

React

שיעור ראשון

מה זה React

- סביבת הקוד הפתוח React שפותחה על ידי פייסבוק בשנת 2011 ושוחררה כקוד פתוח בשנת 2013
- נועדה לעזור למתכנתים לפתח אפליקציות בקלות בכל מערכת הפעלה.
- השפה עליה בנויה הפלטפורמה מבוססת על הרחבה של שפת Java Script, שנקראת JSX והיא פשוטה ללמידה.
- React מאפשרת לייצר ממשקי משתמש מורכבים מחתיכות קטנות ומבודדות של קוד בשם component

הבדל בין React ל Angular

- אנגולר היא פלטפורמה המסייעת לפיתוח אפליקציות שלכם שפותחה ע"י גוגל. למערכת זו מבנה MVC. קיימות סיפריות רבות באנגולר אשר מסייעות לנו לעשות את הפעולות השותפות – אחת מהן היא Rxjs. נועדה לעזור למתכנתים לפתח אפליקציות בקלות בכל מערכת הפעלה.
- react היא ספרייה של Java Script אשר פותחה ע"י פייסבוק. בריאקט אין לנו מבנה שבו אנחנו חייבים לפתח את האפליקציה שלנו, לכן אנחנו יכולים ליישם כמה פתרונות. אנחנו יכולים לכתוב את האפליקציה במבנה MVC סטנדרטי. לחלופין אנחנו יכולים לממש את redux, שהיא ספרייה שמנהלת את החילופי המידע באפליקציה

התקנה – יצירת סביבת עבודה

הפקודה Create React App יוצרת פרויקט חדש וסביבת עבודה שלבים:

- פתחו תיקייה במיקום בו אתן רוצות שישמר הפרויקט.
- פתחו command line בניתוב התיקייה שפתחתן.
- הריצו את הפקודה `npx create-react-app app_name` שימו לב : אסור שיהיה רווח בשם הפרויקט.
- הריצו את הפקודה `cd app_name` – יעביר את הניתוב לתיקיית הפרויקט.
- הריצו את הפקודה `npm start` – מפעיל ופותח את הפרויקט. בררת המחדל של פתיחת הדפדפן בריאקט הוא `localhost:3000`, אם דפדפן זה כבר תפוס מפרויקט אחר של ריאקט או מפרויקט שונה נקבל שאלה אם אנחנו רוצים להפעיל על פורט שונה – נבחר ב: `yes-Y`.
- בשביל שנוכל לפתח את הפרויקט נפתח אותו בתוכנת visual studio code – סביבת עבודה מומלצת לעבודה עם ריאקט.

הכרות עם הפרויקט – react components

- בreact אנחנו עובדים עם components – חלוקת הקוד לרכיבים עצמאיים הניתנים לשימוש חוזר – כל עמוד או אפילו חלק עמוד בפרויקט שלנו יהיה ב component שונה כך שנוכל לנתב אליו בכל מקום שנרצה בפרויקט ולהשתמש בו היכן שנצטרך. חוסך כתיבת קוד חוזר.
- הפרויקט מתחיל בApp.js component – היא אב ה components שבפרויקט ריאקט, כל התצוגה שנראה במהלך ההרצה מתחיל ב component זה.
- Components בריאקט ניתנים בשני סוגים class component ו function component, הנפוץ הוא כתיבה בצורה של function component הנקרא גם react hooks. – בקורס זה נכיר את שני האפשרויות אך נתמקד ב react hooks.
- שני סוגי ה components מכילים אותם אפשרויות עליהם נלמד במהלך הקורס. המימושים קצת שונים, כדאי להיצמד בפרויקט לסוג אחד של components ולא לערבב בין הסוגים.
- Component הוא רכיב המקבל props (קיצור של מאפיינים – properties) ומחזיר תצוגות – views דרך המתודה render או return

Class component

```
import React from "react";  
class FirstClass extends React.Component {  
  render() {  
    return <h1>Hello, I am the first class</h1>;  
  }  
}  
export default FirstClass;
```

- נייבא את הספרייה React
- React component חייב להתחיל באות גדולה
- React component מממש את רכיב ה render – בה יחזיר את תוכן ה html. – ה view של ה component
- כדי שנוכל להכיר ולהשתמש ב component זה בענודים נוספים נייצא אותה - export

Function components

```
function First() {  
  return <h2>Hello, I am the first!</h2>;  
}  
export default First;
```

- React component חייב להתחיל באות גדולה.
- return הוא כמו רכיב ה render ב class component. בה יחזיר את תוכן ה html – ה view של ה component
- כדי שנוכל להכיר ולהשתמש ב component זה בענודים נוספים נייצא אותה - export

