

Lesson 9

REACT INTRODUCTION

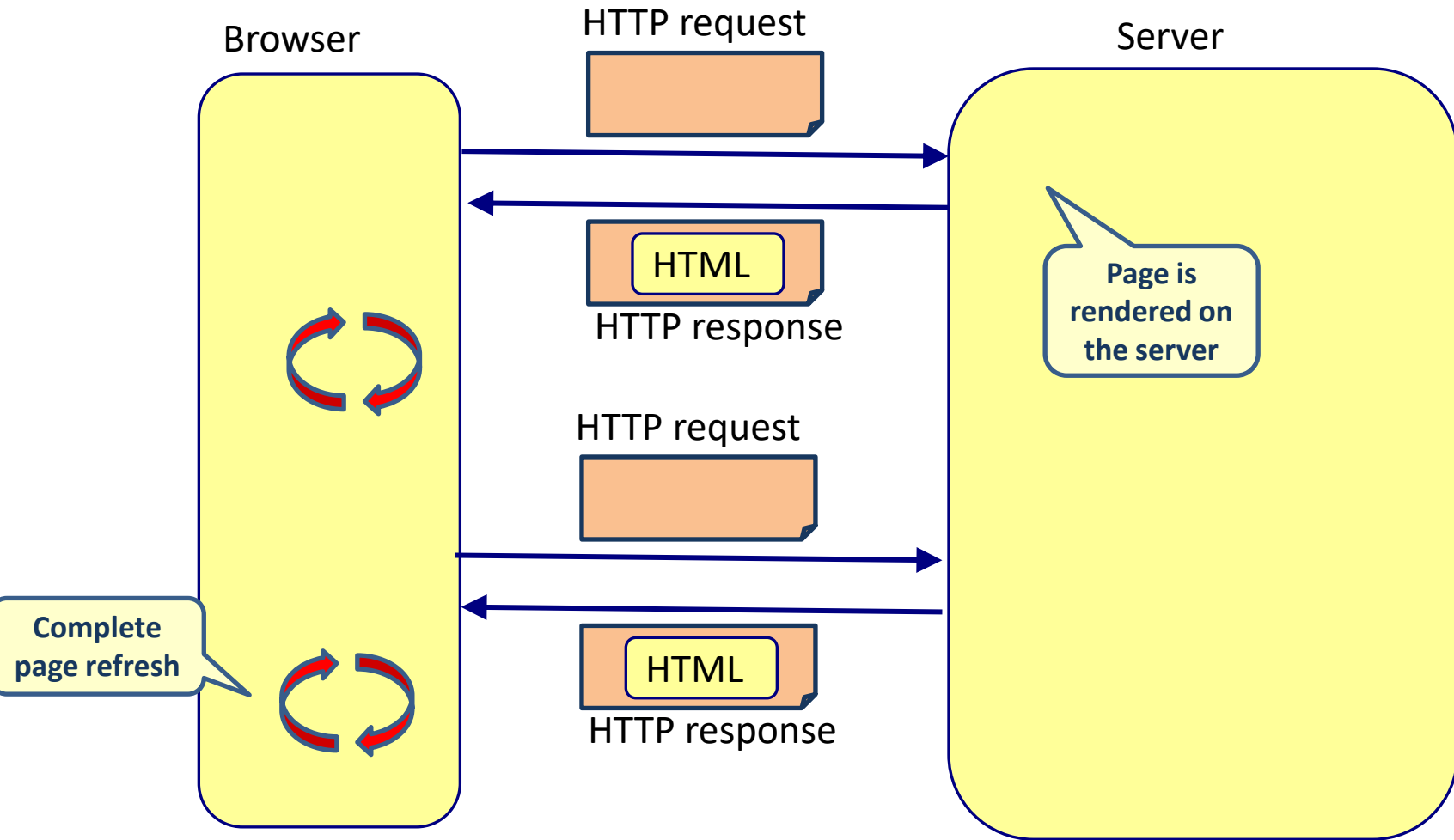
What is React?

- A Javascript library for creating web user interfaces for a Single Page Application (SPA)
- Declarative
 - You focus only on what your component should look like/do, rather than how to access certain elements in the DOM.
- Component-based

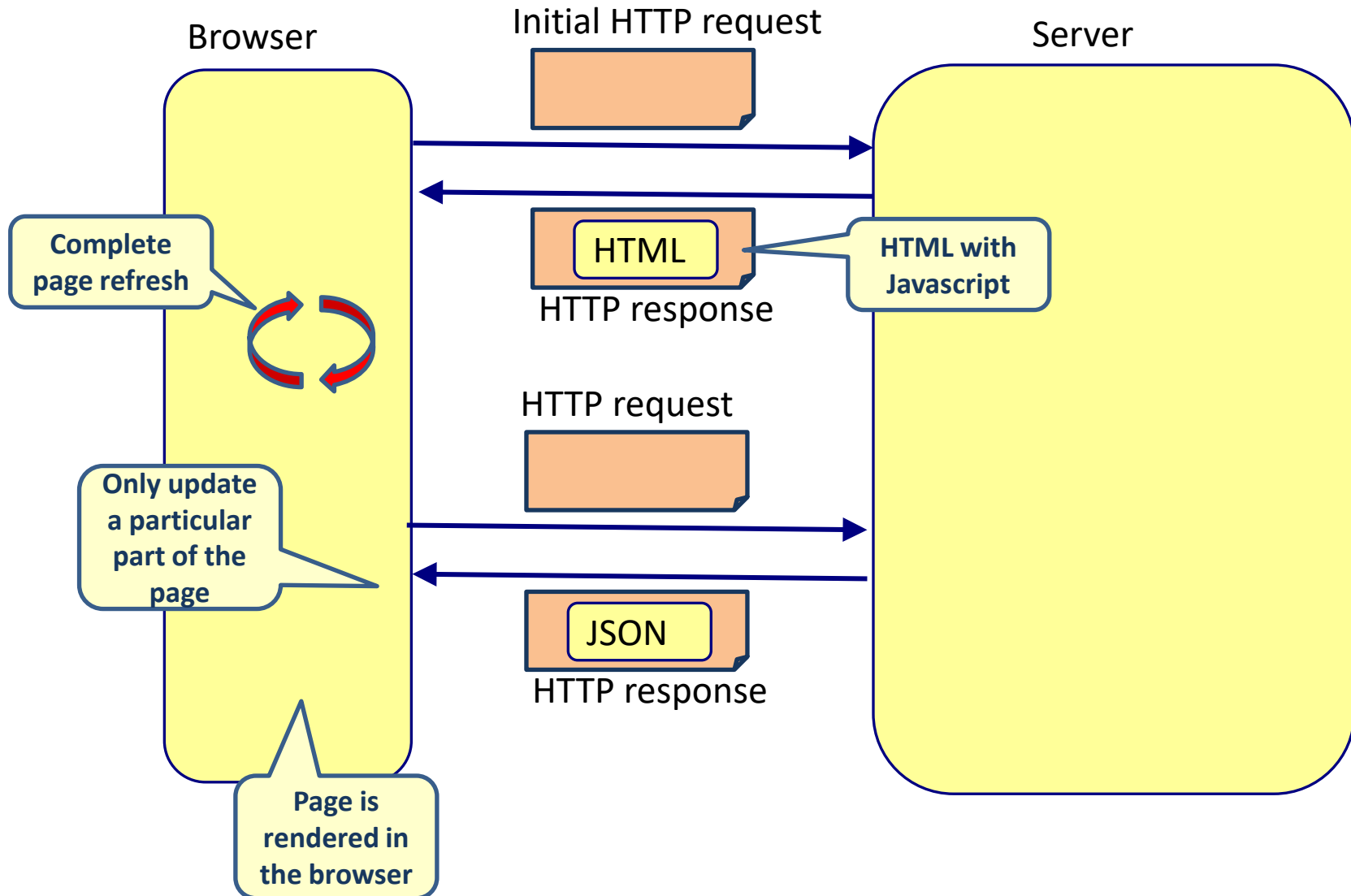
What is React?

- Initially developed by Facebook
 - Used in WhatsApp & Instagram
- Now open source
 - Used by Netflix, Airbnb, Yahoo!Mail, Dropbox,...
- Simple to learn
- Fast performance
- Small in size

Spring MVC



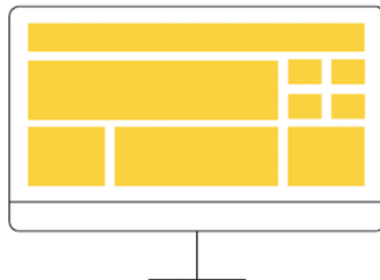
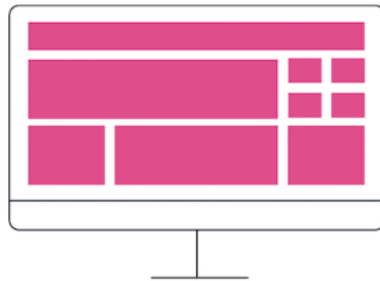
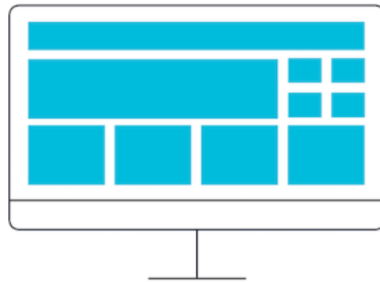
Single Page Application (SPA)



Spring MVC vs. React

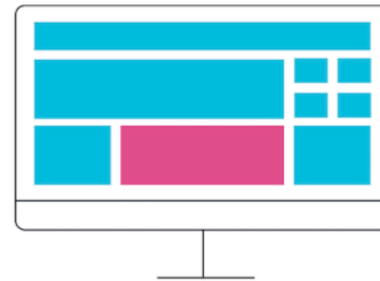
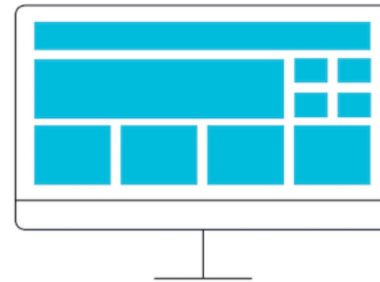
Traditional

Every request for new information gives you a new version of the whole page.

