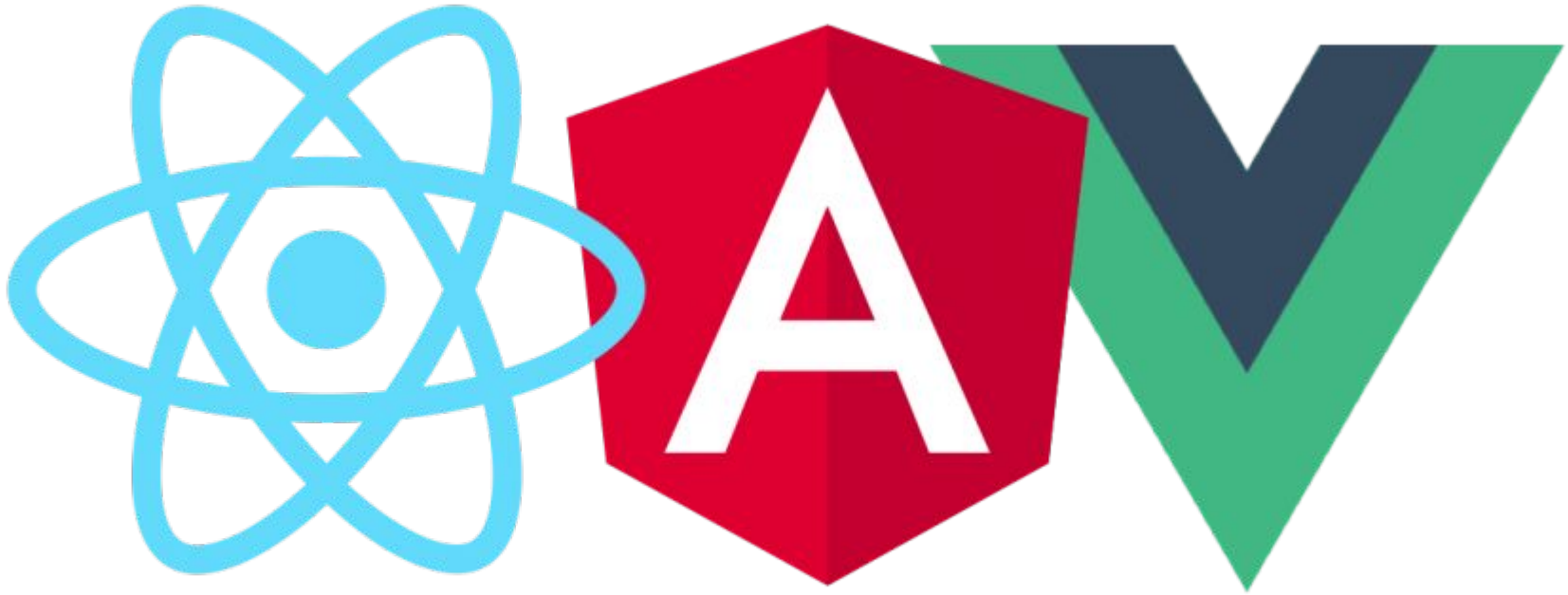
שיעור 14

מתחילים...

React

React - What is it?

- ספריית צד לקוח של js מהנפוצות בעולם.
- נוסדה ע"י חברת Facebook בשנת 2013.
- נועדה לבנייה דינאמית של ממשק המשתמש בעזרת עיבוד "רכיבים" נפרדים.
- ספריית קוד פתוח בשפת js המשמשת לפיתוח ממשק משתמש (טפסים, גלריות תמונות...כל מה שהמשתמש רואה ב web).



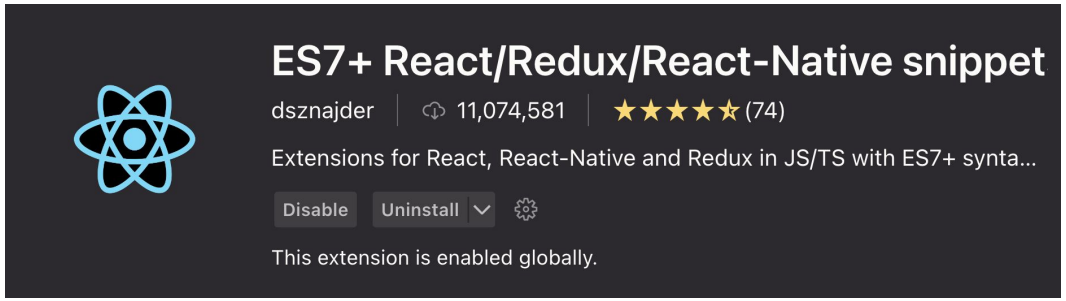
למה
React?

