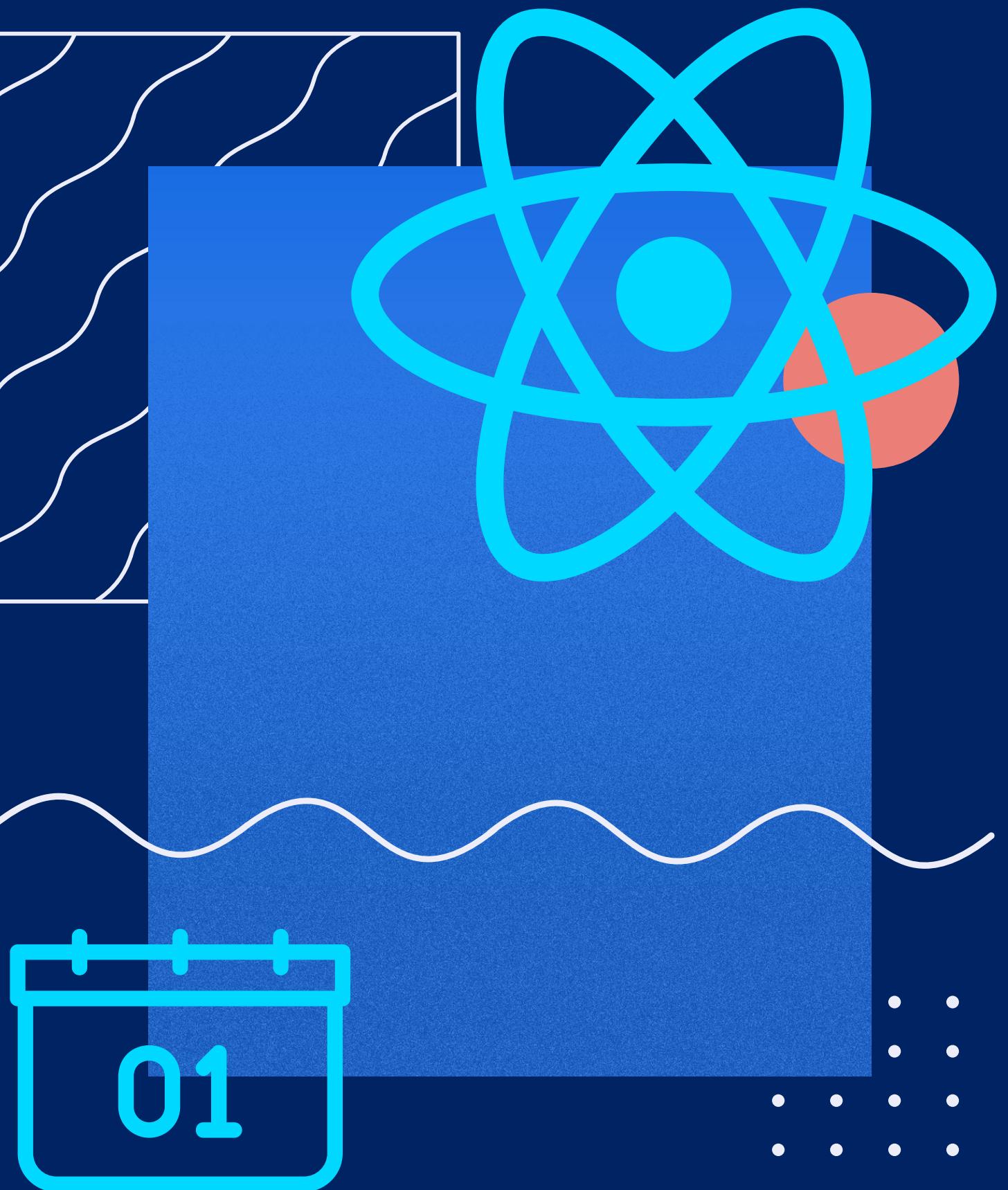
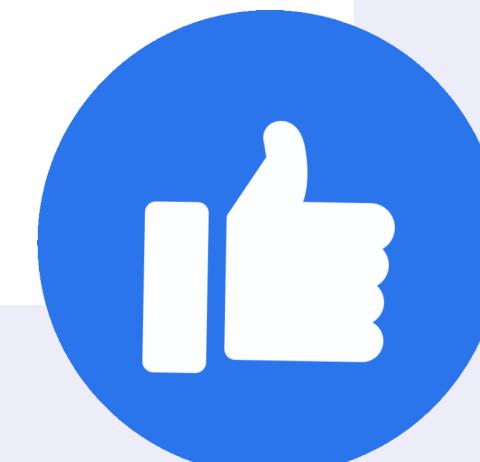
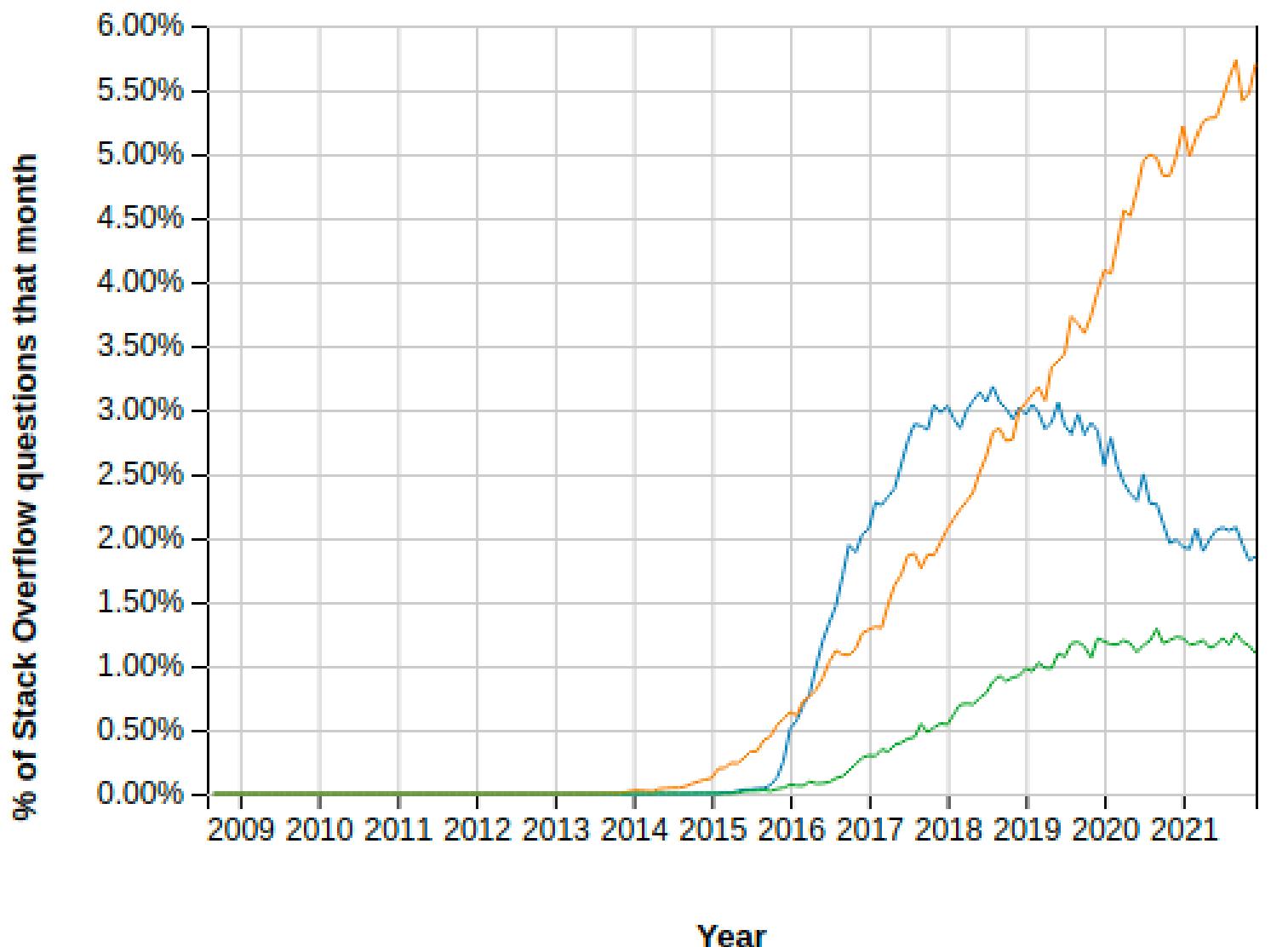


