WHAT IS REACT?

React is a javascript library for building user interface

https://www.airbnb.com/

We create smaller custom html elements with styles And put them inside the big structure

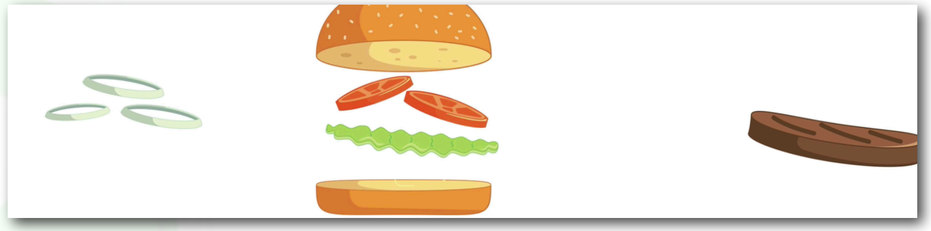
This helps to keep the code clean and reusable components

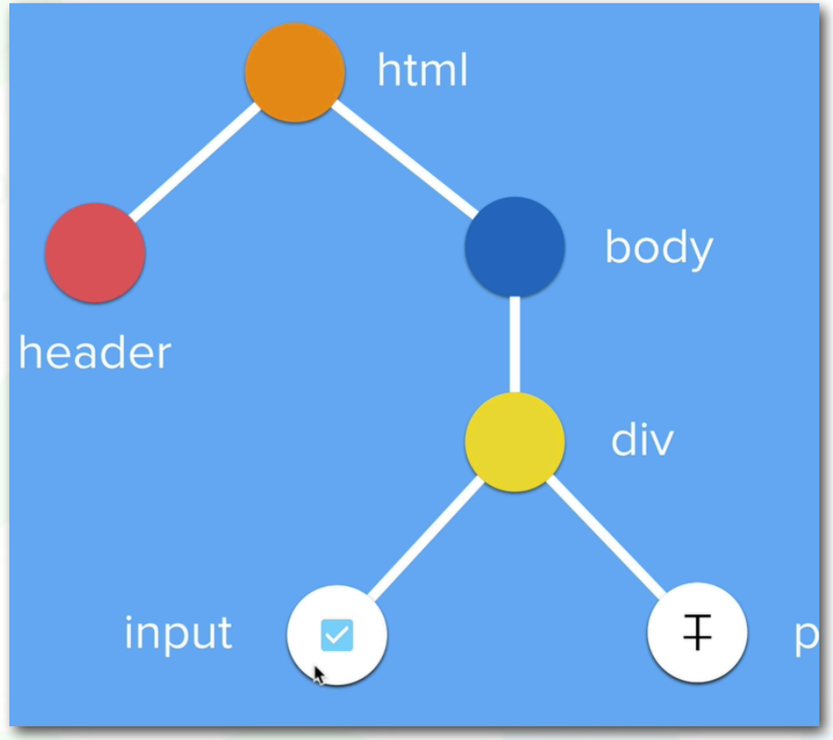
We can think this component as an ingredient inside a burger

We can add small components inside the app like an ingredients

To create a single component, we will use a mix of html, css, and javascript

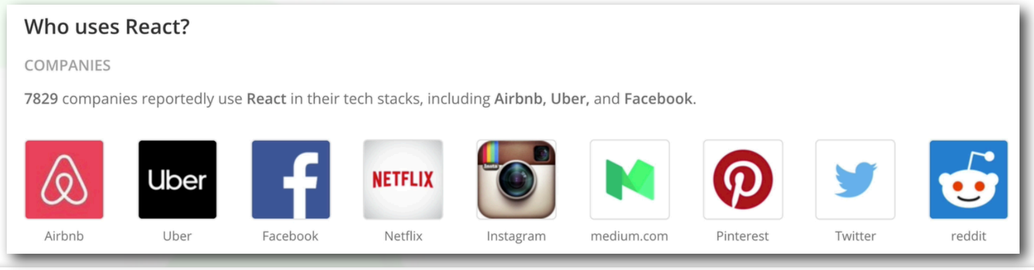
React only renderes/updates the changing components.