1. קלות בפיתוח אפליקציות .
1. משמשת לאתרים דינאמיים .
1. מבוססת js.
1. נתמך בכל הדפדפנים (ללא הידור/פרשון) .

- React Native

Installing React

1. הורדות תוסף לדפדפן: React developer tools .
2. הורדת תוסף לVS Code :



3. פותחים תיקייה חדשה ב VS Code
4. כניסה ל Terminal

- הרצת הפקודה npm create vite@latest
- לחיצה על y וenter ברגע שנשאל אם נרצה להתקין את create vite
- בחירה בשם הפרוייקט
- בחירה בReact כframework שלנו
- בחירה בJavascript
- המשך שקופית הבאה.

Installing React

```
✔ Select a framework: › React
✔ Select a variant: › JavaScript

Scaffolding project in /Users/eit

Done. Now run:

  cd test
  npm install
  npm run dev
```

לאחר מכן בטרמינל יופיע המלל הבא:

- השם ‘test’ הוא שם הפרוייקט, זה נתון לבחירתנו.
- ניגש לתוך התיקייה החדשה בשם ‘test’ שנוצרה בעזרת ‘cd test’
- נריץ את הפקודה npm install ונמתין שיסיים להוריד הכל.
- על מנת להרים את הפרוייקט שלנו באוויר נריץ ‘npm run dev’

Files in React

1. **index.html:** מסמך html ראשי, הכל עובר דרך root " id ".

1. **main.jsx:** מגדיר את ה root הראשי.

1. **App.js:** הקומפוננטה הראשית של האפליקציה.
(אחראית לקריאת כל הקומפוננטות האחרות).

1. **App.css:** אחראית לעיצוב הקומפוננטה.

Components

רכיב – הינו חלק מהאתר/אפליקציה אשר בנוי מ html/css/js.
כל אתר בנוי מכמה רכיבים שונים אשר מקבלים ומעבירים מידע מאחד לשני ומאפשרים דינמיות רבה ביצירת וחלוקת האתר.