Intro to React

DOCKER + CREATE-REACT-APP



WHY REACT?



- Most popular front-end Javascript framework, released by Facebook in May 2013.
- Modular, component-based UI.
- Single page applications. Render changes to the browser through the Virtual DOM.

Image Source: **Stack Overflow Trends**
(<https://insights.stackoverflow.com/trends?tags=angular%2Creactjs%2Cvue.js>)

Old School Facebook (2004)

[thefacebook]

home search global social net invite faq logout

Scranton

Scott Pearson's Profile (This is you)

Picture [edit]



Information [edit]

Account Info:

Name: Scott Pearson, BS
Member Since: January 12, 2005
Last Update: February 3, 2005

Basic Info:

Email: persons2@scranton.edu
Status: Alumnus/Alumna
Sex: Male
Year: 2004

Concentrations: Computing Sciences, Mathematics

Phone: 570.499.4818
High School: Dunmore HS '00

Extended Info:

Screenname: ScottiePP7
Looking For: Friendship, Dating, A Relationship, Random play, Whatever I can get

Interested In: Women
Relationship Status: Single
Political Views: Liberal
Interests: Drinking, Football, Basketball, Tennis, saying you'll have that

quick search 00

My Profile [edit]
My Groups
My Friends
My Messages
My Away Message
My Mobile Info
My Account
My Privacy