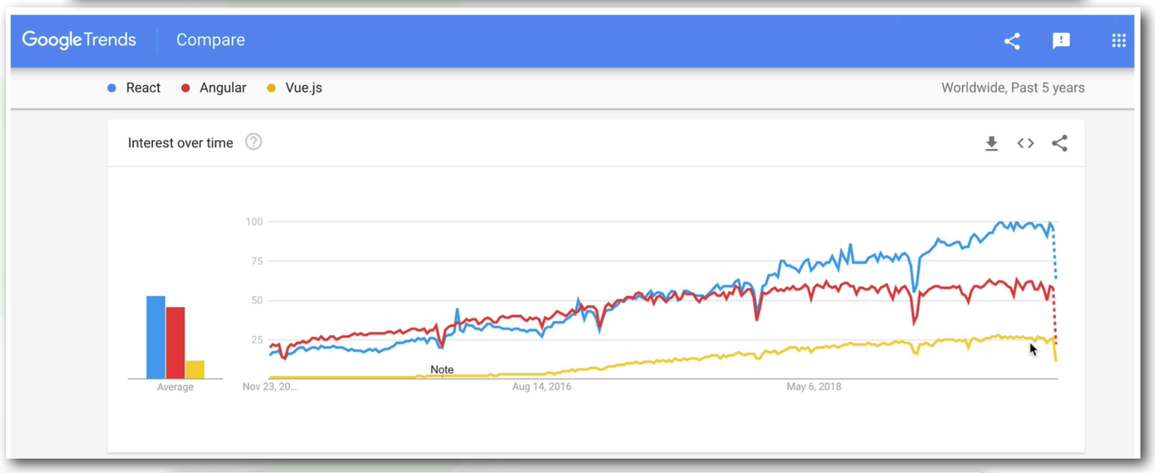


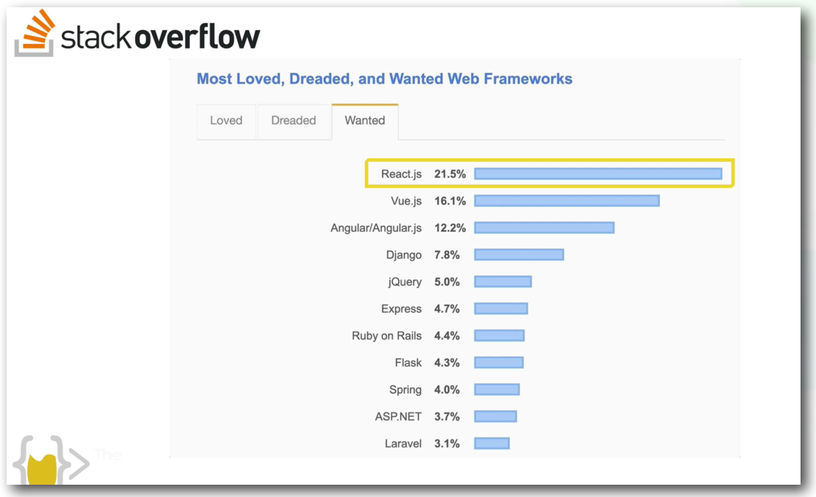


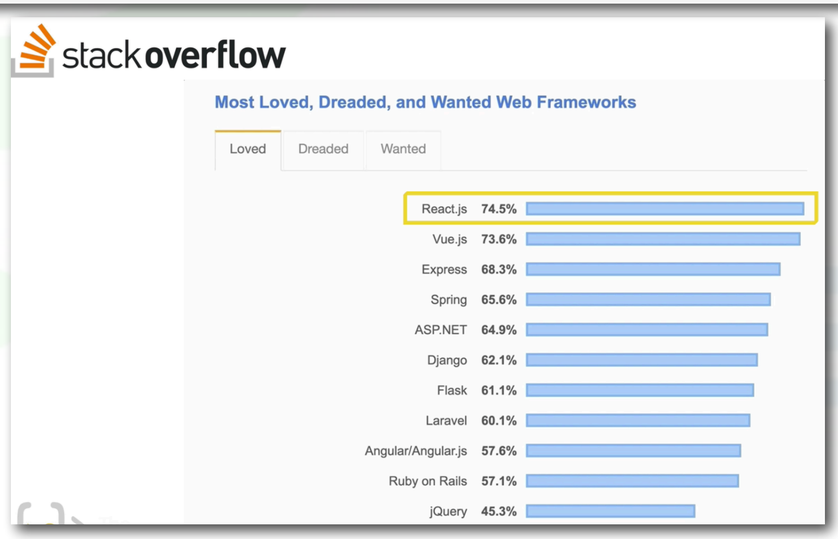
WHY REACT?

- Simplicity
- Predictable
- Flexibility
- Performance
- Testability
- Community support



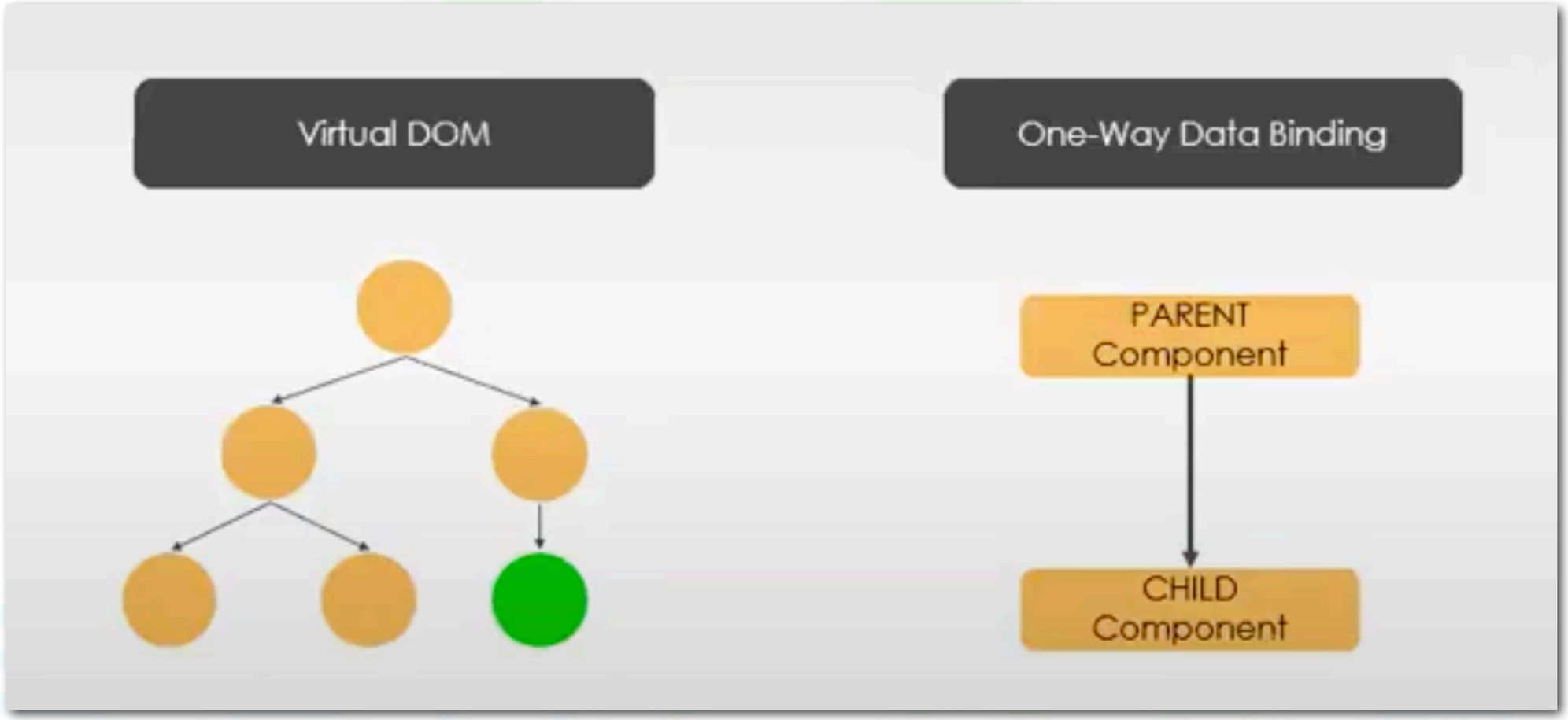






WHY REACH IS FAST

VIRTUAL DOM
ONE-WAY DATA BINDING



WHAT IS VIRTUAL DOM?