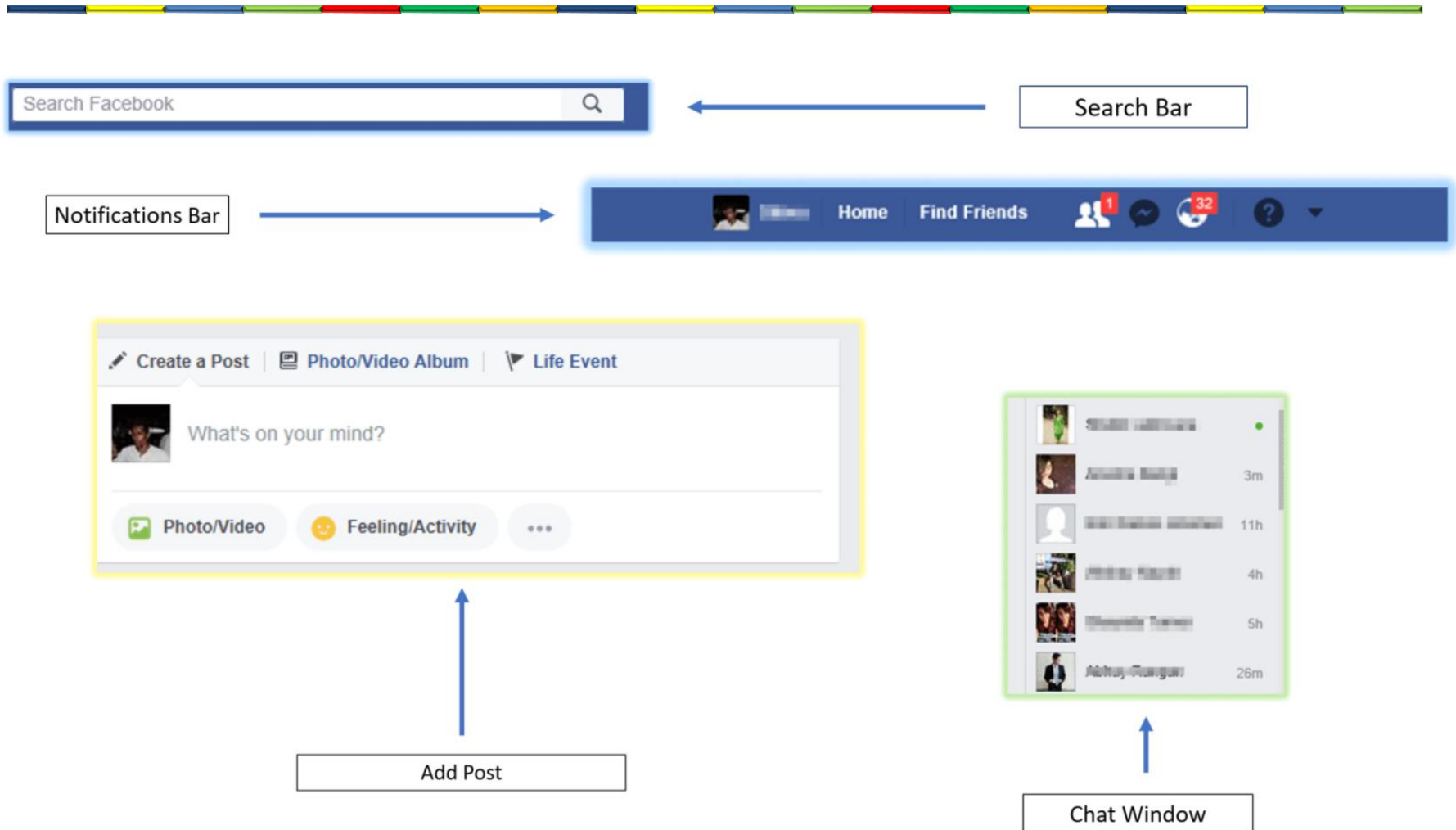


SPA

You request just the pieces you need.



Component based



JSX

- Javascript XML
- JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement() and/or appendChild() methods.

The DOM is completely hidden in React

HTML in JavaScript

```
let isMorning = (new Date()).getHours() < 12;
let greetingElement = isMorning
  ? <h3>Good Morning</h3>
  : <h3>Good Evening</h3>

function App() {
  return (
    <div>
      {greetingElement}
      <p>
        Hello world
      </p>
    </div>
  );
}
```


Babel

- Translates JSX to regular JavaScript

```
1 function GroceryList() {  
2   render (  
3     <div>  
4       <h1>Grocery List</h1>  
5       <h3>Produce</h3>  
6       <ul>  
7         <li>Bananas</li>  
8         <li>Bell Peppers</li>  
9         <li>Zucchini</li>  
10      </ul>  
11      <h3>Pantry Items</h3>  
12      <ul>  
13        <li>Black Beans</li>  
14        <li>Pasta</li>  
15      </ul>  
16      <h3>Freezer Items</h3>  
17      <ul>  
18        <li>Ice Cream</li>  
19      </ul>  
20    </div>  
21  )  
22 }
```

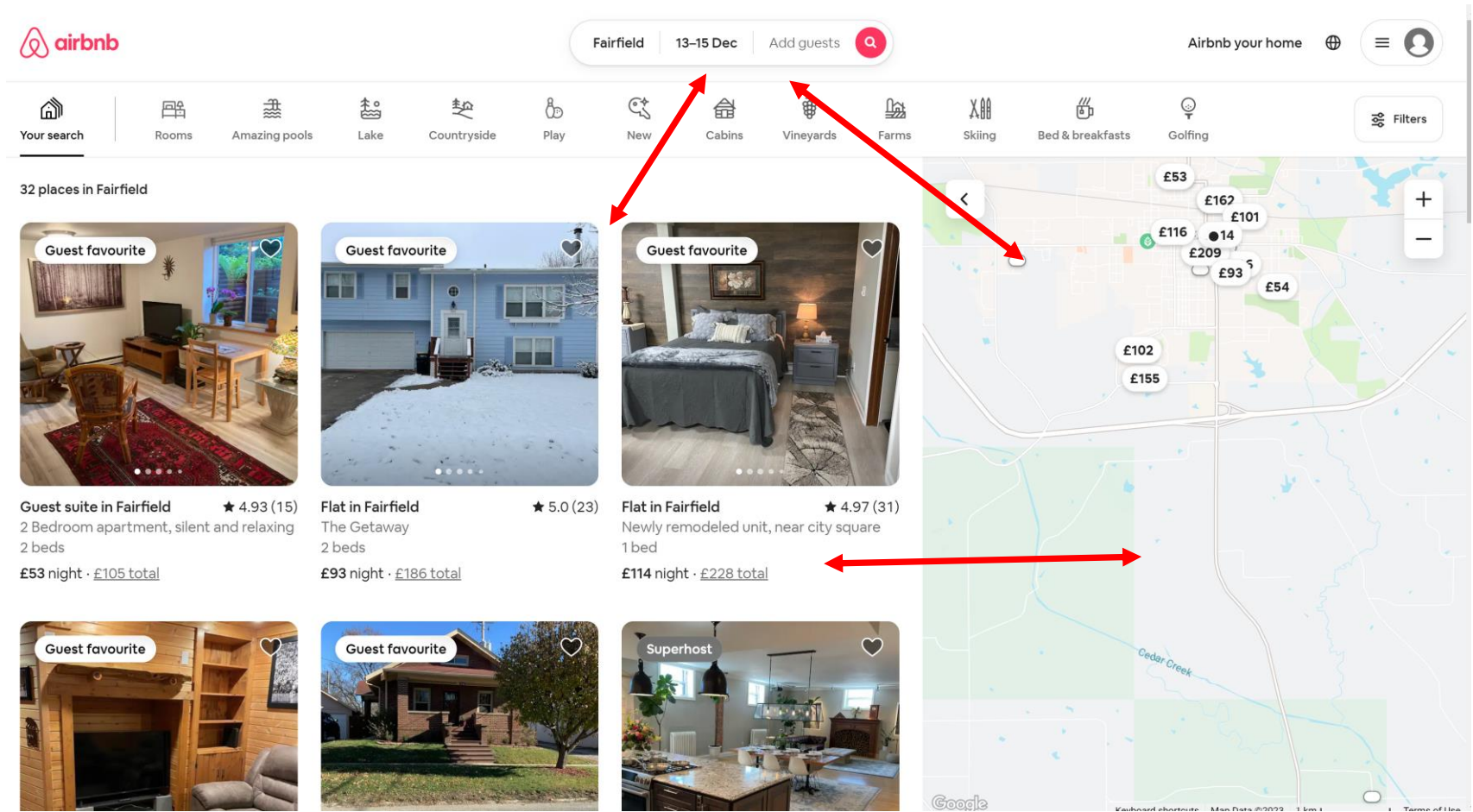
```
1 "use strict";  
2  
3 function GroceryList() {  
4   render( /*#__PURE__*/React.createElement("div", null,  
    /*#__PURE__*/React.createElement("h1", null, "Grocery List"),  
    /*#__PURE__*/React.createElement("h3", null, "Produce"),  
    /*#__PURE__*/React.createElement("ul", null,  
      /*#__PURE__*/React.createElement("li", null, "Bananas"),  
      /*#__PURE__*/React.createElement("li", null, "Bell Peppers"),  
      /*#__PURE__*/React.createElement("li", null, "Zucchini")),  
    /*#__PURE__*/React.createElement("h3", null, "Pantry Items"),  
    /*#__PURE__*/React.createElement("ul", null,  
      /*#__PURE__*/React.createElement("li", null, "Black Beans"),  
      /*#__PURE__*/React.createElement("li", null, "Pasta")),  
    /*#__PURE__*/React.createElement("h3", null, "Freezer Items"),  
    /*#__PURE__*/React.createElement("ul", null,  
      /*#__PURE__*/React.createElement("li", null, "Ice Cream"))));  
5 }
```