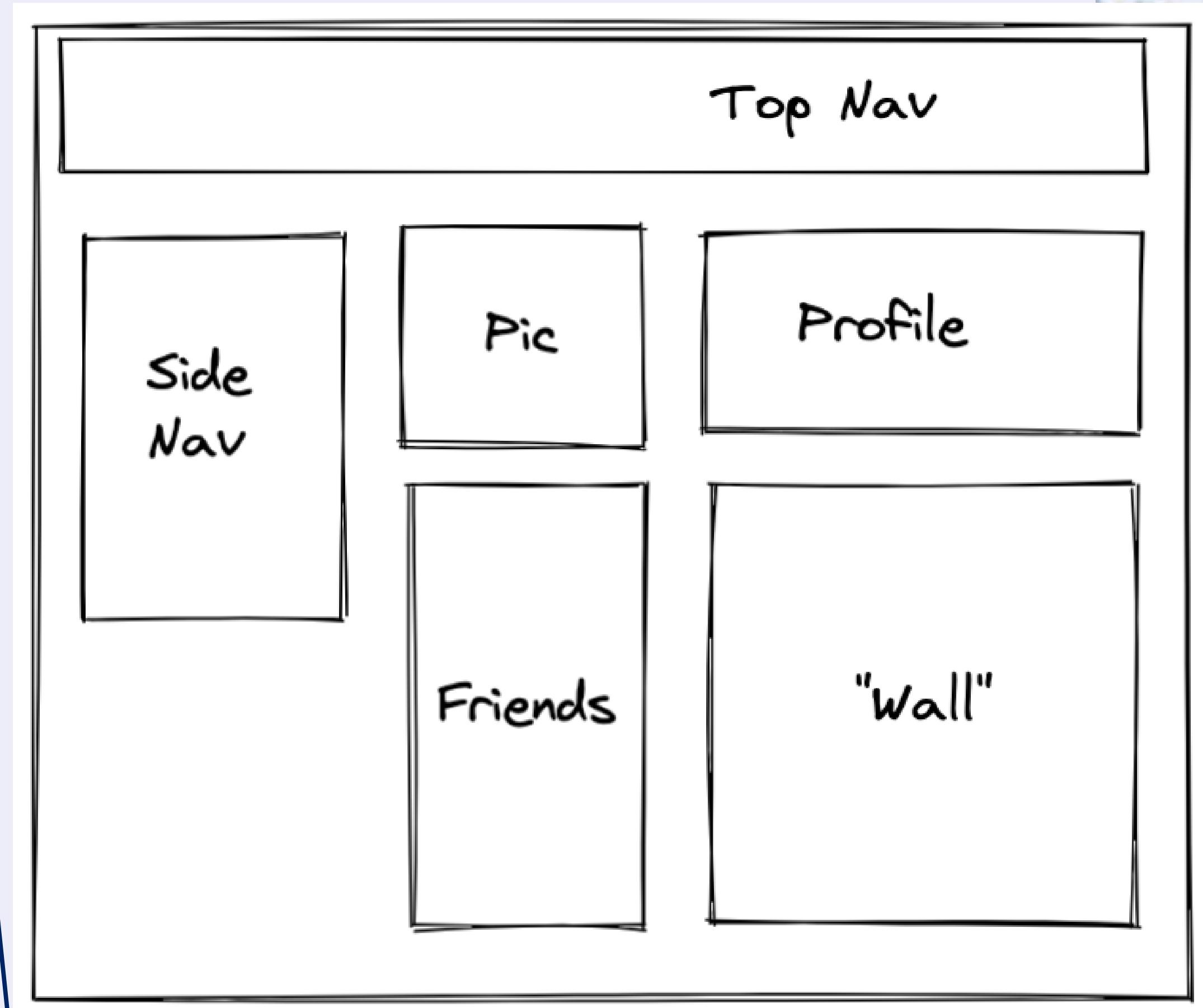
Visualize My Friends
Edit My Profile
My Account Preferences
My Privacy Preferences

Connection
This is you.

Access
Scott is currently logged in from a non-residential location.

Other Scranton [edit]

Old School Facebook



Mini-Feed (née 2006)

facebook Profile edit Friends Networks Inbox (8) home account privacy logout

Search Applications edit

- facehack
- Photos
- Groups
- Events
- Marketplace
- My Questions
- Top Friends
- + more

View Photos of You What are you doing this wee... Edit My Profile You are online now.

Gina Bianchini Profile Update your status...

Networks: Stanford Alum

Mini-Feed edit See All

Displaying 10 stories.

Today

Gina added the Handbell Podcast Player application. 8:40pm

June 19 Gina and Paul Weller are now friends. 12:02pm

June 18 Gina and Jay Tannerbaum are now friends. 4:53pm

June 17 Gina added the Ning Network Creator's Video application. 7:52pm

Gina and Sam Jadalah are now friends. 7:51pm

Gina and Ian McCarthy are now friends. 7:51pm

Gina added the House of Kyle Latest Photos application. 7:50pm

Gina added the Top Friends application. 7:49pm

Gina added the My Questions application. 7:48pm

Gina and Brooke Hammerling are now friends. 7:47pm

Stanford Friends edit See All

13 friends at Stanford.

Valerie Rozycki Reid Hoffman David Sze

Steve Vassallo Ian McCarthy Hollie Moore

Handbell Podcast Player edit X

Handbell Podcast 0711fs - Listener F 00:37 00:00 Handbell Community

Share [create | see all]

To fill out the rest of your profile, click here.

Information edit

Contact Info [edit]