Reach DOM-Virtual DOM- Copy of actual DOM

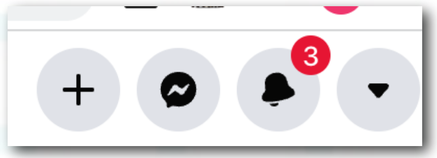
React let us update the page without changing the actual DOM.
(no more document.getElementById("id")).innerHTML='Hello')

React create a virtual copy of actual DOM. Every time there is a chance in the in the virtual DOM, then really update only that changing part. This makes app faster.

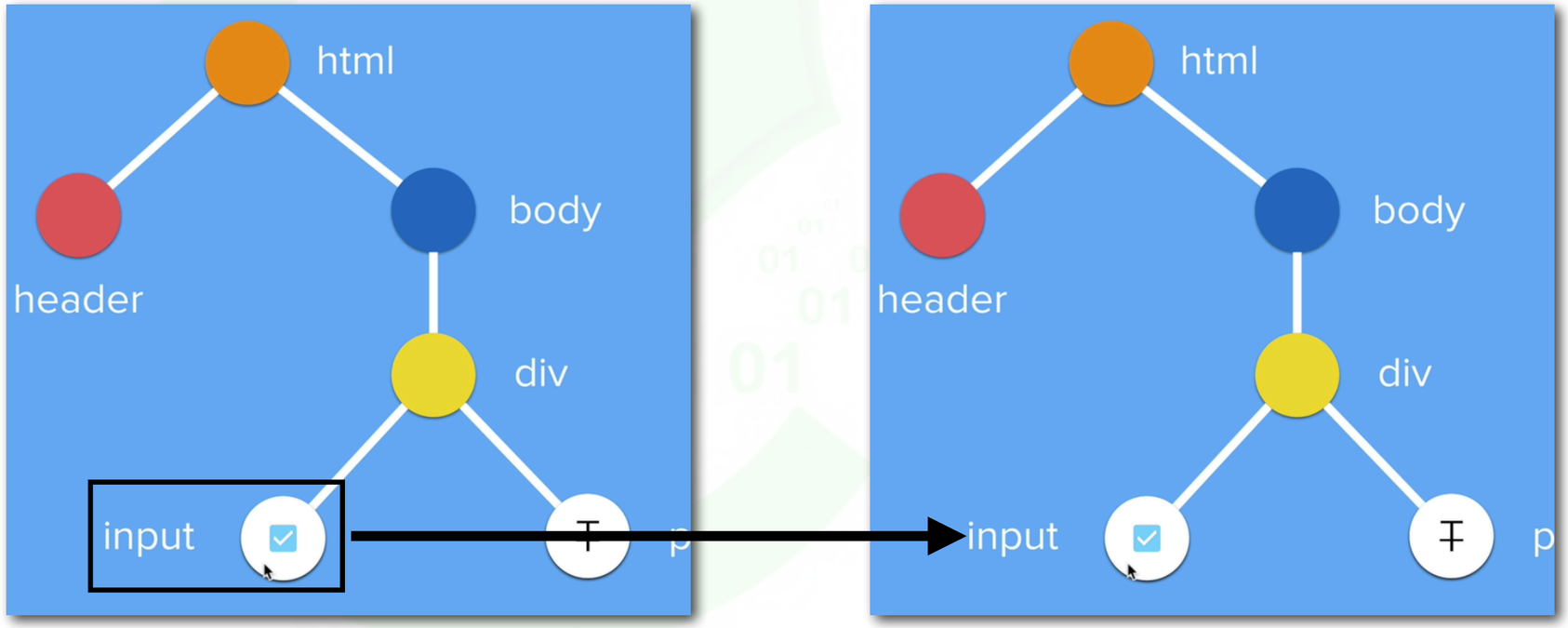
React can detect single component changes and render ONLY the changes, without rendering the entire application. This makes the application faster.

In this DOM tree, react will only render the input if that is the only changing element

For this reason, react help us to handle even small part of the application effectively.



