

React

React? What is it?

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

הסבר...

user



http://www.|

FrontEnd



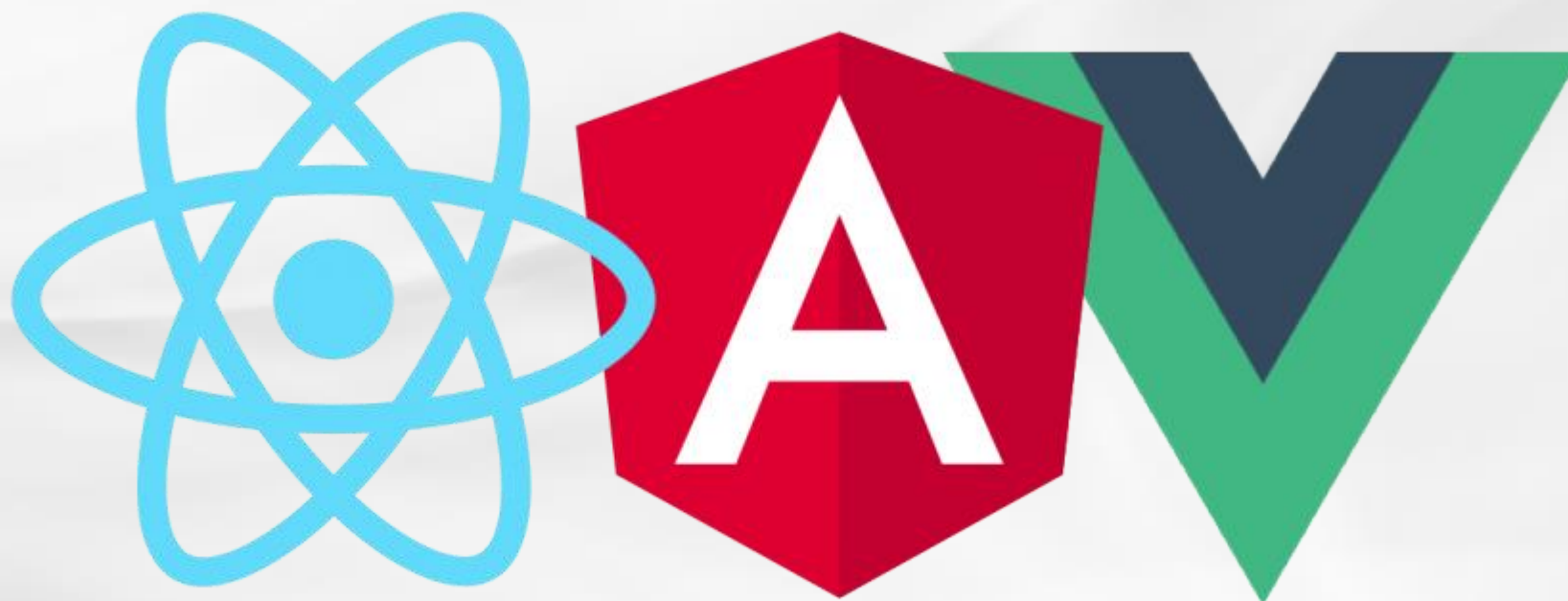
BackEnd



Server request



svcollege
ללמוד. לדעת. לעבוד.



svcollege
ללמוד. לדעת. לעבוד.

למה React?

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

• React Native

Install



svcollege
ללמוד. לדעת. לעבוד.