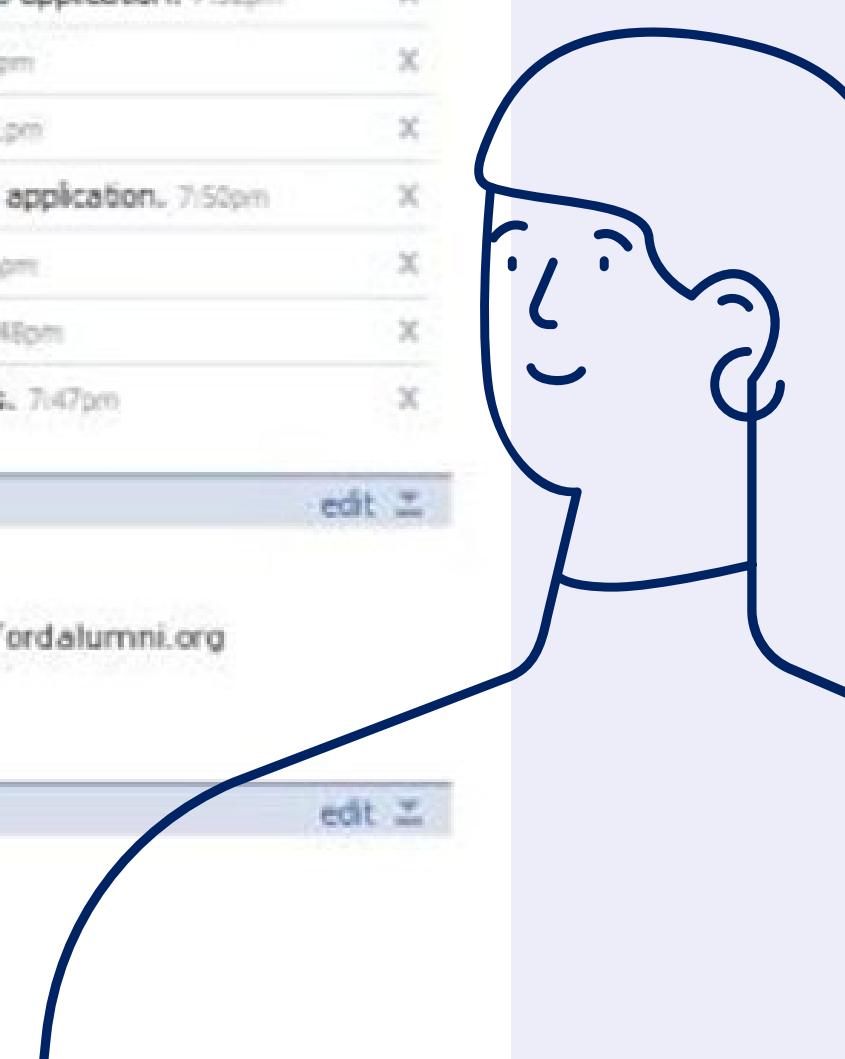
Email: gina_bianchini@stanfordalumni.org

Education and Work edit

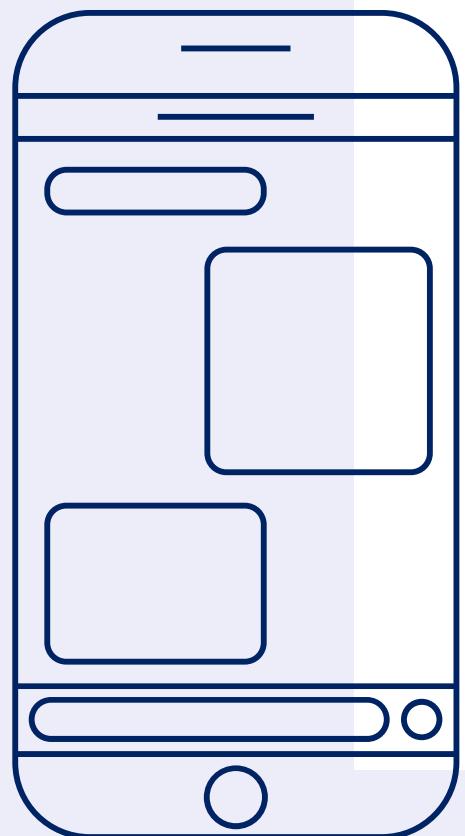
Education Info [edit]

College: Stanford Political Science

High School: Lynbrook High



The News Feed



facebook Welcome Matt!

home search browse share invite help logout

Requests 1 group invitation

My Status edit

Keep your friends updated on your current status.

Shared with You see all

ShowBizSpy - Movies From Eddie Lim

Microsoft launches the Zune! - Engadget From Soleio

Drew From Soleio

Birthdays see all

Today's Birthdays Kyle

October 7th Birthdays Joe Alisha Nick

October 8th Birthdays

News Feed

Sponsored: iTunes

Athalie and Meredith commented on the note tab energy: an anomaly in my life. 3:59pm

Carrie joined the group Marvel: Ultimate Alliance – The World's Largest Army of Super Heroes. 3:39pm