Actual DOM



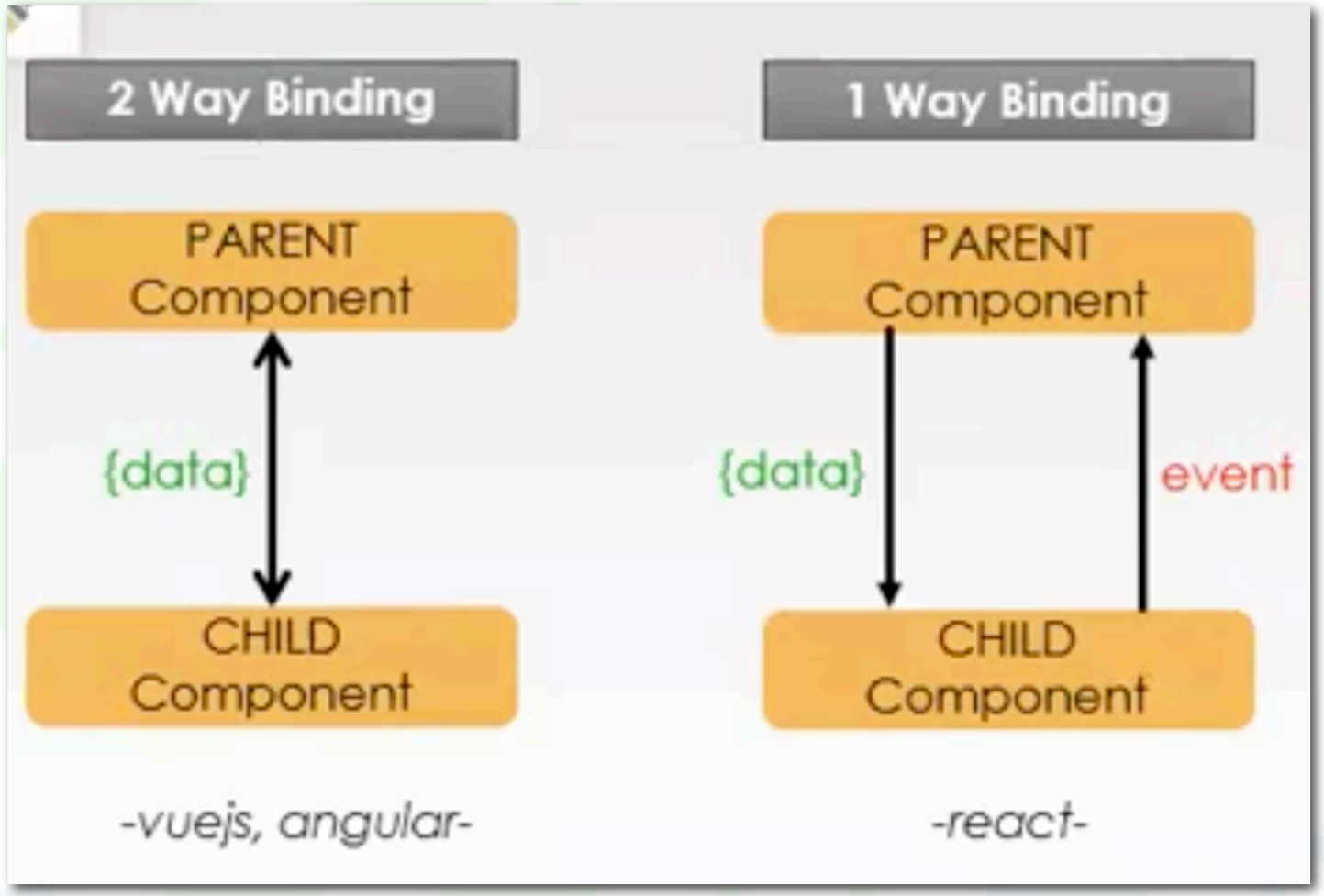
When there is a change in input
Then react will update only this specific element in the actual DOM
This makes app work faster
https://www.youtube.com/watch?v=BYbgopx44vo

ONE WAY DATA BINDING

IN 2 WAY DATA BIDING, WHEN STATE OF A PARENT CHANGES, THEN STATE OF A CHILD CHANGES VICE VERSA.

IN 1 WAY DATA BIDING, WHEN STATE OF A PARENT CHANGES, THEN STATE OF A CHILD. BUT WHEN STATE OF CHILD CHANGES, ADDITIONAL EVENT MUST BE USED TO REFLECT THE SAME CHANGES IN THE PARENT

ONE WAY DATA BINDING IS A LITTLE FASTER



REACT IS COMPONENT BASED

REACTS IS BASED ON REUSABLE COMPONENTS

COMPONENTS ARE REUSABLE

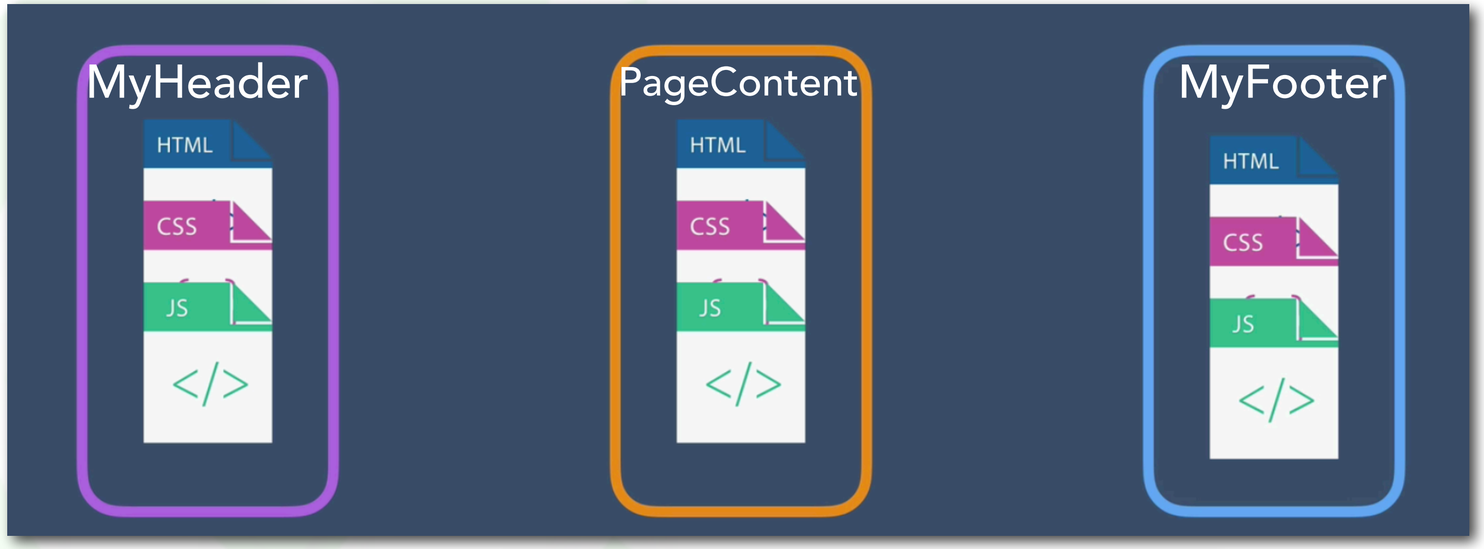
COMPONENTS CAN WORK INDEPENDENTLY

ONE COMPONENT CAN BE USED IN OTHER COMPONENTS

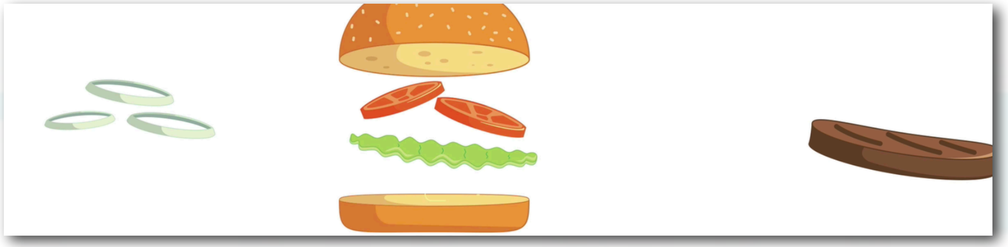
DATA CAN BE CALLED BETWEEN THE COMPONENTS