1. הורדת node.JS ← NPM (cmd node -v).
2. הורדת git (cmd git).
3. תוספת לדפדפן: React developer tools
4. תוספות לסביבת VSC :
 - ES7 React/Redux...
 - Prettier

`npm -g install create-react-app` ← (or Terminal) CMD

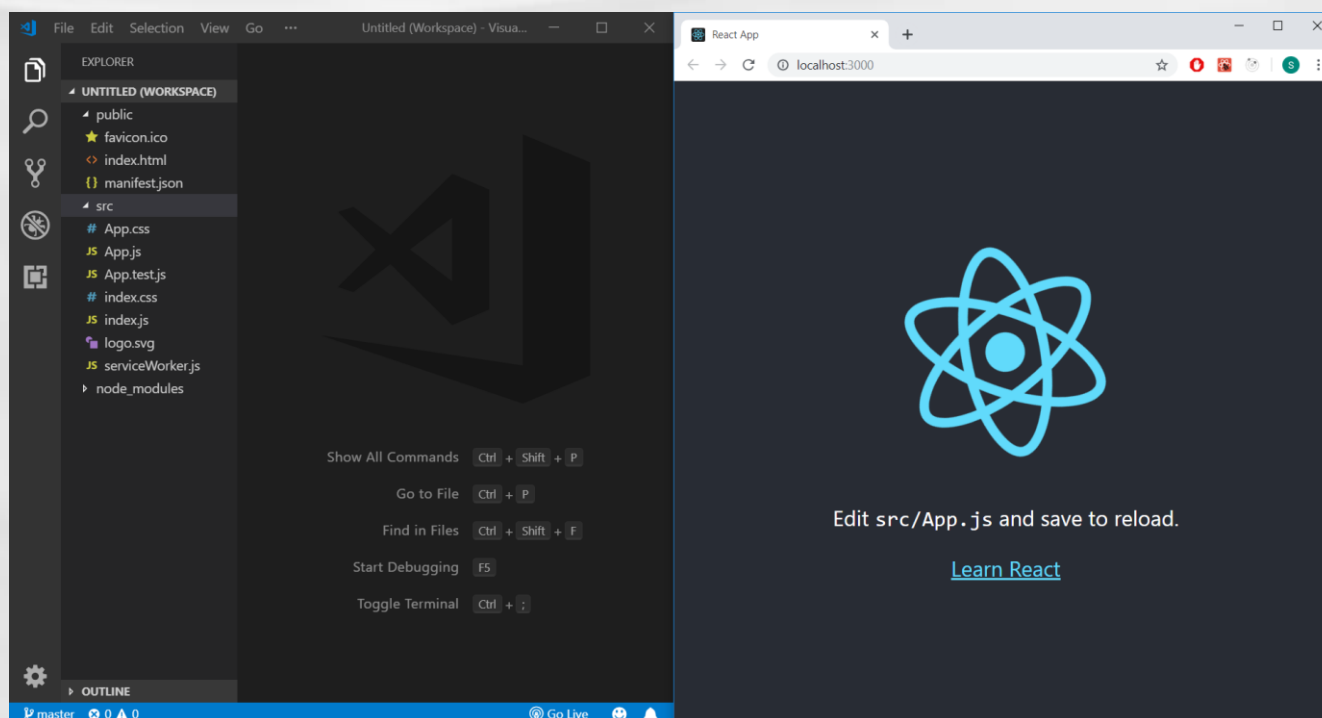
5. פותחים תיקייה חדשה בVSC
6. כניסה לTerminal (view)
 - `create-react-app appname` (שם אפליקציה באותיות קטנות בלבד)
 - `cd appname`
 - `npm start`

Create your first APP



svcollege
ללמוד. לדעת. לעבוד.

1. יצירת תיקייה ← להגיע אליה בעזרת CLI ← `npx create-react-app myapp`
2. פתיחת הקבצים בסביבת הפיתוח.
3. יצירת סביבת פיתוח ← `npm start`



FILES in React

1. **index.html**: מסמך html ראשי, דרכו מועבר הroot.
2. **Index.js**: מגדיר את הroot הראשי.
3. **App.js**: קומפוננטה הראשית של האפליקציה.
(אחראית לקריאת כל הקומפוננטות האחרות).
4. **App.css**: אחראית לעיצוב הקומפוננטה.

Component



svcollege
ללמוד. לדעת. לעבוד.

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



**Header
component**

**Body
component**

**Contact
component**

App.JS



svcollege
ללמוד. לדעת. לעבוד.