WHY A FRONT-END FRAMEWORK LIKE REACT

Vanilla JavaScript

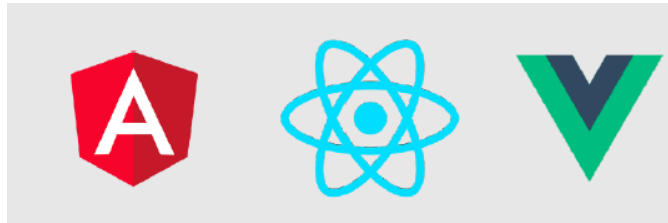
- Imperative
 - Write code how to do things
 - A lot of DOM traversal and manipulation code
 - A lot of spaghetti code to keep the state and the UI in sync.

Keep state and UI in sync



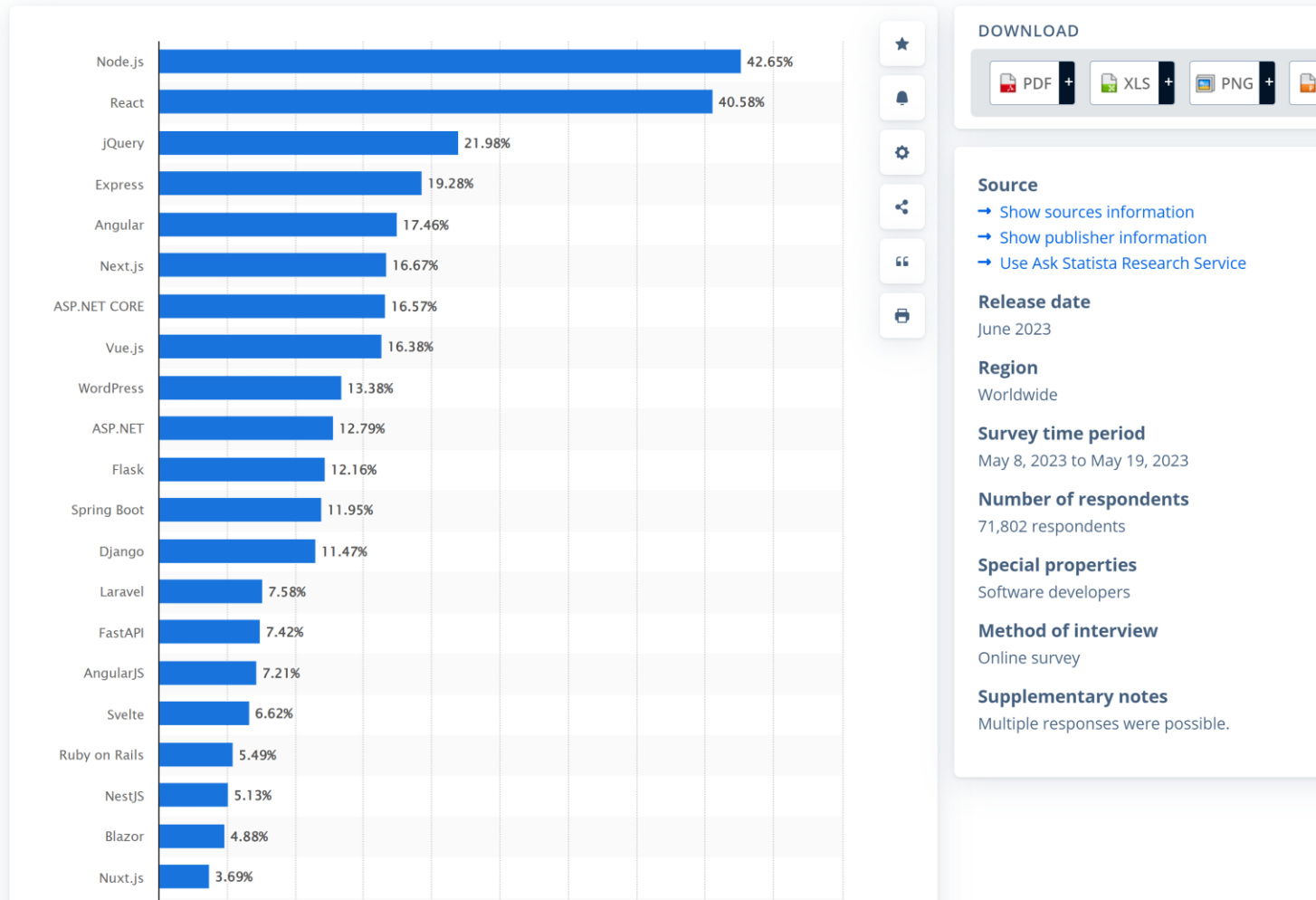
Front-end frameworks

- Declarative
 - Write what you want, not how you want to do it.
 - Abstract the DOM away
 - Easy to keep data and UI in sync



Popularity of React

Most used web frameworks among developers worldwide, as of 2023



Main point

- A front-end web framework completely abstracts the DOM from the developer. *The Unified Field is an abstract field from which the whole relative creation arises.*

HELLO WORLD REACT

Minimal Hello World React

Just open this file in the browser

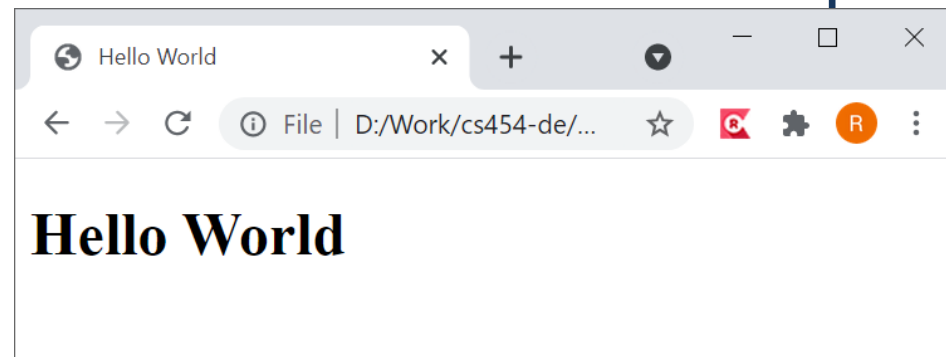
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Hello World</title>
  <script src="https://unpkg.com/react@15/dist/react.js"></script>
  <script src="https://unpkg.com/react-dom@15/dist/react-dom.js"></script>
  <script src="https://unpkg.com/babel-standalone/babel.min.js"></script>
  <script type="text/babel">

    const HelloWorld = () => {
      return (
        <h1>Hello World</h1>
      );
    }

    ReactDOM.render(
      <HelloWorld />,
      document.getElementById("content")
    );
  </script>
</head>
<body>
  <div id="content"></div>
</body>
</html>
```

Component

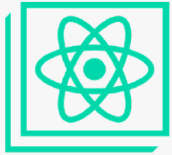
Render the component in the
DOM element with id "content"



JavaScript

- Node
 - A JavaScript runtime that enables us to run JavaScript outside of a browser.
 - It also enables us to run JavaScript on the server-side
- NPM
 - Stands for Node Package Manager and is a tool that allows us to install and manage node packages
- NPX
 - NPX is an NPM package runner that makes it really easy to install any sort of package

Creating a react app



CREATE-REACT-APP

- Simple
- Slower than vite
- For learning and tutorials



VITE

- More complex to setup
- Very fast
- For real world applications

Creating a React application

- *\$ npx create-react-app my-app*
- Generate a react app, named **my-app**, in the path that the command was run in using the create-react-app package.
- This will create a react application including all necessary packages and dependencies
- For details see
 - <https://create-react-app.dev/docs/getting-started>

Starting a React application

- *\$ npm start*
- Start a webserver and run the current react app (index.html) on this webserver at port localhost:3000

create-react-app

✓ HELLOWORLD

> node_modules

Javascript modules

✓ public

★ favicon.ico

<> index.html

index.html is the single page in the SPA and is shown by default

🖼 logo192.png

🖼 logo512.png

{ } manifest.json

☰ robots.txt

✓ src

App.css

JS App.js

JS App.test.js

index.css

JS index.js

index.js is rendered in index.html

🖼 logo.svg

JS reportWebVitals.js

JS setupTests.js

📄 .gitignore

{ } package-lock.json

{ } package.json

📖 README.md

create-react-app

index.html