A COMPONENT CAN HAVE ITS OWN STATE
THIS STATE CAN HOLD THE DATA AND STYLE.
STATES CAN BE USED IN OTHER COMPONENTS

COMPONENTS ARE LIKE INGREDIENTS OF A SANDWICH

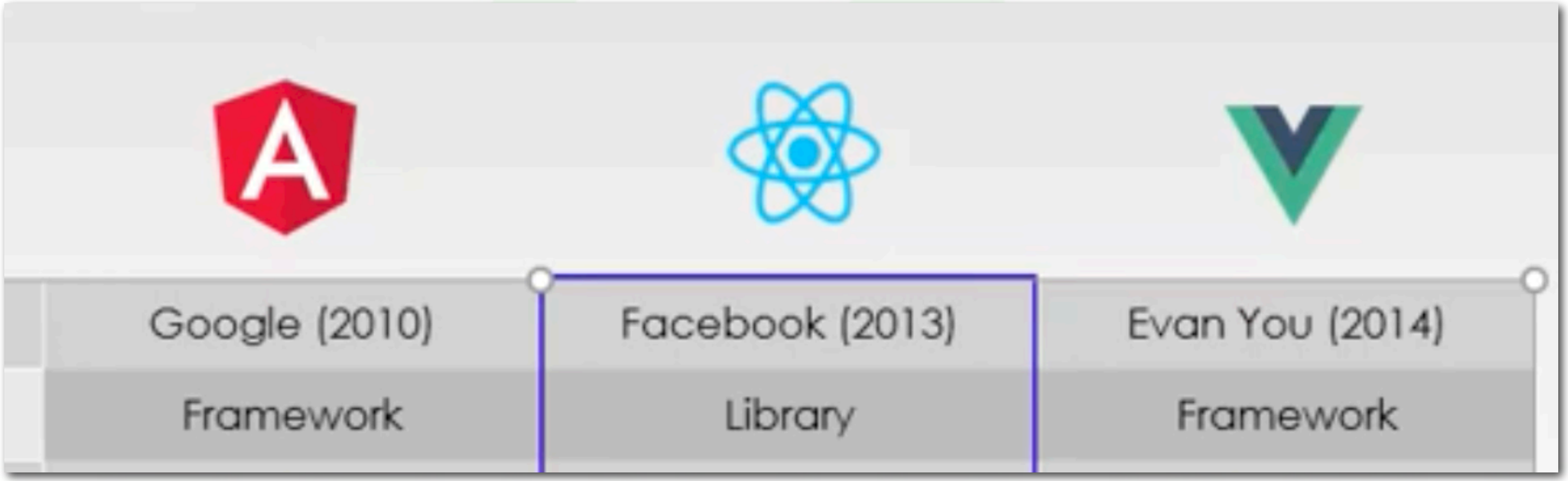



```
<body>
<MyHeader />
<PageContent />
<MyFooter />
</body>
```



REACT ALTERNATIVES

3 POPULAR JAVASCRIPT FRAMEWORK OR LIBRARY



USEFUL LINK

https://reactjs.org/docs/getting-started.html

https://codesandbox.io/s/new?file=/src/App.js

1. LOCAL ENVIRONMENT SETUP & CREATE FIRST REACT APP

1. Check node, npm. npm —version AND node —version

2. Create a react app using package manager: https://reactjs.org/docs/create-a-new-react-app.html

> Create a folder on your react-classes folder: my-first-react-app

> Open your VS Code terminal and run below command to create a react project

1. Check Node is Up to Date
2. Install VSCode
3. Create React App
4. Run App

npx create-react-app app-name => This will install react related packages

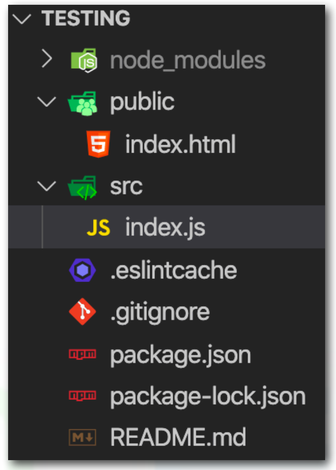
npx create-react-app 01-my-first-app

3. cd app-name ->>>> cd 01-my-first-app

```
cd testing
npm start
```

```
Compiled successfully!
```