Header

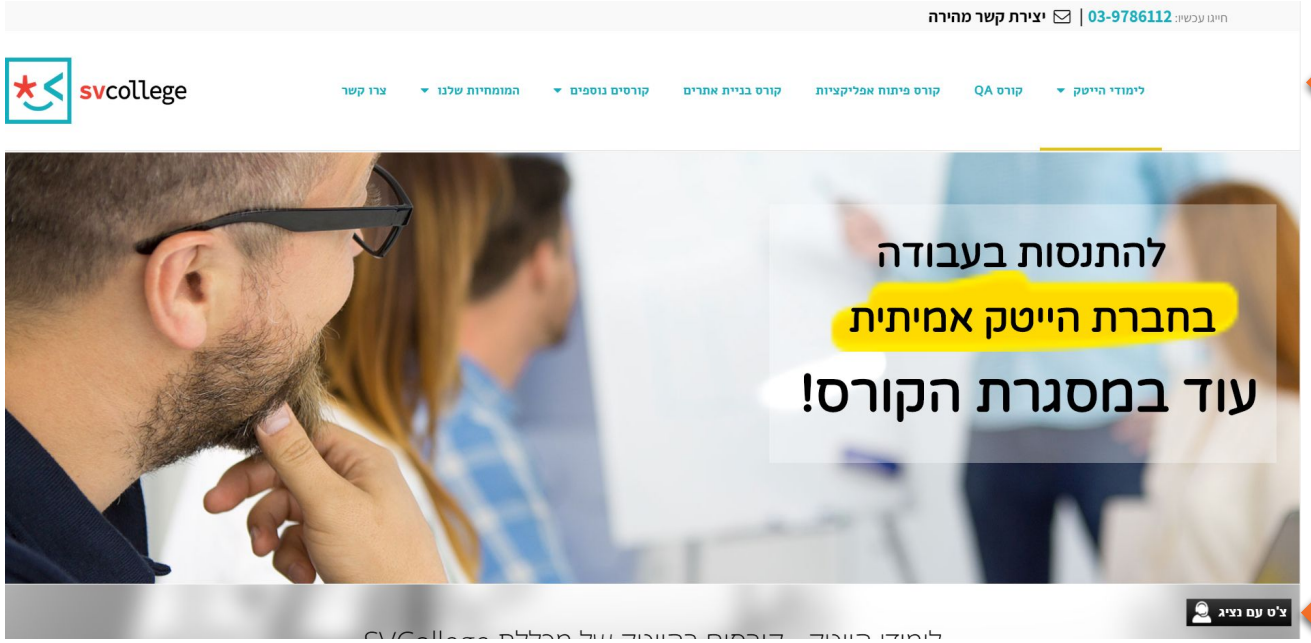
component

Contact

component

Body

component



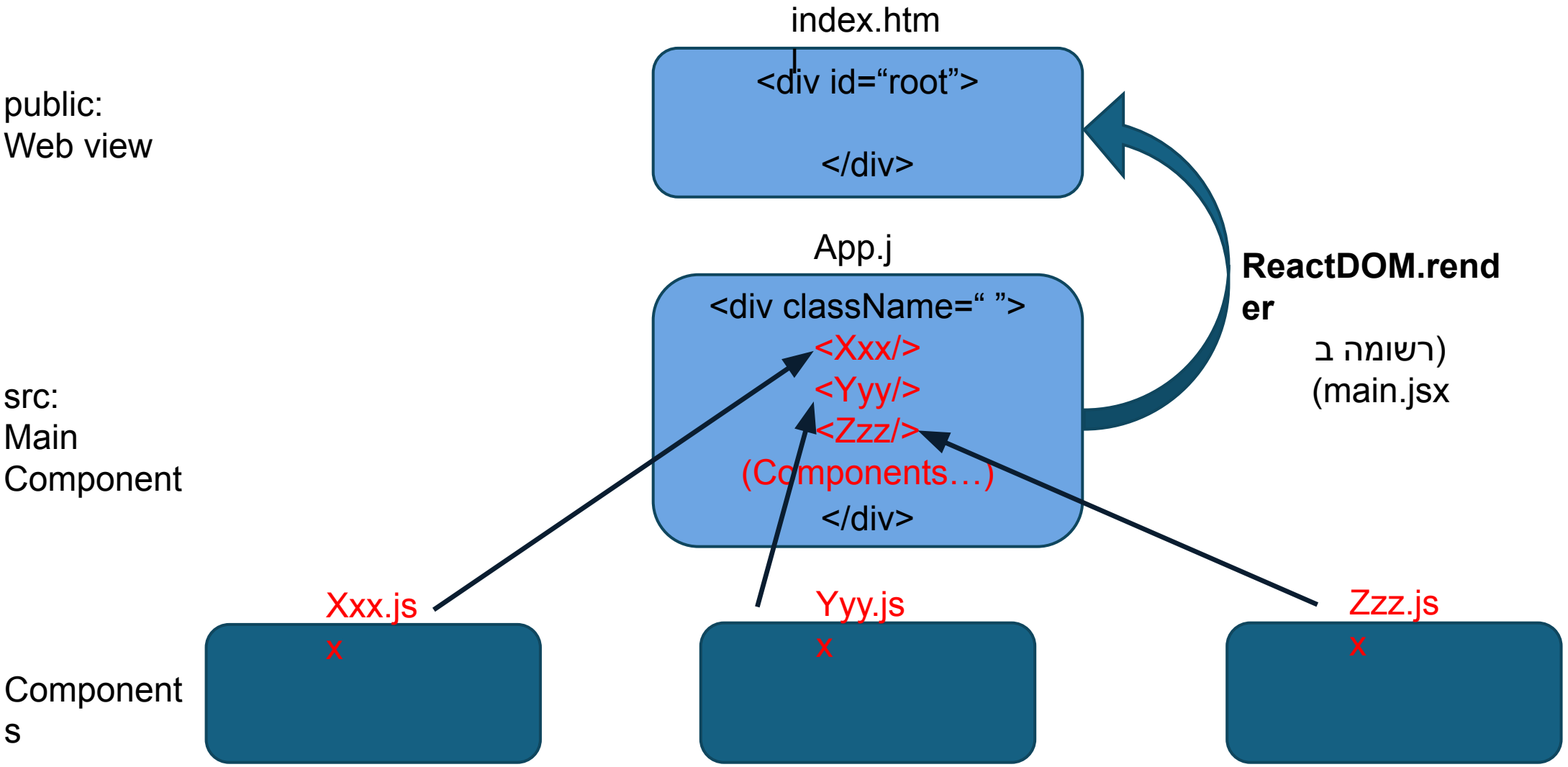
App.js

רכיב ראשי – אחראי לקריאה של רכיבים נוספים.

```
import React from 'react’;  ייבוא ספריית ריאקט

import './App.css';ייבוא קובץ העיצוב

function App() {פונקציה ראשית
    return (חייבת להחזיר אלמנט אחד בלבד
    <div className="App">

    כל הרכיבים יקראו כאן

    </div>
    );
}
export default App;מאפשר קריאה לרכיב זה
```

RFC
React Function Component
קומפוננטה אשר מחזירה פונקציות בלבד.
כל פונקציה תחזיר אלמנט אחד בלבד.