```
<title>React App</title>
</head>
<body>

  <div id="root"></div>

</body>
```

index.js

```
ReactDOM.render(
  <React.StrictMode>
    | <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

Show the App component in index.html at the element with id "root"

app.js

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

Hello World React

App.js

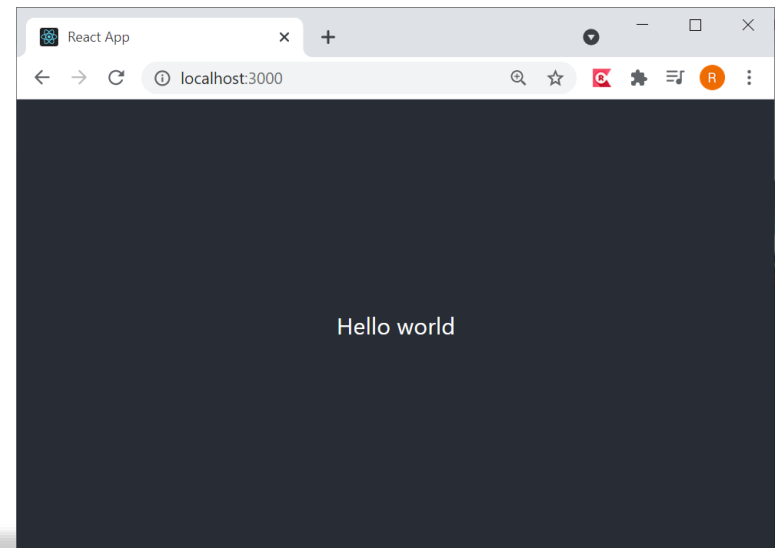
```
import './App.css';

function App() {
  return (
    <div>
      <header>
        <p>
          Hello world
        </p>
      </header>
    </div>
  );
}

export default App;
```

JSX code

React
component that
is shown on the
page



Hello World React

App.js

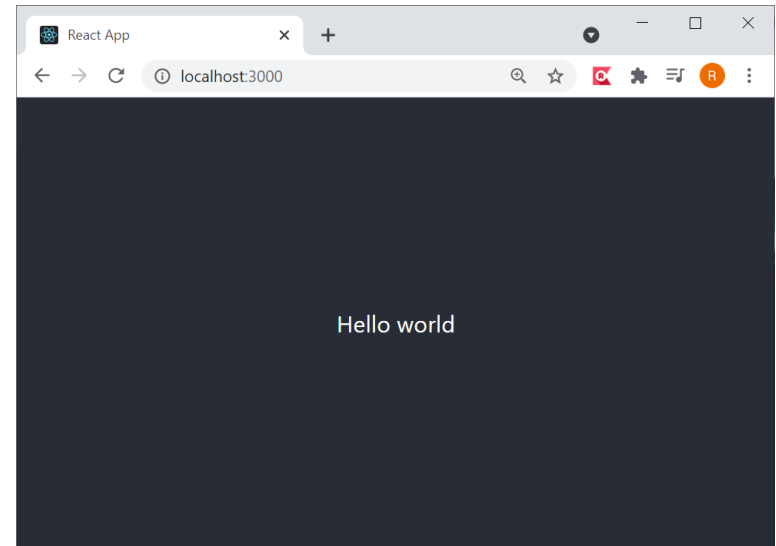
```
import './App.css';  
let message = "Hello world";
```

Define a variable

```
function App() {  
  return (  
    <div>  
      <header>  
        <p>  
          {message}  
        </p>  
      </header>  
    </div>  
  );  
}
```

Show the
variable

```
export default App;
```



Hello World React

App.js

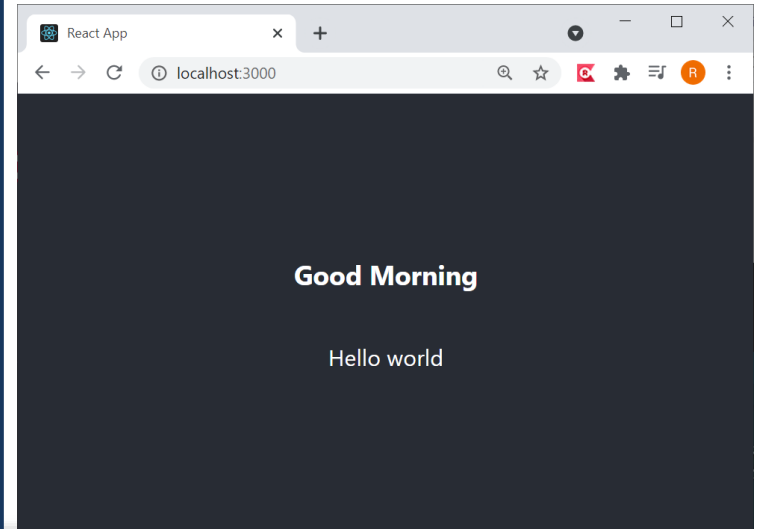
```
import './App.css';
let isMorning = (new Date()).getHours() < 12;
let greetingElement = isMorning
? <h3>Good Morning</h3>
: <h3>Good Evening</h3>

function App() {
  return (
    <div>
      <header>
        {greetingElement}
        <p>
          Hello world
        </p>
      </header>
    </div>
  );
}

export default App;
```

Define a variable

Show the variable



Button clicks

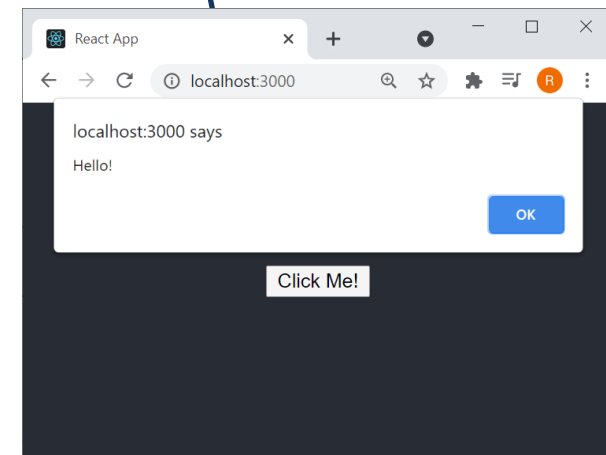
App.js

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

function App() {
  const displayAlert = () => {
    alert('Hello!');
  }

  return (
    <div>
      <header>
        <p>
          <button onClick={displayAlert}>Click Me!</button>
        </p>
      </header>
    </div>
  );
}

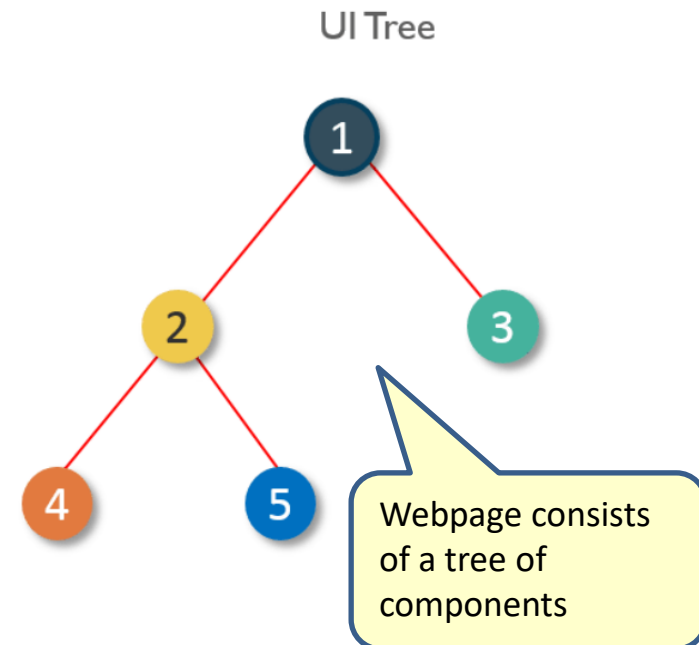
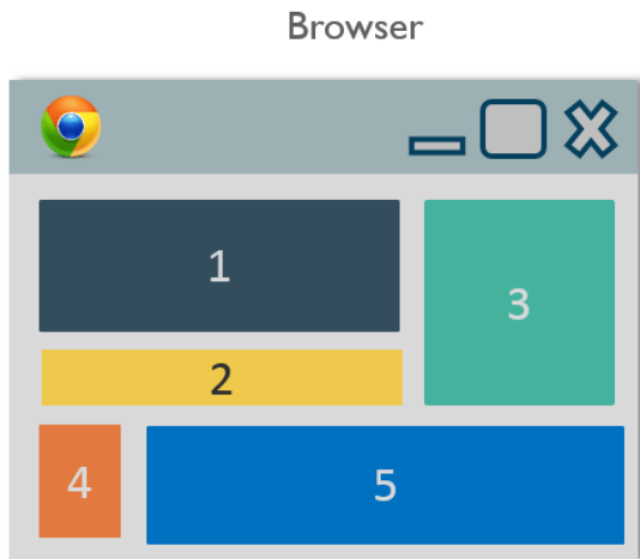
export default App;
```



REACT COMPONENTS

React components

- In react everything is a component
- Components have their own state
- Components are rendered individually
- Components can be nested



Simple component

Greeting.js

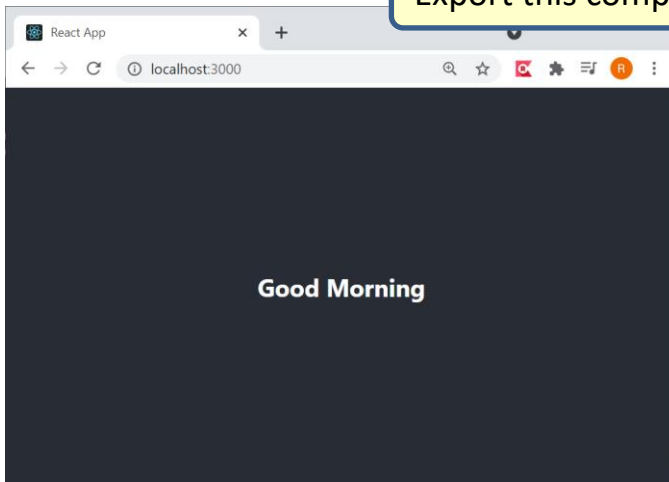
```
import React from 'react';
```

This import is always needed