4. **npm start** => Now you should see the browser opened react



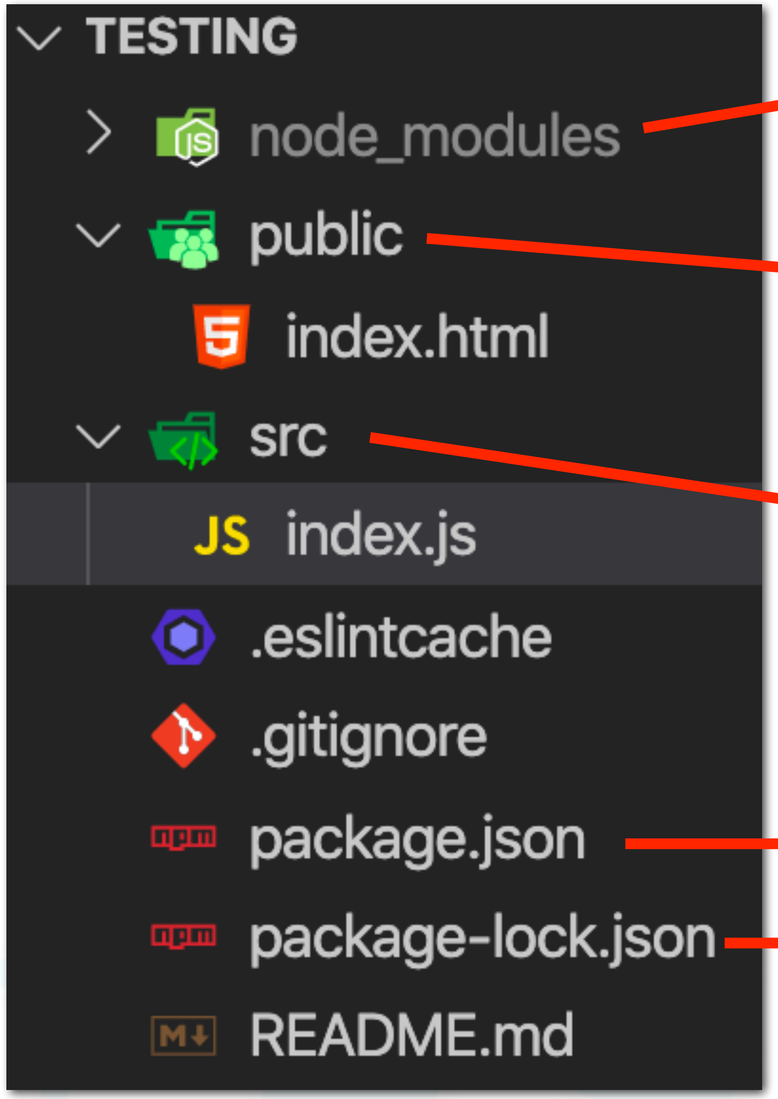
6. Delete the unnececery files. Go to public and delete everything **but index.html**

7. Go to src delete everything **but index.js**

8. Go to **index.html,** delete the unnecessary parts , and **add <script>**

9. Go to **index.js, delete unnecessary files,** write the code to display Hello World.

PROJECT STRUCTURE



Contains all react modules and libraries.
npm install creates this folder
Size is too big so we should not push to git when you share

Public folder has static files, such as index.html, images, data files, etc.
We don't write code in public folder

Src folder has resource files.
index.js, and other component will be created here
This will has the react codes

Project information is stored here.

Project information is stored here.
This is created after npm install

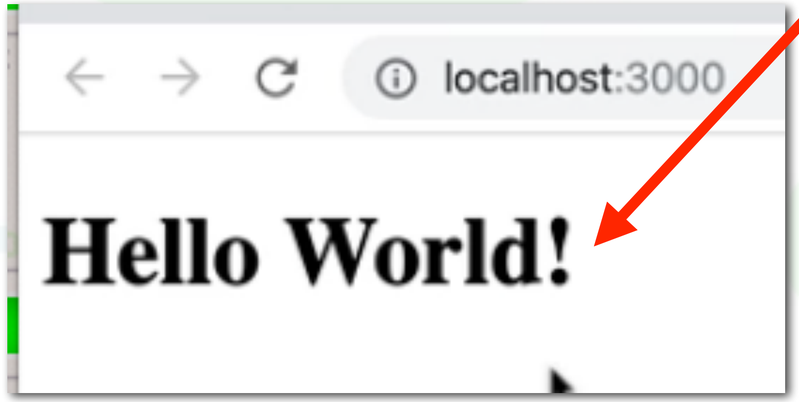
HELLO WORLD

index.html

```
public > index.html > html > body > script
1   <!DOCTYPE html>
2   <html lang="en">
3     <head>
4       <title>React App</title>
5     </head>
6     <body>
7       <div id="root"></div>
8       <script src="./src/index.js" type="text/jsx"></script>
9     </body>
10  </html>
```