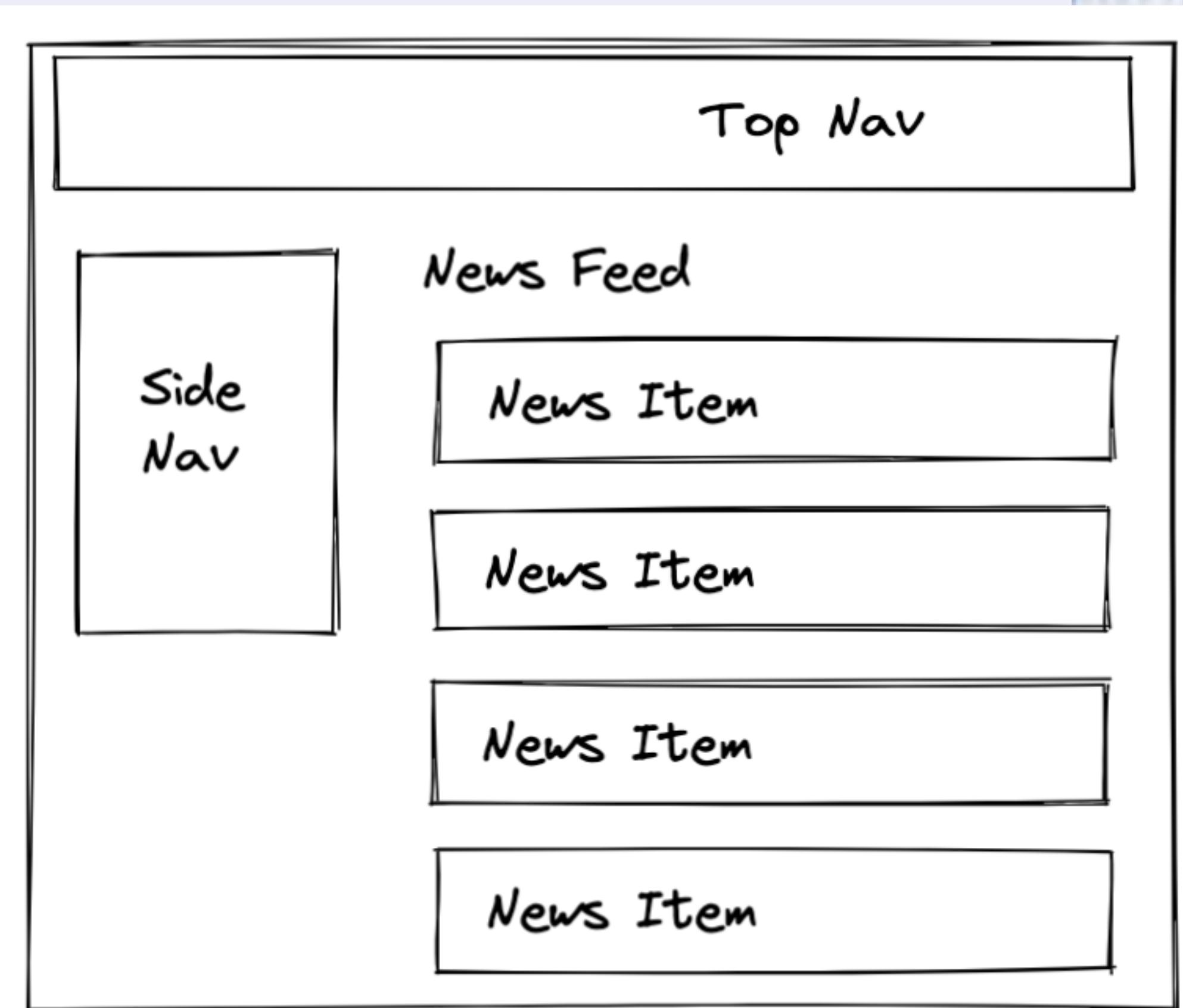
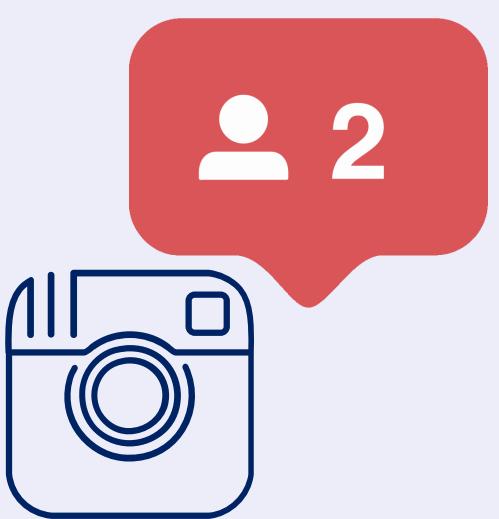
Payam Imani added new photos. 3:30pm Share Save

Gotham City: Chicago – 58 photos