```
function Greeting(){  
  let isMorning = (new Date()).getHours() < 12;  
  let greetingElement = isMorning  
    ? <h3>Good Morning</h3>  
    : <h3>Good Evening</h3>;  
  return greetingElement ;  
}
```

```
export default Greeting;
```

Export this component



App.js

Import component

```
import './App.css';  
import Greeting from './Greeting';  
  
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <p>  
          <Greeting />  
        </p>  
      </header>  
    </div>  
  );  
}
```

Show component
on page

```
export default App;
```

Arrow function

Greeting.js

```
import React from 'react';

export const Greeting = () =>{
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning</h3>
        <h4>Enjoy your day</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening</h3>
        <h4>Enjoy your day</h4>
      </>
    );
  return greetingElement ;
}
```

Arrow function

Named import

App.js

```
import './App.css';
import {Greeting} from './Greeting';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <p>
          <Greeting />
        </p>
      </header>
    </div>
  );
}

export default App;
```

Class and function components

```
import React, {Component} from 'react';
export class HelloComponentClass extends Component{
  render(){
    return(
      <h1>Hello World</h1>
    );
  }
}
```

Class component

```
import React from 'react';
export function HelloComponentSimpleFunction () {
  return (
    <h1>Hello World</h1>
  );
}
```

Functional component

```
import React from 'react';
export const HelloComponentFunction = () => {
  return (
    <h1>Hello World</h1>
  );
}
```

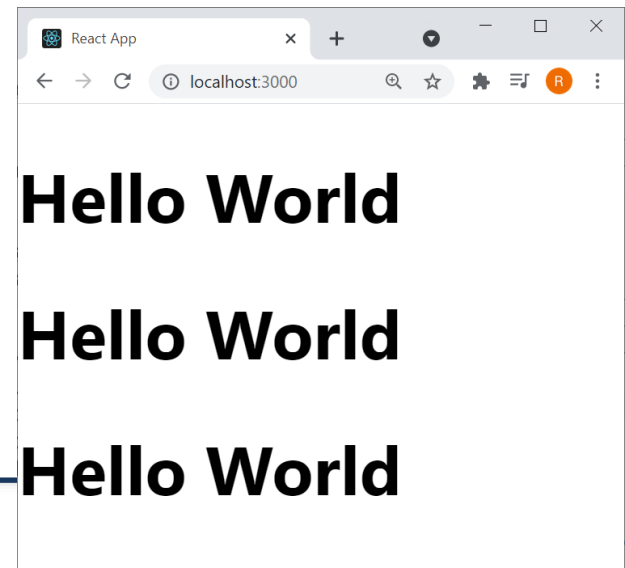
Functional component using arrow notation

Class and function components

```
import './App.css';
import {HelloComponentClass} from './pages/HelloComponentClass';
import {HelloComponentFunction} from './pages/HelloComponentFunction';
import {HelloComponentSimpleFunction} from './pages/HelloComponentSimpleFunction';

function App() {
  return (
    <div>
      <header>
        <HelloComponentClass />
        <HelloComponentFunction />
        <HelloComponentSimpleFunction />
      </header>
    </div>
  );
}

export default App;
```



Simple component

Greeting.js

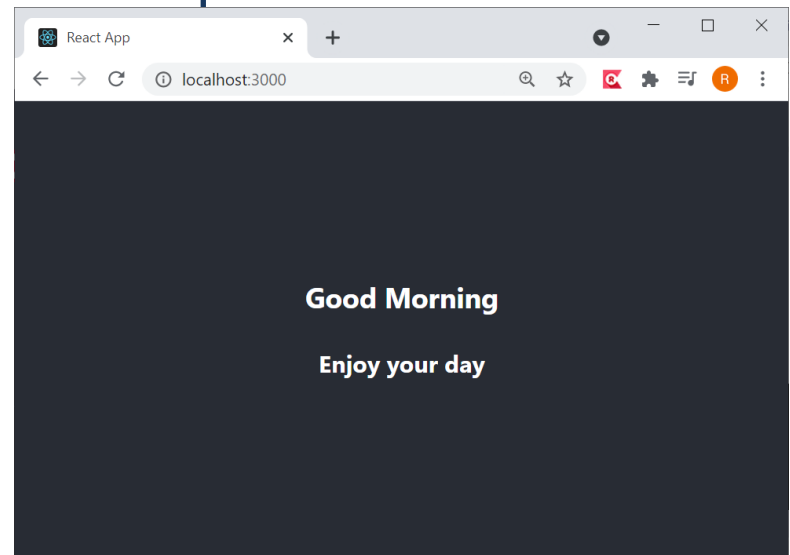
```
import React from 'react';

function Greeting(){
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning</h3>
        <h4>Enjoy your day</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening</h3>
        <h4>Enjoy your day</h4>
      </>
    );
  return greetingElement ;
}

export default Greeting;
```

A component should have a top-level element

Wrap 2 elements together



Multiple components

App.js

```
import {Component1} from './pages/Component1';
import {Component2} from './pages/Component2';
import {Component3} from './pages/Component3';
import {Component4} from './pages/Component4';
import './App.css';

function App() {
  return (
    <table>
      <tr>
        <td><Component1 /></td>
        <td><Component2 /></td>
      </tr>
      <tr>
        <td><Component3 /></td>
        <td><Component4 /></td>
      </tr>
    </table>
  );
}
```

Import component

Show components
on page

COMPONENTS

- > node_modules
- > public

src

pages

JS Component1.js

JS Component2.js

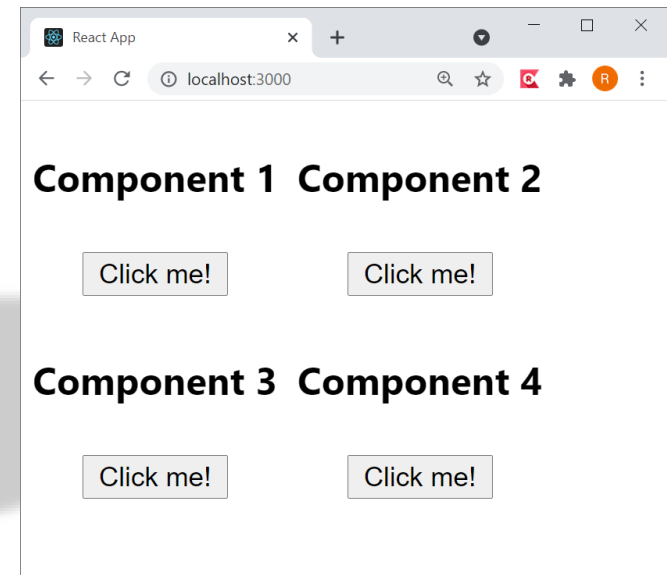
JS Component3.js

JS Component4.js

App.css

JS App.js

JS App.test.js



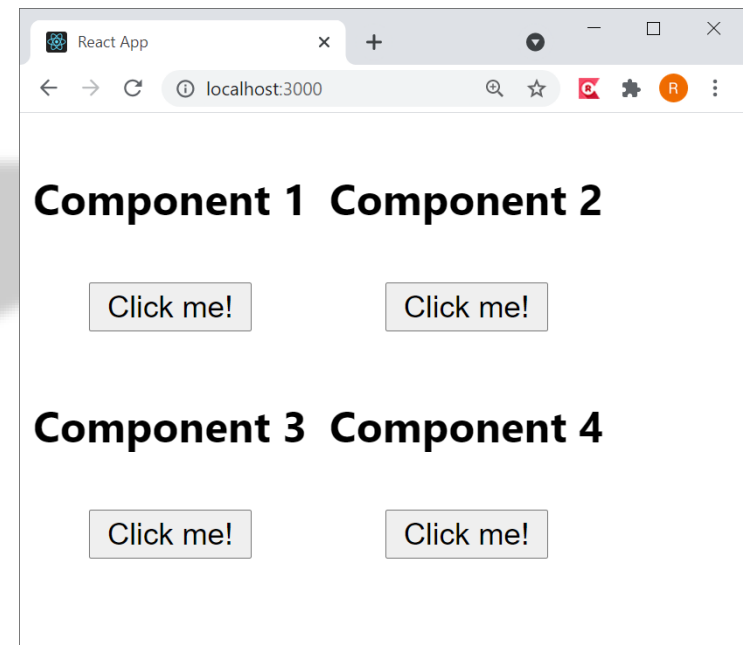
Multiple components

Component1.js

