לכתוב הכל מאפס.(בדומה ל bootstrap), ההבדל ש Material UI הוא ספריה הקיימת רק ל react, לעומת bootstrap שניתן לממש בהרבה שפות.
- שלבים:
התקנת הספרייה:

```
react_course> npm install @material-ui/core
```

כל רכיב שנרצה לעצב ע"י material נצטרך לייבא מהספריה.
לדוגמא – פיתוחים מגוונים המעוצבים ע"י material:

```
import { AppBar, TextField, Toolbar, Typography } from '@material-ui/core';
```

```
<Button color="red" variant="contained">Hello World</Button>
<TextField id="name" label="Name" variant="outlined" />
<AppBar position='static'>
    <Toolbar>
        <Typography>React Navbar Example</Typography>
    </Toolbar>
</AppBar>
```

המשך React material

- ניתן גם להוסיף attributes מסויימים כגון disabled... לדוגמא לכפתור לא מאופשר:
-
```
<Button disabled variant="contained">Hello World</Button>
```

כמובן ניתן להוסיף תנאים על איפשור הכפתור את התנאים נכתוב בתוך הסוגריים.

```
<Button disabled = {} color="red" variant="contained">Hello World</Button>
```

- דוגמאות ל materials נוספים תכלו למצוא בקישור : https://mui.com/material-ui/

useId hook

- useId hook משמש ליצירת מזהים ייחודיים שיהיו יציבים על פני השרת והלקוח. המזהה תמיד יהיה ייחודי ולא חוזר על עצמו במהלך הפרויקט.
- שלבים:

ייבא את useId מ react

```
import { useId } from "react";
```

ניצור אובייקט של useId

```
const id = useId();
```

באלמנט html נעביר את ה id שיצרנו לאלמנט:

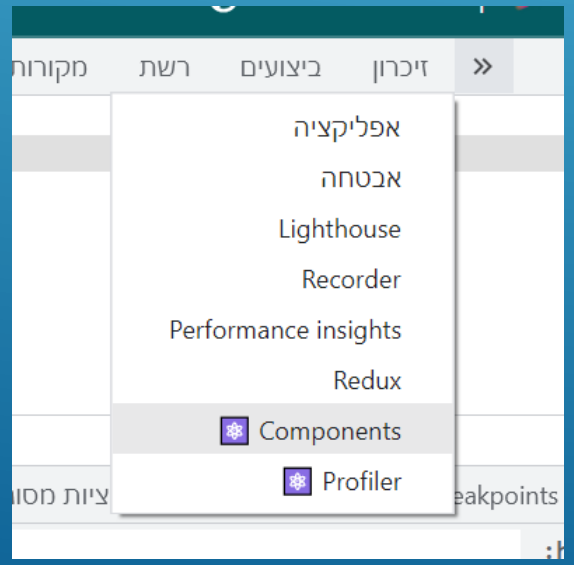
```
<input id={id} type="checkbox" name="react" />
```

- אם נרצה מספר אלמנטים ליצור להם id באמצעות ה useId נעשה באותה צורה אך נוסיף לid עוד ערך ייחודי לכל אלמנט לדוגמא:

```
<input id={id + '-firstName'} type="text" />
<input id={id + '-lastName'} type="text" />
```

React developer tools

- על מנת לראות את המטרה של useDebugValue צריך להתקין את התוסף react developer tools:
https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi
נתקין תוסף זה.
- לאחר ההתקנה כשנריץ את ריאקט נוכל לפתוח את ה devtools ולבחור לדבג מתוך התוסף ע"י לחיצה על components: יוצגו ה components שאנחנו מציגים ב app.js



React

useDebugValue hook

- useDebugValue עוזר לנו לדבג בצורה טובה בהרצה בreact. אם עד היום היינו כותבים דברים ללוג על מנת לדבג, ע"י useDebugValue נוכל להתחבר לstate מסוים ונוכל בתוך אובייקט זה להעביר מידע ל console (לתוך ה react dev tools) ולדבג בצורה הטובה ביותר.
- ניתן לבדוק דיבוג דרכו מתוך ה react develop tools שהתקנו. באופן המוסבר בשקופית הקודמת.
- שלבים:
  נייבא את useDebugValue מ react.
  עכשיו ניתן בכל פעם שנרצה להשתמש באובייקט על מנת לשאול תנאים ולהציג לוגים בהתאם. התנאים נעשים על states שבקומפוננט.

```
import React, { useDebugValue, useState } from "react"

export default function useExampleHook() {
  const [rock, setRock] = useState(true);
  useDebugValue(rock ? "Rocket is Active" : "Rocket is Inactive")
  return [rock, setRock]

}
```

- הפעלה: יש לשים לב שניתן לעשות את זה רק על פונקציות אז ההפעלה היא הפעלת פונקציה.
  כאשר נפתח את ה components ב react dev tools
  נראה את מצב ה state ב component

useDebugValue hook דוגמא נפוצה – קריאות API

- בקריאות API הכי קשה למפות מאיפה מגיעה הבעיה. בד"כ אנחנו רק כותבים ללוג, ובצורה זו לא ניתן לדעת איפה מכל השלבים הייתה הנפילה.
- useDebugValue עוזר לנו לדבג ולכתוב על כל שלב אם עבר או לא וכך ניתן לדבג בצורה הטובה ביותר.
- דוגמא לקריאת API עם שימוש ב useDebugValue hook:

```
const [response, setResponse] = useState([]);
const [error, setError] = useState(null);
const [httpResponse, setHttpResponse] = useState();
useDebugValue(
    httpResponse ? "status code " + httpResponse.status : "no response"
);
useDebugValue(error, (e) =>
    e ? `fetch failed due to ${e.message}` : "fetch successful"
);
useEffect(() => {
    async function fetchFiles() {
        try {
            const response = await fetch(url);
            setHttpResponse(response);
            const json = await response.json();
            setResponse(json);
        } catch (error) {
            setError(error);
        }
    }
    fetchFiles();
}, [setError, setResponse, url]);
```

React