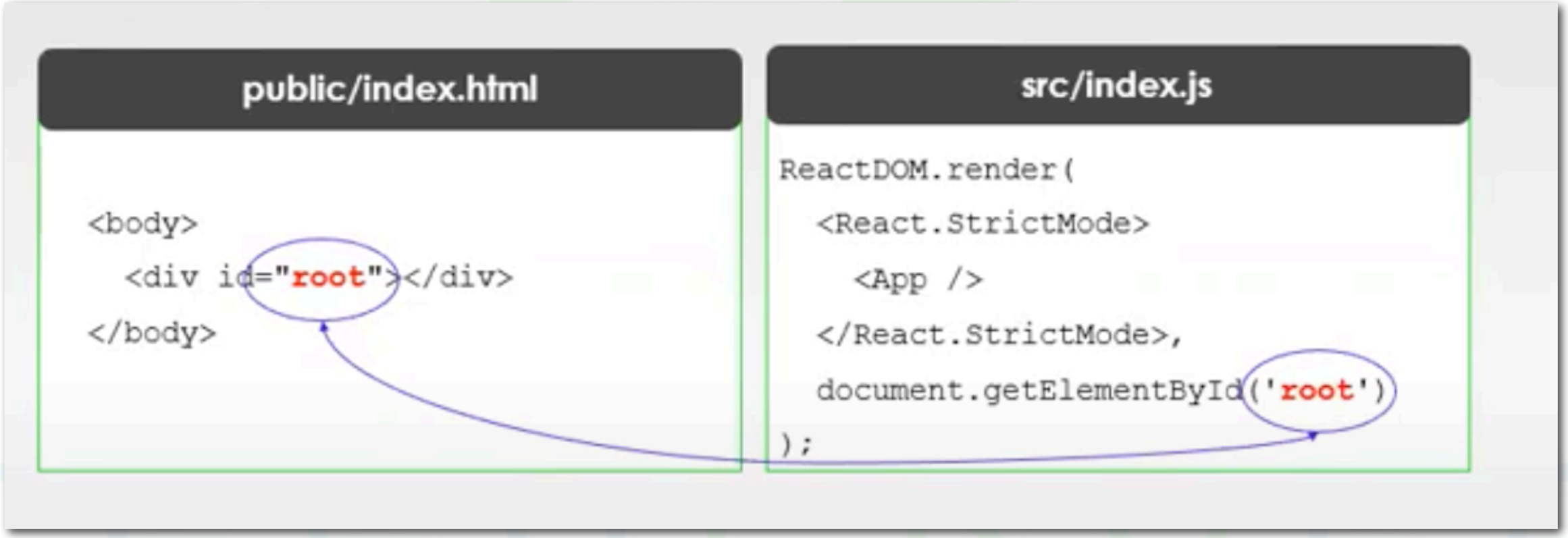
index.js

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3
4
5   ReactDOM.render(
6     <div>
7       <h1>Hello Wold!</h1>
8     </div>,
9     document.getElementById('root')
10  );
```



RENDERING

REACTDOM.RENDER GETS THE CODE FROM INDEX.JS AND RENDERS IN THE BODY OF HTML



CREATING A NEW PROJECTS WHEN YOU HAVE EXITING REACT APP

We can create a template for new react apps

That template will not have node_module cause it is too big

To install : npm install

to run : npm start

Steps:

1. Copy the previous code in the my-react-project folder

2. Delete node-module folder

3. Change the name of the folder : template

3. Now that I created a template, I can Copy the template one more time and change name : 02-intro-to-react

4. Open the terminal in vs code and change the directory to this folder

5. install : npm install

6. run : npm start

REACT STARTING FILE

Download the starting folder

Unzip

Open in vs code

In terminal run these scripts:

npm install

npm start

ERROR Then RUN: npm install react-scripts start

RUN npm start

Then you should see a blank page

TRADITIONAL HELLO WORLD RENDERING

index.js > ...

```
const ReactDOM = require('react-dom')

// Show Hello World on as h1 in the website.
//TRADITIONAL JS EVELOPMENT
var h1=document.createElement('h1');
h1.innerHTML="Hello World !!!!"
document.getElementById("root").appendChild(h1);
```

Hello World !!!!

```
7
8     <body>
9       <div id="root"></div>
10      <script src="../src/inde
```

WITHOUT REACT AND JSX

index.js U

src > index.js > ...

```
// *************Createing an h1 and displaying on the
//1. creating an h1 element
var heading = document.createElement('h1');
// 2. assign a text value
heading.innerHTML='Hello World!';
// 3. show this on the UI
document.getElementById('root')
.appendChild(heading);
//Then you will see the Hello world on the UI
// *******Creating a paragrapgh and displaying on the
var parag=document.createElement('p');
parag.innerHTML='<i>Hello World!</i>';
document.getElementById('root')
.appendChild(parag)
// *******Creating an input and diplaying on the UI**
var passwordInput = document.createElement('input');
document.getElementById('root')
.appendChild(passwordInput);
//********Giving Line Break */
var lineBreak = document.createElement('hr');
document.getElementById('root')
.appendChild(lineBreak);
//******Creating a button and displaying on the UI***
var submitButton = document.createElement('button');
submitButton.innerHTML='Submit!!!'
document.getElementById('root')
.appendChild(submitButton)
```

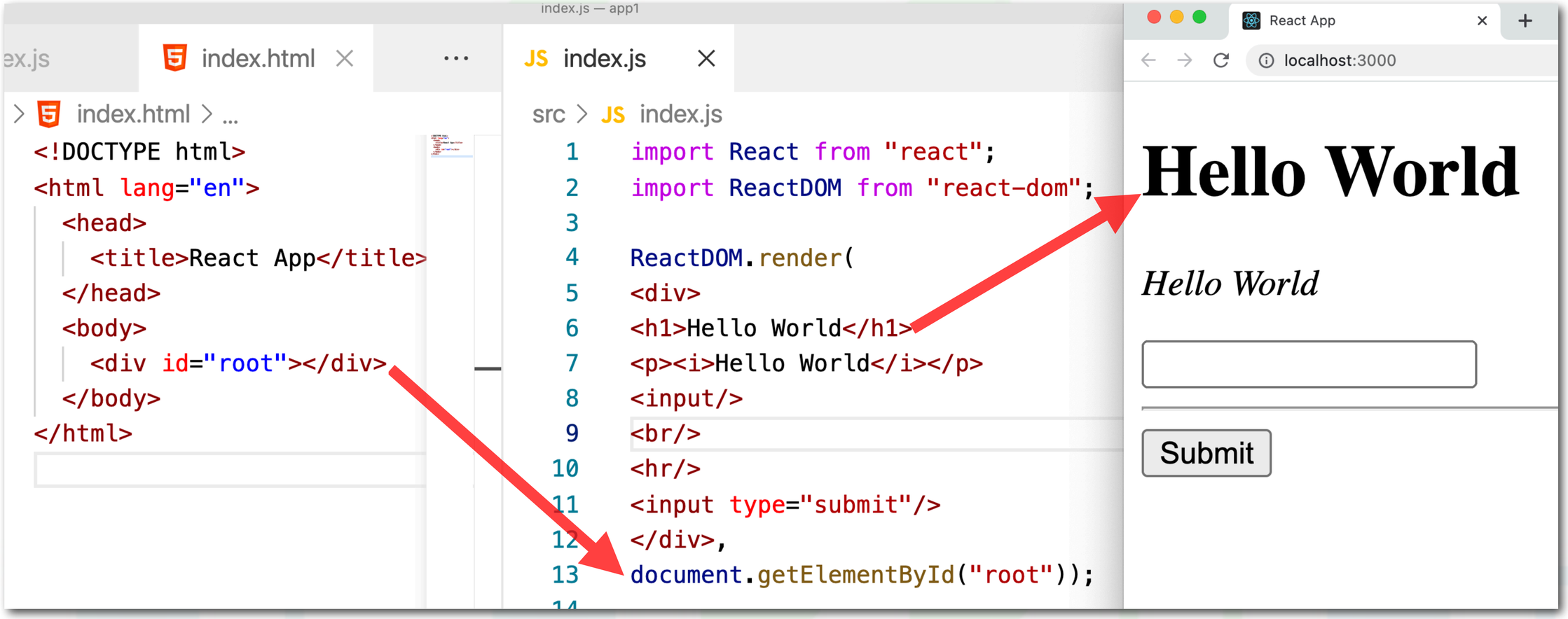
index.html U

public > index.html > html > body

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>React App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

WITH REACT AND JSX

WE CAN WRITE JSX(HTML) IN JS FILES



JSX

We will not touch index.html. All our code will be written using js in index.js

It has some html code, and

All react apps have an id=root by convention.Everything in our app will go into this div

TASK:

Show Hello World on as h1 in the website.

We are in a js file but we are able to write html code.How this is possible?

React works by creating JSX FILES

Html is picked up and compiled to javascript using react modules.

IF YOU DELETE react("react"), then you get compiler issue

WE SHOW h1 IN in index.html so use document.getElementById("root"))