```
import React from 'react';

export const Component1 = () => {
  let component = (
    <table>
      <tr>
        <th><h3>Component 1</h3></th>
      </tr>
      <tr>
        <th><button>Click me!</button></th>
      </tr>
    </table>
  );
  return component;
}
```

- COMPONENTS
 - node_modules
 - public
 - src
 - pages
 - JS Component1.js
 - JS Component2.js
 - JS Component3.js
 - JS Component4.js
 - # App.css
 - JS App.js
 - JS App.test.js



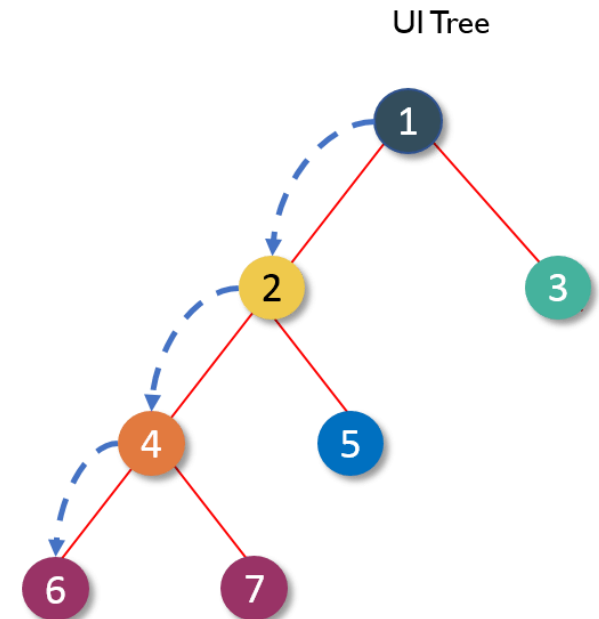
Main point

- An React component contains data, logic and appearance . *Rishi (knower), devata (process of knowing) and chhandas (known) coexist together within unity.*

PROPERTIES

props

- Props stand for properties
- Props is a way of passing data from parent to child component
 - Like arguments being passed into a function
- All properties are combined into a single object.
- Props are read-only inside of a component.



Pass props to components

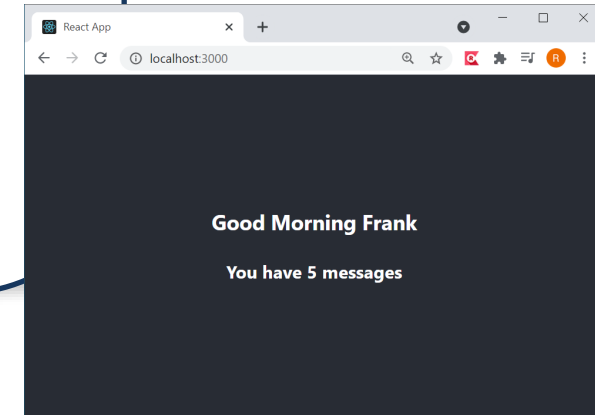
Greeting.js

```
import React from 'react';

export const Greeting = (props) =>{
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning {props.name}</h3>
        <h4>You have {props.nrOfMessages} messages</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening {props.name}</h3>
        <h4>You have {props.nrOfMessages} messages</h4>
      </>
    );
  return greetingElement ;
}
```

Object that contains all properties

Use the properties



Pass props to components

App.js

```
import './App.css';
import {Greeting} from './Greeting';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <p>
          <Greeting name="Frank" nrOfMessages = {5}/>
        </p>
      </header>
    </div>
  );
}

export default App;
```

Call component with properties



Good Morning Frank

You have 5 messages

Pass props to components

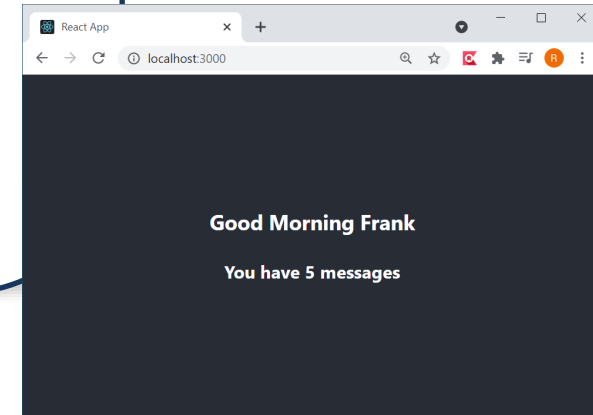
Greeting.js

```
import React from 'react';

export const Greeting = (props) =>{
  const {name, nrOfMessages } = props;
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning {name}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening {name}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    );
  return greetingElement ;
}
```

Better way

You do not have to put props for every property



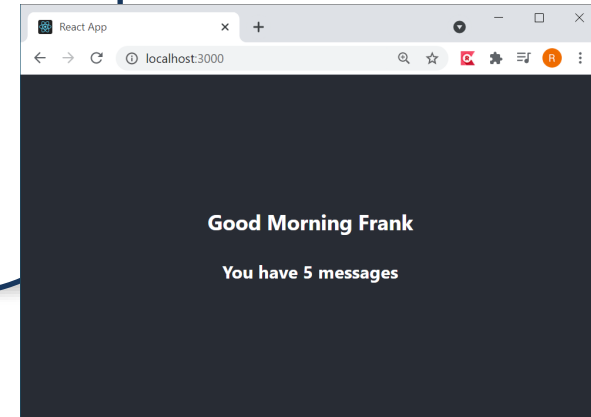
Pass props to components

Greeting.js

Even better way

```
import React from 'react';

export const Greeting = ({name, nrOfMessages } ) =>{
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning {name}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening {name}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    );
  return greetingElement ;
}
```



Passing an object

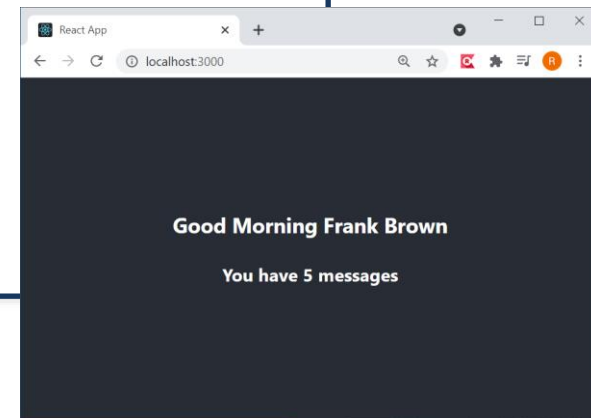
Greeting.js

```
import React from 'react';

export const Greeting = ({person, nrOfMessages }) =>{
  let isMorning = (new Date()).getHours() < 12;
  let greetingElement = isMorning
    ? (
      <>
        <h3>Good Morning {person.firstName} {person.lastName}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    )
    : (
      <>
        <h3>Good Evening {person.firstName} {person.lastName}</h3>
        <h4>You have {nrOfMessages} messages</h4>
      </>
    );
  return greetingElement ;
}
```

object

object



Passing an object

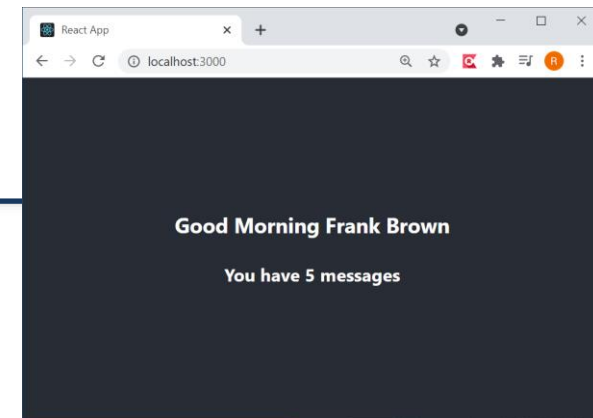
App.js

```
import './App.css';
import {Greeting} from './Greeting';

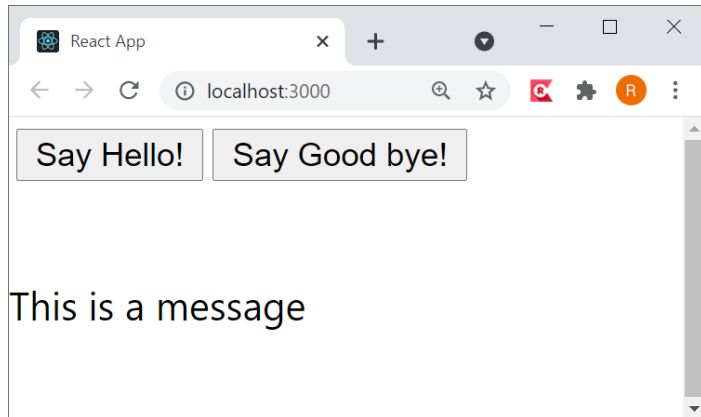
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <p>
          <Greeting person={{ firstName:"Frank", lastName:"Brown"}} nrOfMessages = {5}/>
        </p>
      </header>
    </div>
  );
}

export default App;
```

Pass an object



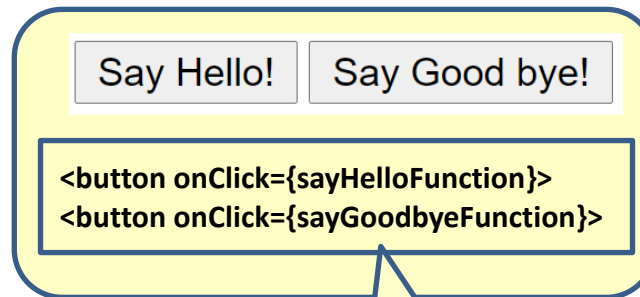
Passing a function



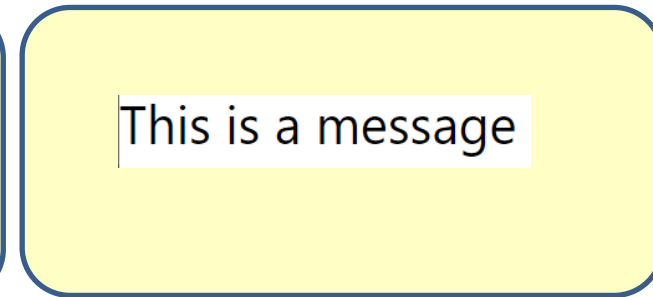
App.js