רכיב ראשי – אחראי לקריאה של רכיבים נוספים.

```
import React from 'react'; // הוספת ריאקט
```

```
import './App.css'; // הוספת עיצוב
```

```
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.

JSX

Java Script XML

בריאקט ניתן לשלב בתוך הקוד קטעי HTML.

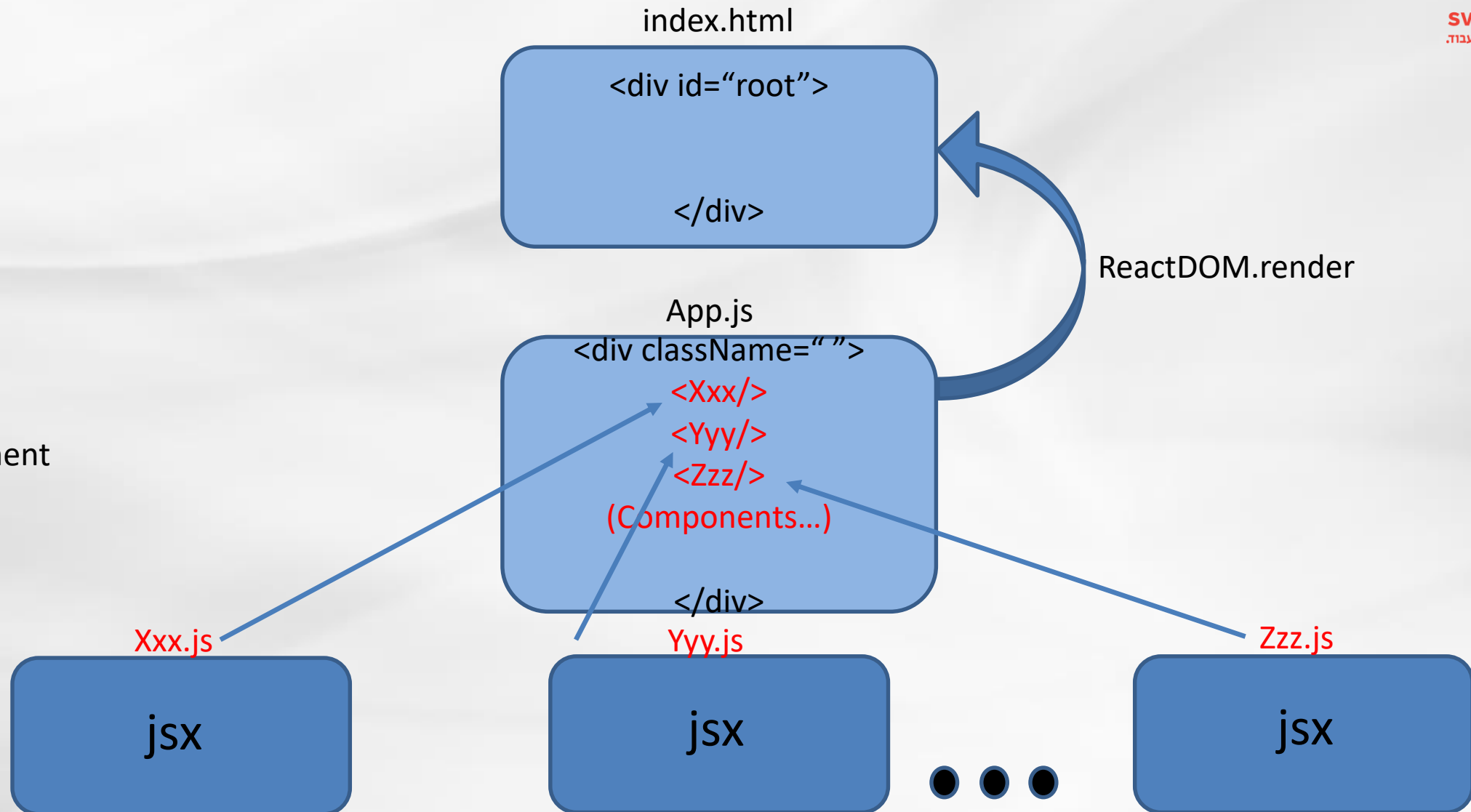
1. חייבים לסגור כל אלמנט ללא סוגר עם /
2. ניתן לכתוב ביטויים בתוך { .. } מתוך JS.
3. כל פונקציה יכולה להחזיר אלמנט אחד בלבד.
4. `ClassName ← Class`

תרשים זרימה

public:
Web view

Src:
Main Component

Components



Create new component



svcollege
ללמוד. לדעת. לעבוד.

1. יצירת תיקייה לקומפוננט (לא חובה).
2. יצירת קובץ js בתוך התיקייה עם שם הרכיב.
3. יש לכתוב את הקוד הבא:

```
import React from 'react'
```

```
export default function Header() {  
  return (  
    <div>  
  
    </div>  
  
  )  
}
```



<Header />

Props

שליחת ערכים לקומפוננטה

1. יצירת מאפיין מאפיין בקומפוננטה שנשלח ל App וחוזר עם הערך הרצוי.

```
import React from 'react'
```

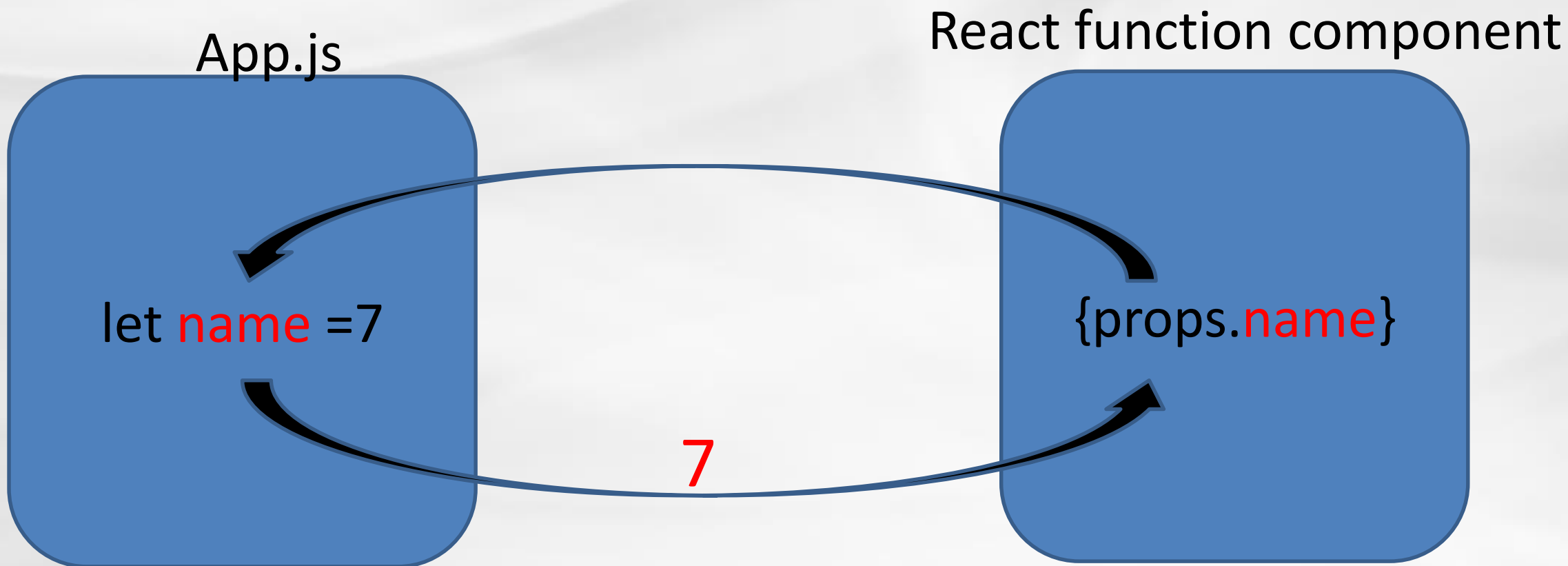
```
export default function Header(props) {  
  return (  
    <div>  
      <label> {props.name} </label>  
    </div>  
  )  
}
```

2. קבלת הערך בתוך הפונקציה.

Props



svcollege
ללמוד. לדעת. לעבוד.



Add CSS

הוספת CSS לכל רכיב (קומפוננטה).

1. הוספת style לאלמנט ספציפי -

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

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

2. יצירת קובץ css. הוספת הקובץ ע"י import './path';

3. הוספת BOOTSTRAP

א. ב Terminal : npm install bootstrap

ב. import 'bootstrap/dist/css/bootstrap.min.css'

Style JSX