תרשים זרימה


index.htm

public:
Web view
App.j

<Xxx/>
<Yyy/>
<Zzz/>
(Components…)

ReactDOM.render
(רשומה ב
main.jsx)
src:
Main
Component
Xxx.jsx
Yyy.jsx
Zzz.jsx
Components

JSX
Java Script
XML

בריאקט ניתן לשלב בתוך הקוד קטעי HTML.

- כל אלמנט ללא תגית סוגרת ייסגר עם /
  זה ייראה כך: <element/>
- ביטויים ב JS ייכתבו בתוך { }.
- כל פונקציה יכולה להחזיר div אחד בלבד.
- המילה Class נקראת כאן ClassName.

Create new component

1. יצירת תיקייה לקומפוננטות (לא חובה אך מומלץ).
2. יצירת קומפוננטה (רכיב) עם סיומת jsx בתוך התיקייה.
3. יש לכתוב את הקוד הבא בקובץ הקומפוננטה:

```
import React from 'react'

export default function Person() {
     return (
             <div>
             קוד הקומפוננטה יירשם כאן
             </div>
   )
}
```

1. יש לייבא את הקומפוננטה הרצויה מ App:

```
import ComponentName from './Components/Person.jsx'
```

5. יש להציג את הקומפוננטה ב App (נרשום בתוך ה div של <Person />App)

**ניתן להשתמש בפקודה rfc ליצירת התבנית

Prop
s

שליחת ערכים לקומפוננטה

Props יודע לקבל ערכים (בהתאם למאפיינים) ולהחזיר אותם ל App.
בדוגמא זו הקומפוננטה Person יודעת לקבל שם וגיל ולהציג אותם דרך ה App.
**ניתן להציג קומפוננטה כמה פעמים שנרצה.

Function Component

```
import React from 'react'

export default function Person(props) {
     return (
             <div>
       <p> {props.name} </p>
       <p> {props.age} </p>
             </div>
   )
}
```

App

```
function App() {
  return (
    <div className="App">
         < Person name={'dor'} age={32} />
        </div>
  );
}
```

Prop
s

שליחת ערכים לקומפוננטה

ניצור מערך של אובייקטים כאשר כל אחד מהם יקבל name ו age
נשלח כל אובייקט במערך דרך App לפונקציית הקומפוננטה שיצרנו (Person).

```
JS App.js  ×  JS Post.js
appp > src > JS App.js > ...
1  import React from 'react'
2  import './App.css';
3  import Post from './Components/Post';
4
5
6  export default function App() {
7
8    const list = [{name:'dor', age:'33'}, {name:'max', age:'64'}, {name:'shem', age:'30'}];
9
10   return (
11     <div className="App">
12       <Post name={list[0].name} age={list[0].age} />
13       <Post name={list[1].name} age={list[1].age} />
14       <Post name={list[2].name} age={list[2].age} />
15     </div>
16   )
17 }
```

Add CSS

הוספת CSS לאלמנטים (קומפוננטה).

1. **<u>הוספת style לאלמנט ספציפי</u>**

```
element style={{ color:'red', fontSize:'50px' }}>
```

**שימו לב לשינוי של {{ }}, ו camelCase.

1. **<u>יצירת קובץ css.</u>**

הוספת הקובץ ע"י import './path';

Map

לא תמיד נדע מראש כמה פעמים נרצה להציג את הקומפוננטה.
למשל בפייסבוק, הפוסטים יוצגו בהתאם לגלילת הדף כלפי מטה.
נרצה לדאוג שעם כל הוספה/הסרה של פוסטים למערך הם יוצגו בהתאם באופן אוטומטי.
לשם כך נשתמש בלולאת Map **שרצה בהכרח על כל המערך** ויכולה להחזיר ערך.

במקום לרשום:

```
<Post name={list[0].name} info={list[0].info}/>
<Post name={list[1].name} info={list[1].info}/>
<Post name={list[2].name} info={list[2].info}/>
```

נרשום:

```
{list.map((element) => {
    return <Post name={element.name} info={element.info} />
})}
```

סוף שיעור 14

האם יש שאלות?