שימוש ב components

- ניתן להכיל את ה components שאנחנו יוצרים בתוך component אחרת או לעשות ניתוב בין components
- הכללת components – component אב מכילה את component הבן.
(לדוגמא פה ה App component מכילה את First Component)

```
function App() {  
  return (  
    <div className="App">  
      <First></First>  
    </div>  
  );  
}  
  
export default App;
```


שימוש ב רכיבי HTML ב components

```
function First(props) {  
  return (  
    <div>  
      <h1 style={{ color: "red"}}>Hello</h1>  
      <button>click me</button>  
    </div>  
  )  
}  
export default First;
```

```
function First(props) {  
  const mystyle = {  
    color: "white",  
    backgroundColor: "DodgerBlue",  
    padding: "10px",  
    fontFamily: "Arial"  
  };  
  return (  
    <div>  
      <h1 style={mystyle}>Hello</h1>  
      <button>click me</button>  
    </div>  
  )  
}  
export default First;
```

```
function First(props) {  
  return (  
    <div>  
      <h1 className="header_style">Hello</h1>  
      <button>click me</button>  
    </div>  
  )  
}  
export default First;
```

- רכיב ה return/render מחזיר אחד של html המכיל את כל תצוגת ה component (כל רכיבי ה html).
- אם נשתמש ביותר מרכיב html אחד ב component - נעטוף את כל הרכיבים ב div עוטף.
- רכיבי ה html הם html רגיל. ניתן להוסיף להם עיצוב באמצעות CSS רגיל
- דוגמא נוספת לעיצוב באמצעות CSS:
- דוגמא נוספת לעיצוב – באמצעות class css - className

props

- Components יכולים גם לקבל ערכים (properties) ולהשתמש בהם. – באמצעות props.
- לדוגמא – עמוד לוגאין בו המשתמש כותב את שמו, נרצה שבעמוד הבית אליו מגיעים מהלוגאין יוצג שמו של המשתמש.

```
<First hobby = "working" name="Efrat"></First>
```

שליחת properties לcomponent:

שימוש ב props:

הגדרת ה component כמקבלת props

שימוש בערכי ה props

```
function First(props) {  
  return <h2>Hello {props.name}, I am the first! and my hobby is: {props.hobby}</h2>;  
}  
export default First;
```

התעסקות עם תמונות

```
import profile from './images/profile.jpeg'
function First() {
  return (
    <div>
      <img src={profile}></img>
    </div>
  )
}
export default First;
```

- התעסקות עם תמונות בריאקט שונה מ-javascript הרגיל.
- שלבים:
- נוסף תמונות לתיקית הפרויקט שלנו.
- נייבא את התמונה לפי הניתוב שלה בתיקייה.
- הצגת התמונה בסוגריים מסולסלים.
- ניתן גם לקבל את ניתוב התמונה ב props ולהציגה.

```
<img src={props.logo}></img>
```

שמירת נתונים - state

- שמירת נתונים ב react components נעשית באמצעות state.
 - לכל component יש state השומר את הנתונים שנרצה ב component זה.
 - ישנו גם state גלובלי – בו ניתן לשמור נתונים השייכים לכל הפרויקט הוא נקרא store ועליו נלמד בהמשך.
- שלבים:

```
import { useState } from 'react';
```

```
const [count, setCount] = useState(0);
```

הסבר: לכל משתנה יש את שמו ואת הפונקציה set שלו באמצעותה נשנה את ערכו. הערך ההתחלתי של המשתנה הוא מה שנאתחל בסוגריים

- דוגמא לשימוש במשתנים ושינוי ערך ה states :

```
<p>You clicked {count} times</p>
<button onClick={() => setCount(count + 1)}>
  Click me
</button>
```

input

- רכיב ה html – input משמש אותנו לשמירת ערכים אותם נבקש מהמשתמש למלא
- נשתמש באירוע onChange – בעת שינוי ערך ה input. כדי לשמור את הערכים באמצעות state
- דוגמא:

```
import { useState } from 'react';
function First(props) {
  var [name, setName] = useState('');
  return (
    <div>
      <input placeholder="Name" onChange={(e) => setName(e.target.value)}></input>
      <h1>{name}</h1>
    </div >
  )
}
export default First;
```

שיעורי בית:

- צרו 2 components מסוג function (hooks).
 - בשני components אלו הוסיפו רכיב input ב html.
 - ב App.js component הפעילו את אחד מה components.
 - ב component זה (המוכל ב App) הפעילו את ה component השניה – זו שאינה נמצאת ב App.js
 - מטרה: שני ה Inputs של שני ה components מוצגים על המסך. בעת שינוי input של אחד מה components שני ה inputs ישתנו – יהיו בעלות אותו ערך.
(שימוש ב state וב props)
- בנוסף: שחקו עם רכיבי HTML, הוסיפו תמונת רקע, השתמשו ב css בהצלחה!