```
<element style={{attribute:'value' , attribute:'value'}}>
```



Bootstrap

1. CLI → npm install bootstrap
2. App.js → import 'bootstrap/dist/css/bootstrap.min.css';

CLASS-OOP

מחלקה - קומפוננטה יעודית לריאקט, מהווה תבנית ליצירת
קומפוננטה מורכבת שתוכל להכיל בתוכה פונקציות, משתנים
גלובאליים, בדיקות (Validation) ועוד...

```
import React, { Component } from 'react'
```

```
export default class Body extends Component {  
  render() {  
    return (  
      <div>  
  
      </div>  
    )  
  }  
}
```

* ייבוא המחלקה יעשה בצורה זהה לייבוא של קומפוננטת פונקציה.

CLASS-OOP

לקומפוננטות מחלקה יש **אובייקט state** מובנה הנקרא **state** בתוכו ניתן לאחסן את כל המשתנים הגלובליים של המחלקה.

```
state={  
  name: '',  
  car: '',  
  list: []  
}
```

כיצד מגדירים:

כל state ניתן לעדכן ע"י שימוש בפונקציה מובנית הנקראת `this.setState`

- לכל מחלקה ניתן לייצר state אחד בלבד.
- כל שימוש ב `setState` יפעיל את `render`.

CLASS-OOP

מחלקה הינה עצם שניתן לייצר ולאפיין אותו כרצוננו.

כיצד ניצור מחלקה -

1. ניצור קובץ js.

2. נגדיר מחלקה `export class NameOfClass{..}`

3. ניצור בנאי שיאתחל את המחלקה

4. נגדיר למחלקה מאפיינים

בנאי - constructor

בנאי – מתודה אשר אחראית להעברת ערכים לקומפוננטה מסוג מחלקה (class).

- מתבצע בעזרת הפונקציה `super()`
 - במידה ונגדיר state הוא ירשם בתוך הבנאי.
- ```
constructor(props) {
 super(props)