by Payam Location: Chicago, IL In this album: Payam

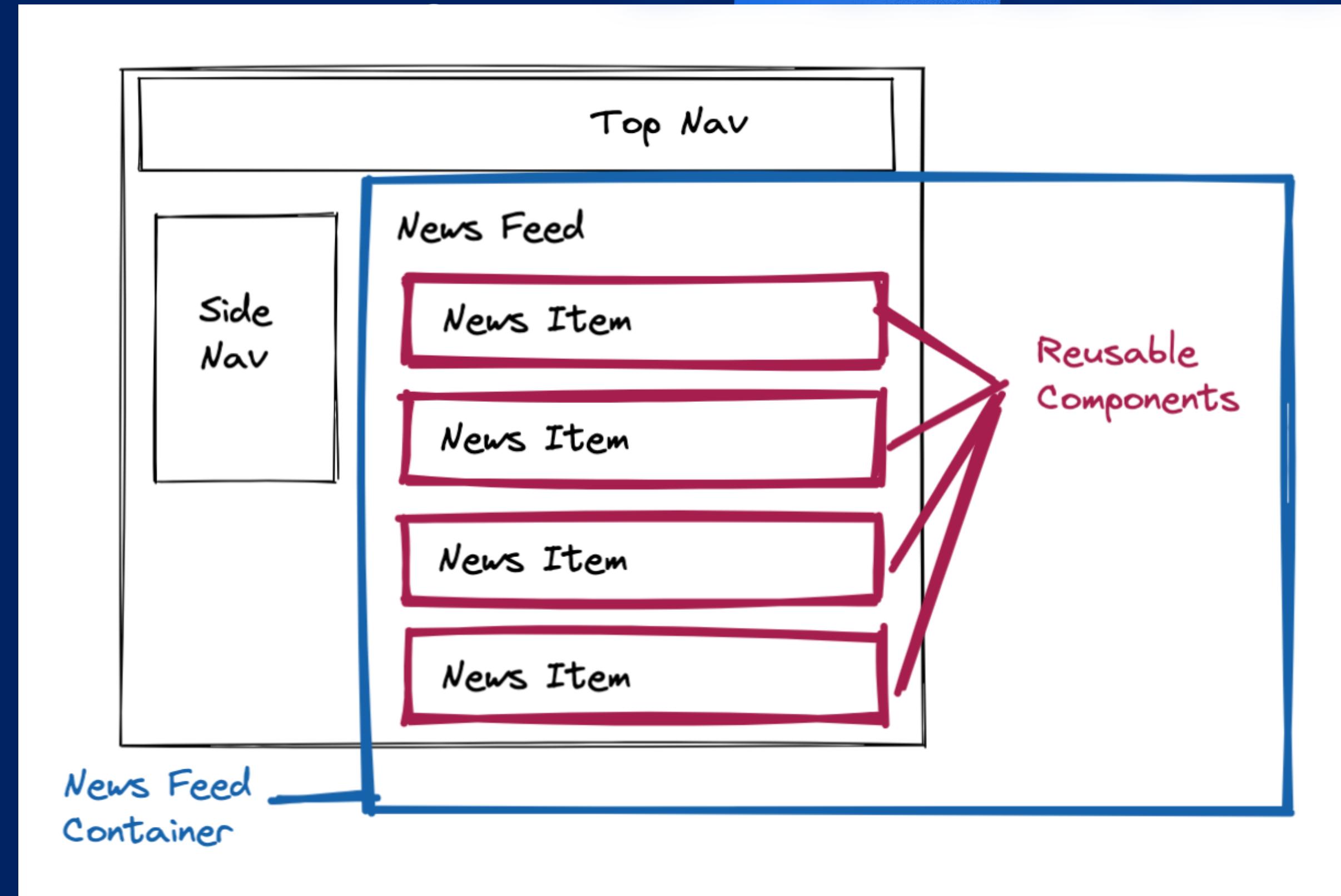
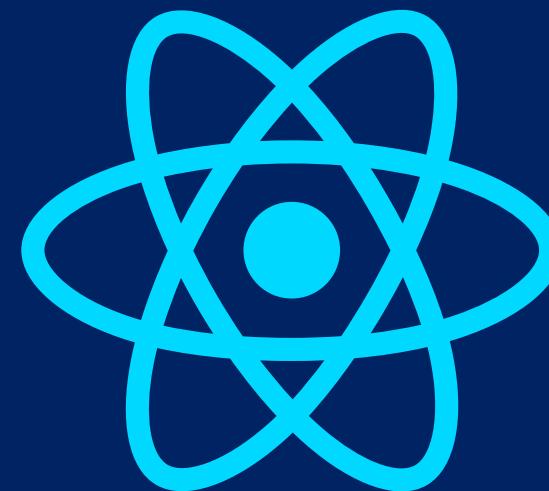
Steve created a group. 2:59pm