```
const sayHello = () => {  
  console.log('Hello was clicked.');}  
const sayGoodbye = () => {  
  console.log('Good bye was clicked.');}
```

TopComponent.js



BottomComponent.js



Call a function defined in
the parent component

Passing a function

App.js

```
import './App.css';
import { TopComponent } from './pages/TopComponent';
import { BottomComponent } from './pages/BottomComponent';

function App() {

  const sayHello = () => {
    console.log('Hello was clicked.');
```

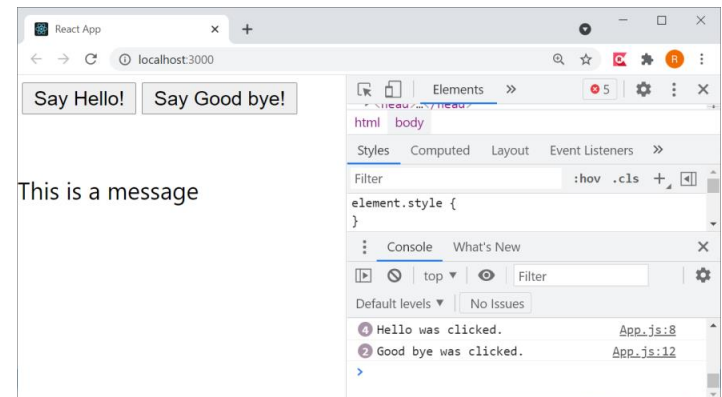
```
  }

  const sayGoodbye = () => {
    console.log('Good bye was clicked.');
```

```
  }

  return (
    <div>
      <TopComponent sayHelloFunction={sayHello} sayGoodbyeFunction={sayGoodbye} />
      <BottomComponent/ >
    </div>
  );
}

export default App;
```



Pass a function

Pass a function

Passing a function

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

Pass a function

Pass a function

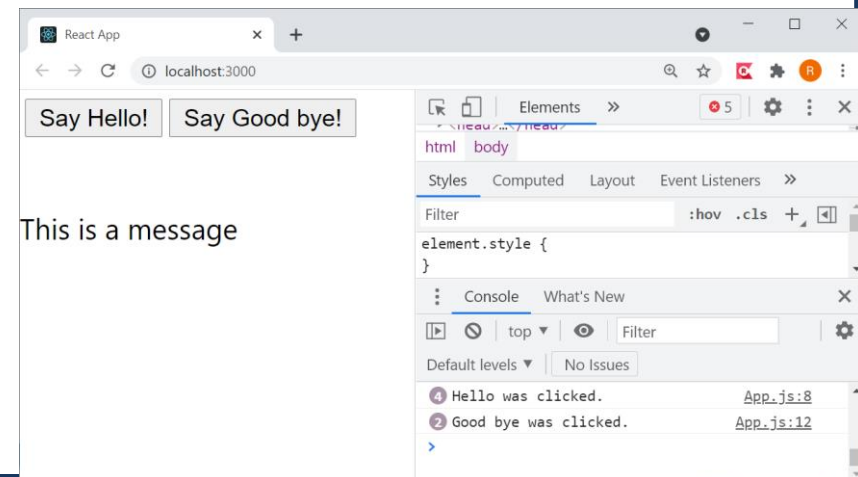
TopComponent.js

```
export const TopComponent= ({sayHelloFunction,sayGoodbyeFunction} ) => {  
  let content =  
    <table>  
      <tr>  
        <th><button onClick={sayHelloFunction}>Say Hello!</button></th>  
        <th><button onClick={sayGoodbyeFunction}>Say Good bye!</button></th>  
      </tr>  
    </table>  
  return content ;  
}
```

Call the function in the parent component

```
import React from 'react';  
export const BottomComponent = () =>{  
  let message = "This is a message";  
  
  let content =  
    <div>  
      <p>  
        {message}  
      </p>  
    </div>  
  return content ;  
}
```

BottomComponent.js



STATE IN COMPONENTS

Use state in components

```
import React from 'react';

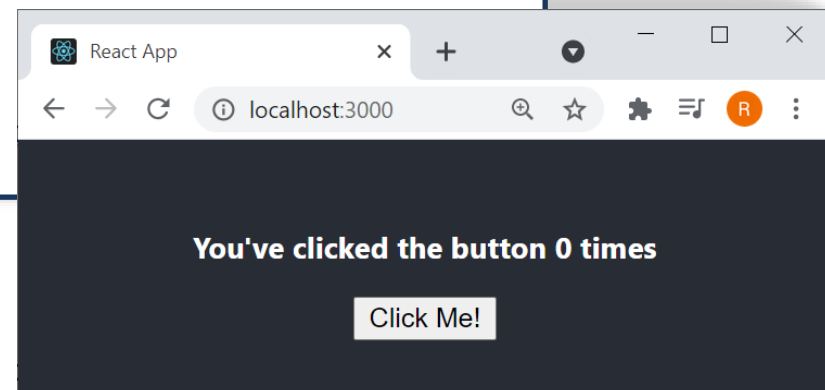
export const Counter = () =>{
  let numberOfClicks=0;

  const increment = () =>{
    numberOfClicks = numberOfClicks + 1;
    console.log(numberOfClicks);
  }

  let counterElement = (
    <>
      <h3>You've clicked the button {numberOfClicks} times</h3>
      <button onClick={increment}>Click Me!</button>
    </>
  );
  return counterElement ;
}
```

The page will not show the updated value of numberOfClicks

React will only render a component if the value of the props or the value of one of the hooks change



React Hooks

- Built-in functions that allow React developers to use state and lifecycle methods inside functional components
 - Makes working with state and lifecycle methods easier in functional components
- Important hooks
 - `useState`
 - `useEffect`

useState hook

```
function App() {
```

Variable age

Function for updating variable age

```
  const [age, setAge] = useState(19);
```

Initial value of age

```
  const handleClick = () => setAge(age + 1)
```

Update age

```
  return (
```

```
    <div>
```

```
      I am {age} Years Old
```

```
      <div>
```

```
        <button onClick={handleClick}>Increase my age! </button>
```

```
      </div>
```

```
    </div>
```

```
  )
```

```
}
```

Hi, I am 19 Years old Increase my Age!

Use state in components

Import the useState hook

Hook

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

export const Counter = () =>{
  const [numberOfClicks, setNumberOfClicks] = useState(0);

  const increment = () => setNumberOfClicks(numberOfClicks + 1);

  let counterElement = (
    <>
      <h3>You've clicked the button {numberOfClicks} times</h3>
      <button onClick={increment}>Click Me!</button>
    </>
  );
  return counterElement ;
}
```

increment method

Call the increment method

Use state in components

```
import React, {useState} from 'react';
import {Counter} from './Counter';
import {CongratulationsMessage} from './CongratulationsMessage';
import './App.css';
```

```
function App() {
  const [numberOfClicks, setNumberOfClicks] = useState(0);

  const increment = () => setNumberOfClicks(numberOfClicks + 1);

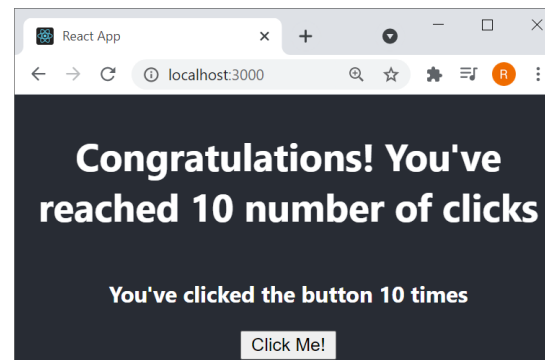
  return (
    <div className="App">
      <header className="App-header">
        <CongratulationsMessage threshold={10} numberOfClicks={numberOfClicks} />
        <Counter numberOfClicks={numberOfClicks} onIncrement={increment}/>
      </header>
    </div>
  );
}

export default App;
```

Put state in the parent

Pass data

Pass a function



Use state in components

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

Pass data

Pass a function

```
export const Counter = ({numberOfClicks, onIncrement}) =>{
```

```
  let counterElement = (
```

```
    <>
```

```
      <h3>You've clicked the button {numberOfClicks} times</h3>
```

```
      <button onClick={onIncrement}>Click Me!</button>
```

```
    </>
```

```
  );
```

```
  return counterElement ;
```

```
}
```

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

```
export const CongratulationsMessage = ({numberOfClicks, threshold}) =>{
```

```
  let messageElement = numberOfClicks >= threshold
```