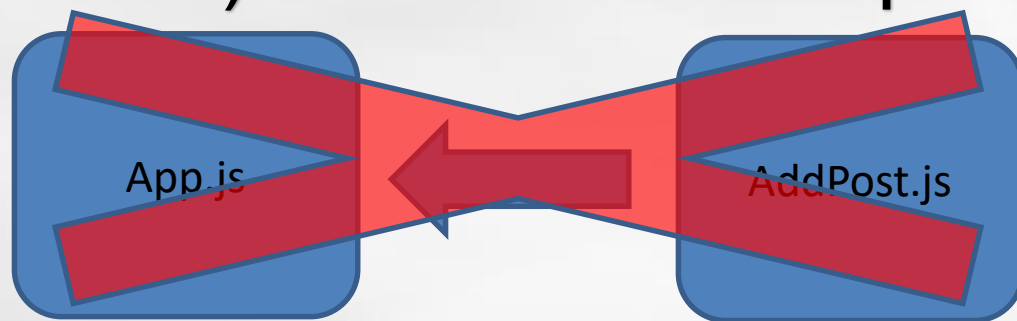
 this.state = {
 name: '',
 car: '',
 list=[]
 }
}
```



# CLASS-OOP

**המטרה:** יצירת פוסט חדש בקומפוננטת AddPost , כאשר נלחץ על כפתור ההוספה הפוסט החדש יתווסף למערך הפוסטים בApp (מהשרת) ויוצג.

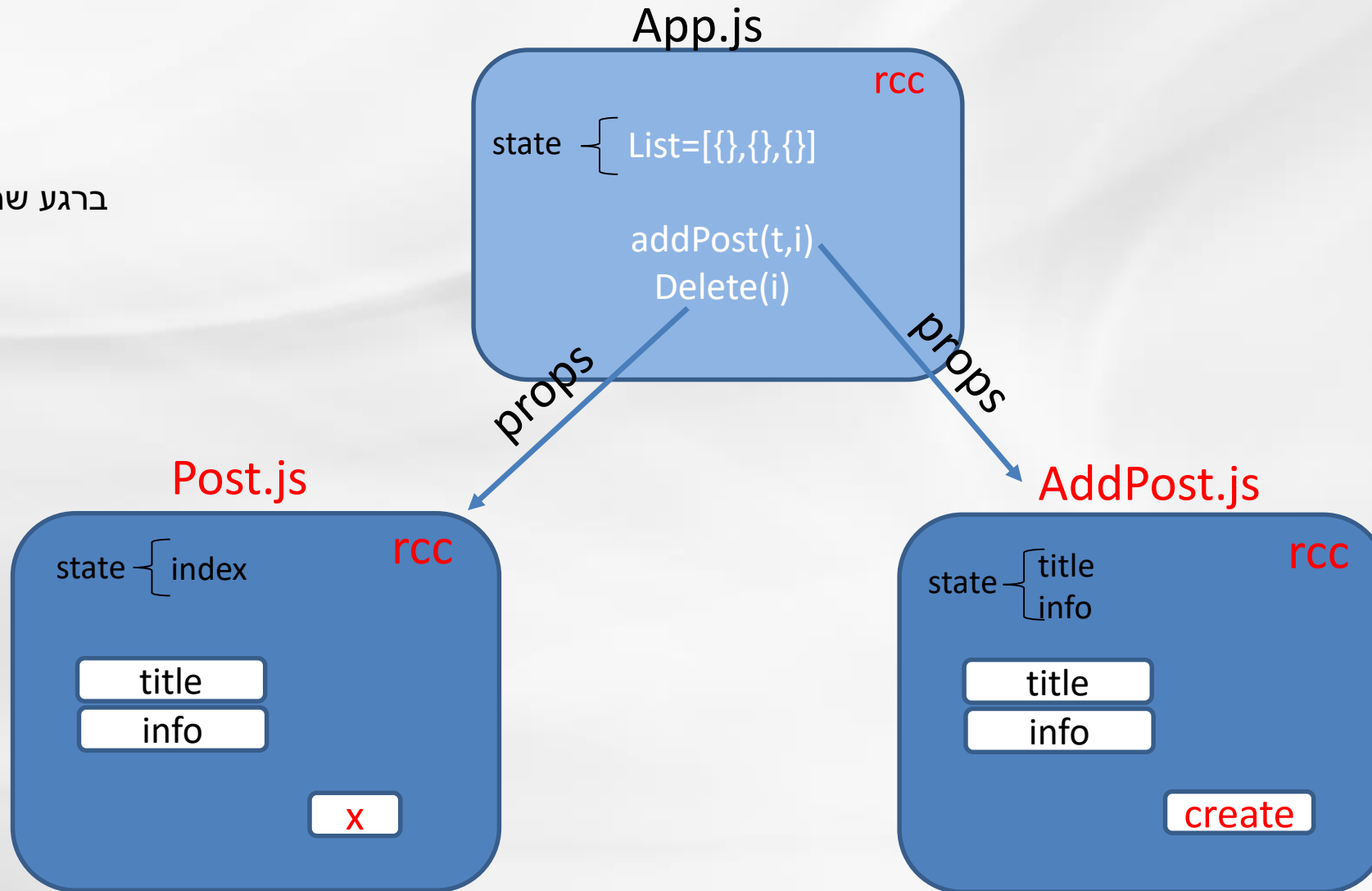
**הבעיה:** לא ניתן להעביר ערכים לקומפוננטה הראשית (לא לבלבל עם props)



**הפתרון:** יצירת פונקציה בקומפוננטה הראשית שתשלח לקומפוננטה AddPost בעזרת props.

# תרשים זרימה

state:  
ברגע שמופעל מבצע רינדור



# hooks

Hooks הינו תוספת חדשה (החל מגרסה 16.8 של react) המאפשר שימוש בstate ובפיצ'רים נוספים של מחלקות, ללא שימוש בקומפוננטה שהיא מחלקה.

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

ייבוא נתונים נוספים מה package :  
אלו נתונים לא דיפולטיביים.

הגדרת ה hooks:

```
const [varname, setVarname]=useState(flse)
```

state

הפונקציה שאחראית לעדכון hooks

אתחול hooks בערך מסויים



# Function component VS class component

| RFC                    |                             | RCC                               |
|------------------------|-----------------------------|-----------------------------------|
| Simple                 | מתי נשתמש?                  | Complex                           |
| props<br>{props.name}  | העברת ערכים לקומפוננטה      | constrauctor<br>{this.props.name} |
| Hooks<br>{useState}    | משתנים גלובאליים            | State<br>{component}              |
| [varname , setVarname] | עדכון ה value ב state/hooks | setState                          |
| let,var,const          | הגדרת משתנים                | -----                             |
| -----                  | קריאה לפונקציות             | this.                             |

# Routers

Routers מאפשר לנו לשנות את כתובת ה URL בהתאם לעמוד שבוא אנחנו נמצאים.

`http://www.yourWebsite.com/???`

הוספת module למערכת שלנו (ב- CLI):

```
npm install react-router-dom
```

App.js

```
import {BrowserRouter as Router , Route , Switch} from 'react-router-dom';
```

# Routers

מנגנון אשר מבצע את העברת המידע בין הערוצים באתר

`<Router/>` - כל הרכיבים הקשורים להחלפת הערוץ באופן ישיר

או עקיף

`<Switch/>` - כל הרכיבים שאמורים להתחלף בפועל.

`<Route/>` - הגדרת הערוץ עבור רכיב מסוים.

```
<Router>
```

```
 <switch>
```

```
 <Route exact path="/" component={main}/>
```

```
 <Route exact path="/about" component={about}/>
```

```
 <Route component={pageNotFound}/>
```

```
 </switch>
```

```
</Router>
```

# Routers

תכנון אתרים (נפוץ):



svcollege  
ללמוד. לדעת. לעבוד.

Web page

Component 1 - header

LOGO

details

contact

Component 2- navigation

Btn -1

Btn - 2

Btn - 3

Btn -4

Btn - 5

Content  
1/2/3/4/5

לא קשור לשינוי הערוצים

קשור לשינוי הערוצים  
אבל לא מתחלף

קשור לשינוי הערוצים  
וגם משתנה בעצמו

# Routers-Link



svcollege  
ללמוד. לדעת. לעבוד.

- קישור המאפשר לנו לשנות יעד בתוך האתר.
- ביצוע ניווט בלחיצה על אלמנט מסוים שעטוף בתגית Link יעביר אותנו לערוץ הרצוי.

בקומפוננטה שבה נרצה להשתמש ב:Link

```
import {Link} from 'react-router-dom';
```

```
<Link to="/"> HOME </Link>
```



# Routers-Link

- קישור המאפשר לנו לשנות יעד בתוך האתר.
- ביצוע ניווט בלחיצה על אלמנט מסוים שעטוף בתגית Link יעביר אותנו לערוץ הרצוי.

בקומפוננטה שבה נרצה להשתמש ב:Link

```
import {Link} from 'react-router-dom';
```

```
<Link to="/"> HOME </Link>
```