```
ReactDOM.render(<h1>Hello World</h1>, document.getElementById("root"));//root in the index.html
Render method is used to display elements in the root div
```

index.html

```
<body>
  <!-- <div id="root"></div>=> THIS IS WHERE WE WILL SHOW OUR CODE-
  <div id="root"></div>
  <script src="../src/index.js" type="text/javascript"></script>
  <!-- ../src/index.js => PATH OF index.js FILE
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
```

```
ReactDOM.render("WHAT TO SHOW , WHERE TO SHOW");
```

```
ReactDOM.render(<h1>Hello World</h1> , document.getElementById("root"));
```

Use ReachDOM.render module to render and display h1 html element

JSX VS VANILLA JS

Which one is shorter?

Rendering using react module

```
ReactDOM.render(<h1>Hello World!</h1>, document.getElementById("root"));
```

Rendering using vanilla javascript

```
var h1 = document.createElement("h1");
h1.innerHTML = "Hello World!";
document.getElementById("root").appendChild(h1);
```

```
1 ReactDOM.render(<h1>Hello World</h1>,
  document.getElementById("root"));
```

```
1 "use strict";
2
3 ReactDOM.render(
  /*#__PURE__*/React.createElement("h1", null,
  "Hello World"),
  document.getElementById("root"));
```

JSX

How can we render multiple elements on a page?

How do add a paragraph on the page write under the heading like. **Below code will crash**

```
ReactDOM.render(
<h1>Hello World</h1>
<p >This is the paragraph</p>,
 document.getElementById(“root"));
```

It is because render method takes **ONLY ONE SINGLE HTML ELEMENT.** So we need to turn multiple html elements into one. HOW?? So we can embed this two html element into one single **div** element just like so

Below we have one div element. The div element has two Childs <h1>, and <p>

```
ReactDOM.render(
   <div>
       <h1>Hello World</h1>
       <p>This is the paragraph</p>
   </div>
,
 document.getElementById("root"));
```

```
// var React = require("react");
// var ReactDOM=require("react-dom");
import React from "react";
import ReactDOM from "react-dom";
ReactDOM.render(
<div>
<h1>Hello World</h1>
<p>This is paragraph</p>
</div>,
document.getElementById("root"));
```

REVIEW

- What is react?
  - React is a JS library to build front end applications
  - React allows you to create re-usable and reactive components consisting of HTML and JavaScript (and CSS)
- React is component based application
- What component?
  - Reusable smaller part of the application. It is like an ingredient of a food

    Heading component

    Information component

    Images component …
- What is Virtual DOM?
  - Copy of the actual DOM. When react detects any change, it will update the actual DOM only that part of change. For example, clock is ticking(hh mm ss(ss will change every second)),Every second, only this element will be refreshed and updated, Not the entire page
  - When we make a change in one of the component it doesn't refresh the whole page, it just updates that specific component
  - react updates a component of app by injecting the necessary part without refreshing all page, what makes react project fast
- What is JSX?
  - JSX is an inline markup that looks like HTML and gets transformed to JavaScript. We use JSX to create React elements/component.

BABEL

Babel is a Javascript Compiler:

It takes next generation javascript and compile it down to javascript so every browser even old browser can understand the code

https://babeljs.io/en/repl

Copy react code and paste, then you should see javascript version

JSX CODE TASK

Complete the task:

Create a new Reach app from scratch:

Name: 03-intro-to-JSX

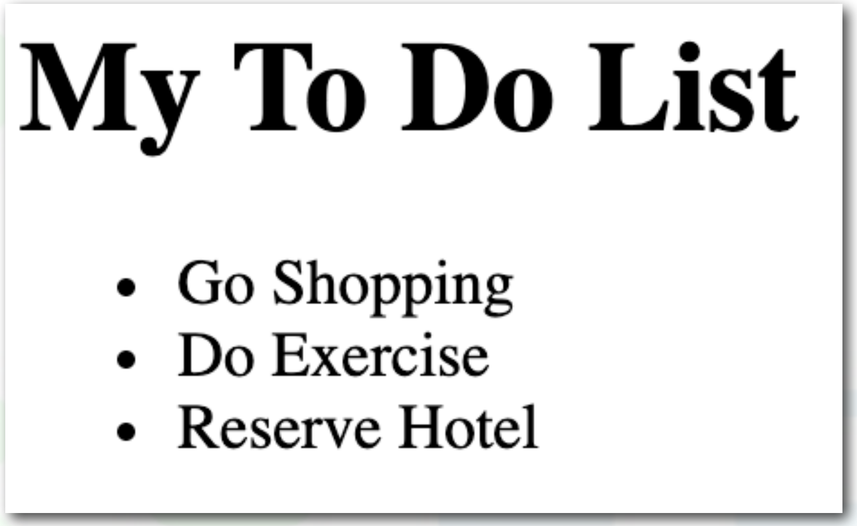
Note that you can use template(RECOMMENDED) or create a new react app from scratch

//It should display a h1 heading.

//It should display an unordered list (bullet points).

//It should contain 3 list elements.

Steps:
1.Duplicate the template and change name : 03-intro-to-JSX
2. Open in CS code and run the commands
npm install
npm start
3. Write your code in index.js



JSX CODE TASK SOLUTION