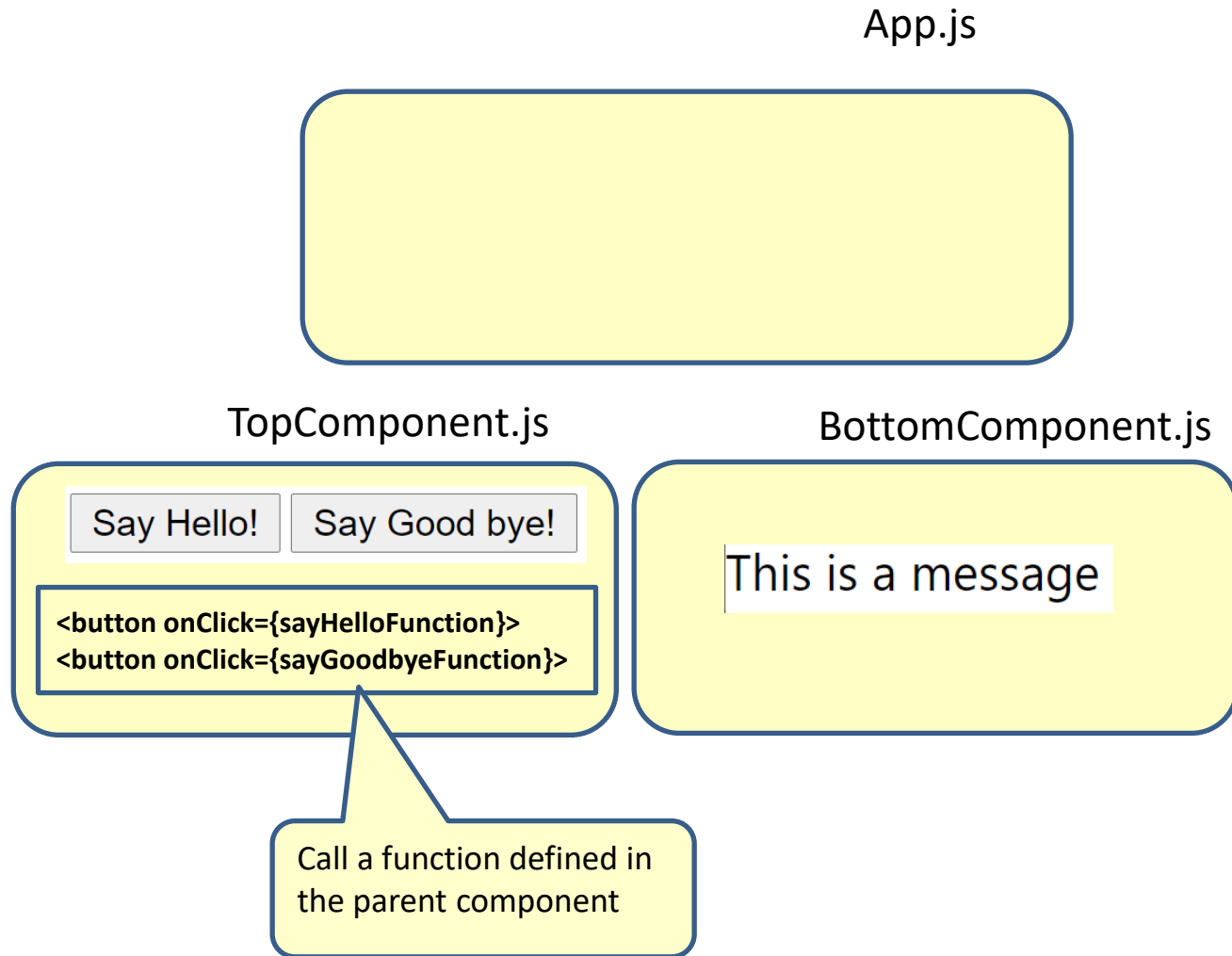
```
    ? <h1>Congratulations! You've reached {threshold} number of clicks</h1>
```

```
    : null;
```

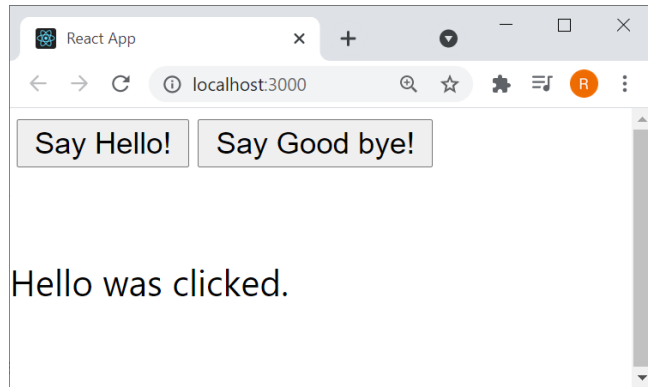
```
  return messageElement ;
```

```
}
```

Passing a function



Passing data between components



App.js

```
const [message, setMessage] = useState("");
```

Pass the setMessage function

TopComponent.js

Pass the message value

BottomComponent.js

Say Hello! Say Good bye!

```
const sayHello = () => {  
  setMessageFunction("Hello was clicked.");  
}  
  
const sayGoodbye = () => {  
  setMessageFunction("Good bye was clicked.");  
}  
  
<button onClick={sayHello}>  
<button onClick={sayGoodbye}>
```

{message}

Hello was clicked.

Use state in components

```
import './App.css';
import React, {useState} from 'react';
import { TopComponent } from './pages/TopComponent';
import { BottomComponent } from './pages/BottomComponent';
```

```
function App() {
  const [message, setMessage] = useState("");
```

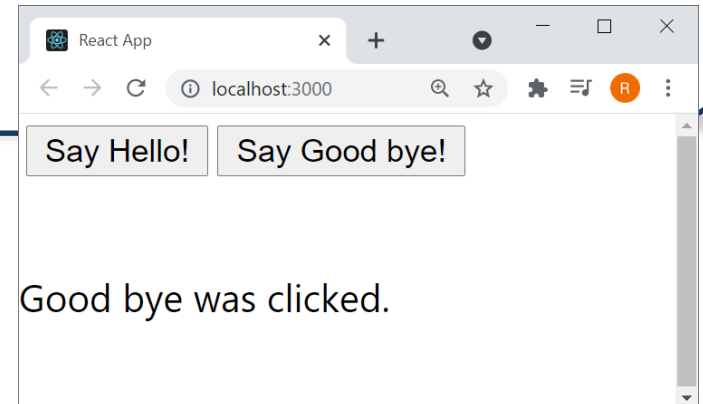
State in the top component

```
  return (
    <div>
      <TopComponent setMessageFunction={ setMessage} />
      <BottomComponent message={message} />
    </div>
  );
}
```

Pass state change function

Pass state

```
export default App;
```



Use state in components

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

TopComponent.js

```
export const TopComponent = ({ setMessageFunction }) => {
```

```
  const sayHello = () => {  
    setMessageFunction("Hello was clicked.");  
  }
```

Pass a function

```
  const sayGoodbye = () => {  
    setMessageFunction("Good bye was clicked.");  
  }
```

Call the passed function

```
  let content =  
    <table>  
      <tr>  
        <th><button onClick={sayHello}>Say Hello!</button></th>  
        <th><button onClick={sayGoodbye}>Say Good bye!</button></th>  
      </tr>  
    </table>
```

```
  return content;
```

```
}
```

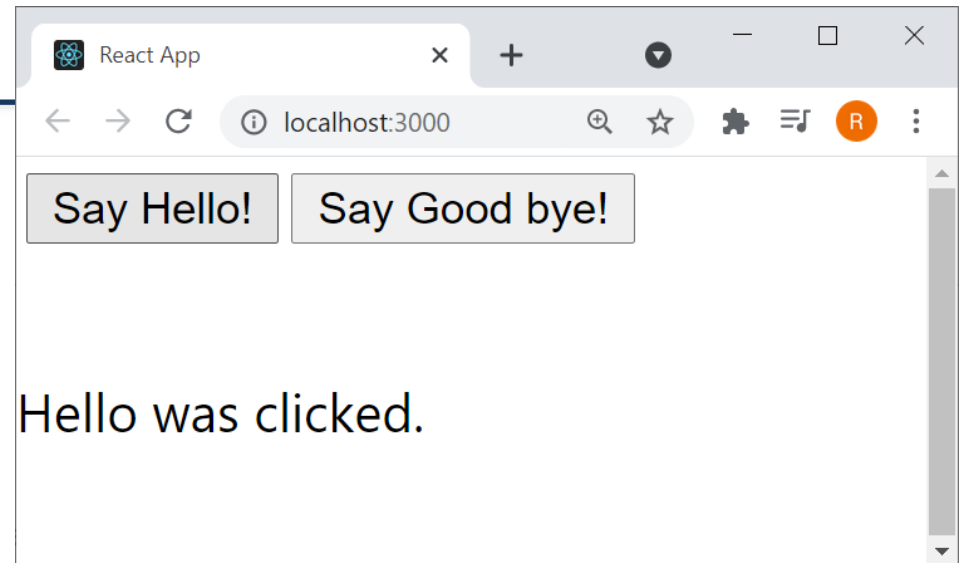
Use state in components

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

Pass the message

BottomComponent.js

```
export const BottomComponent = ({ message }) => {  
  
  let content =  
    <div>  
      <p>  
        {message}  
      </p>  
    </div>  
  return content ;  
}
```



Main point

- State in React can be managed using the useState hook. *Knowledge is different in different states of consciousness.*