Infinite Scroll

React.js created by
Facebook engineer
Jordan Walke (2011)



React Components



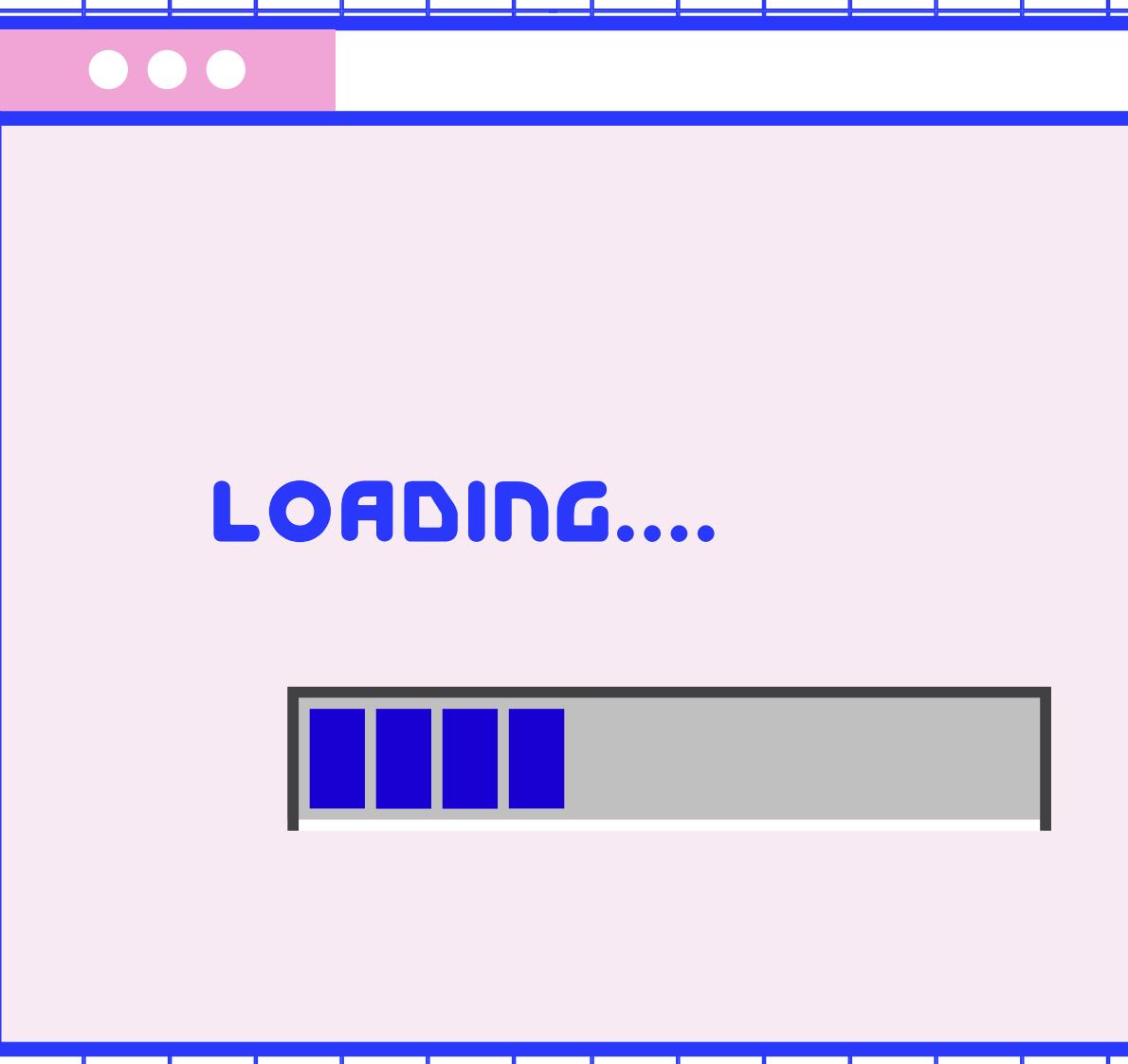


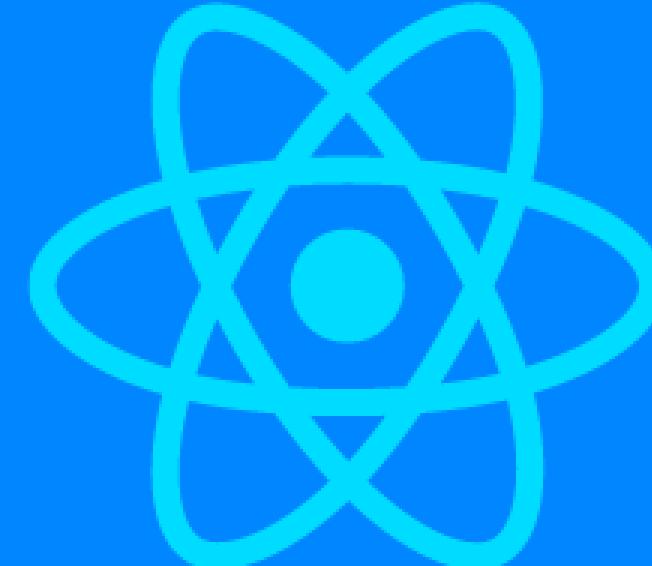
OK COMPUTER!

DEMO

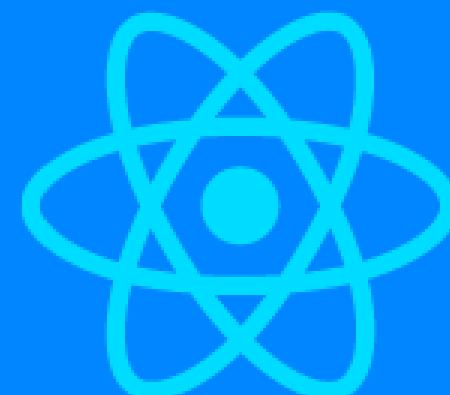
Building a Dockerfile for using React

CREATE-REACT-APP

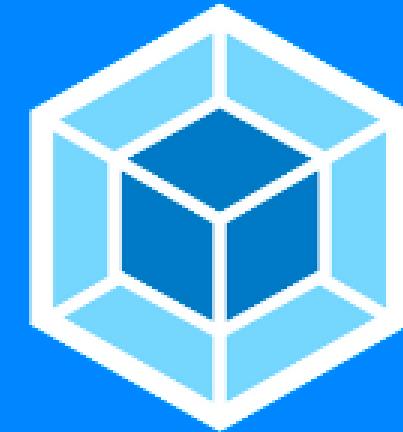




Create React App



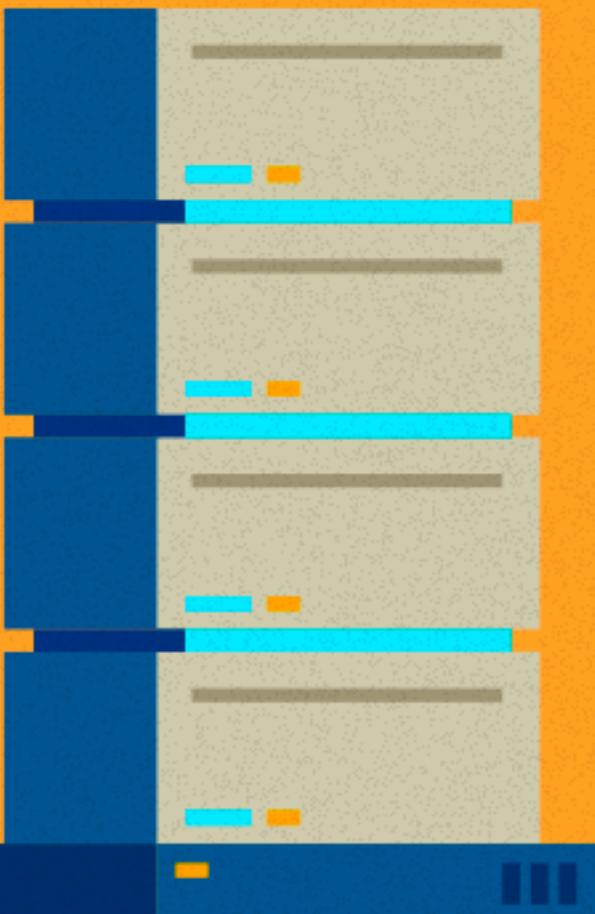
BABEL



webpack

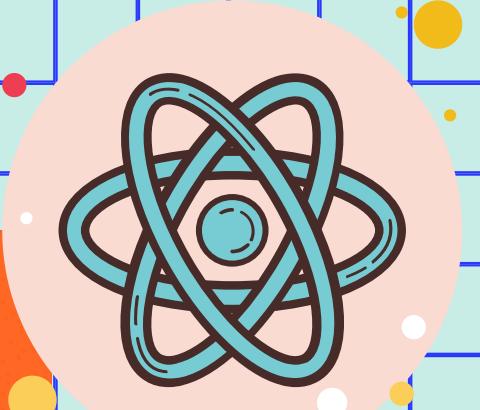
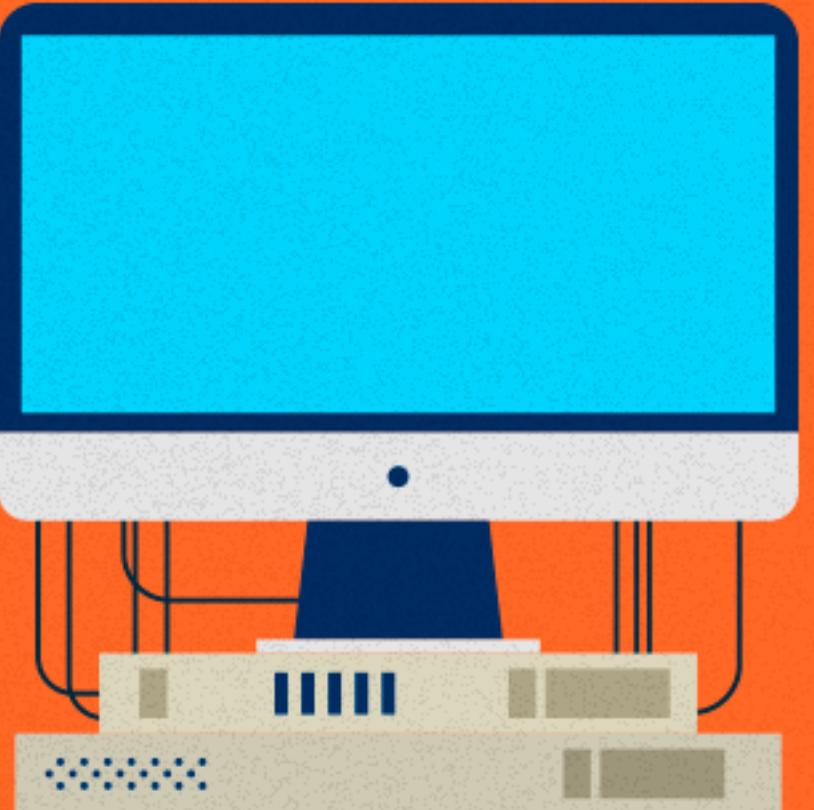
dj

SERVER
SIDE
RENDERING



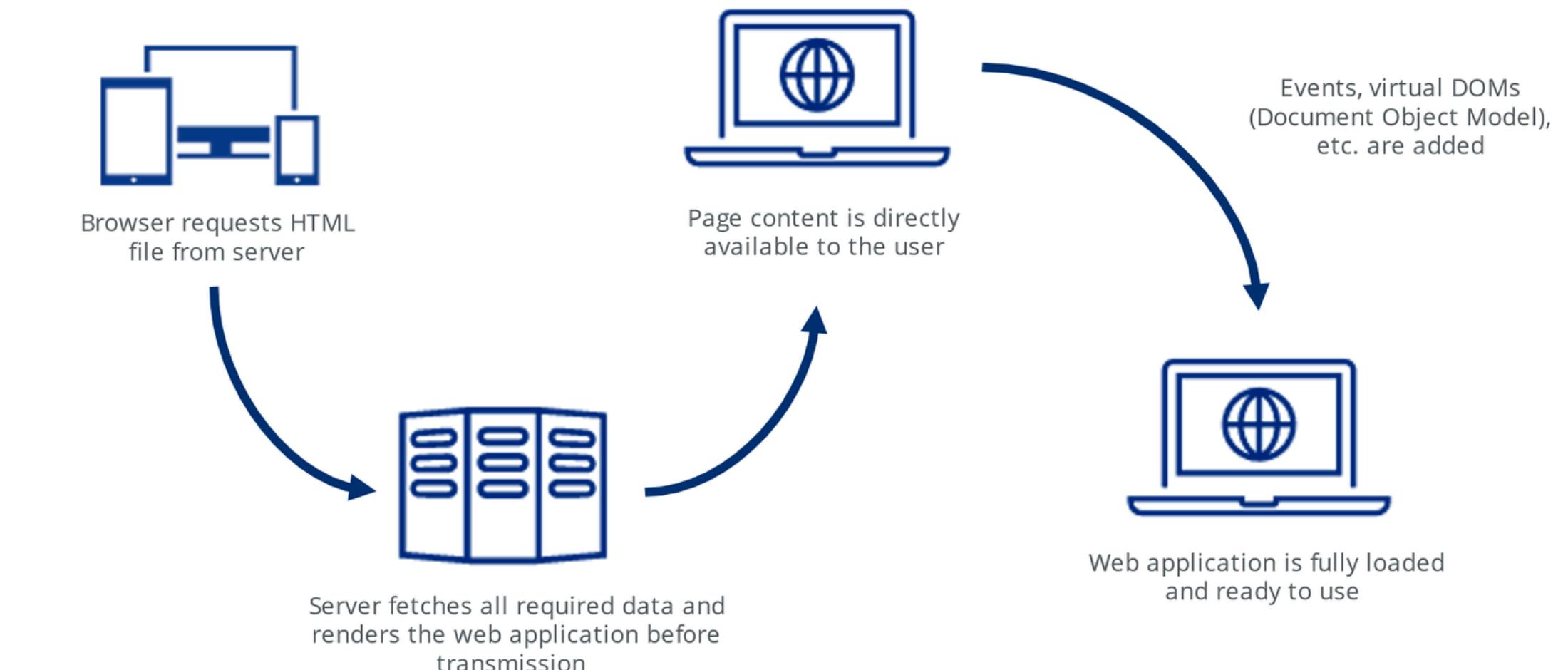
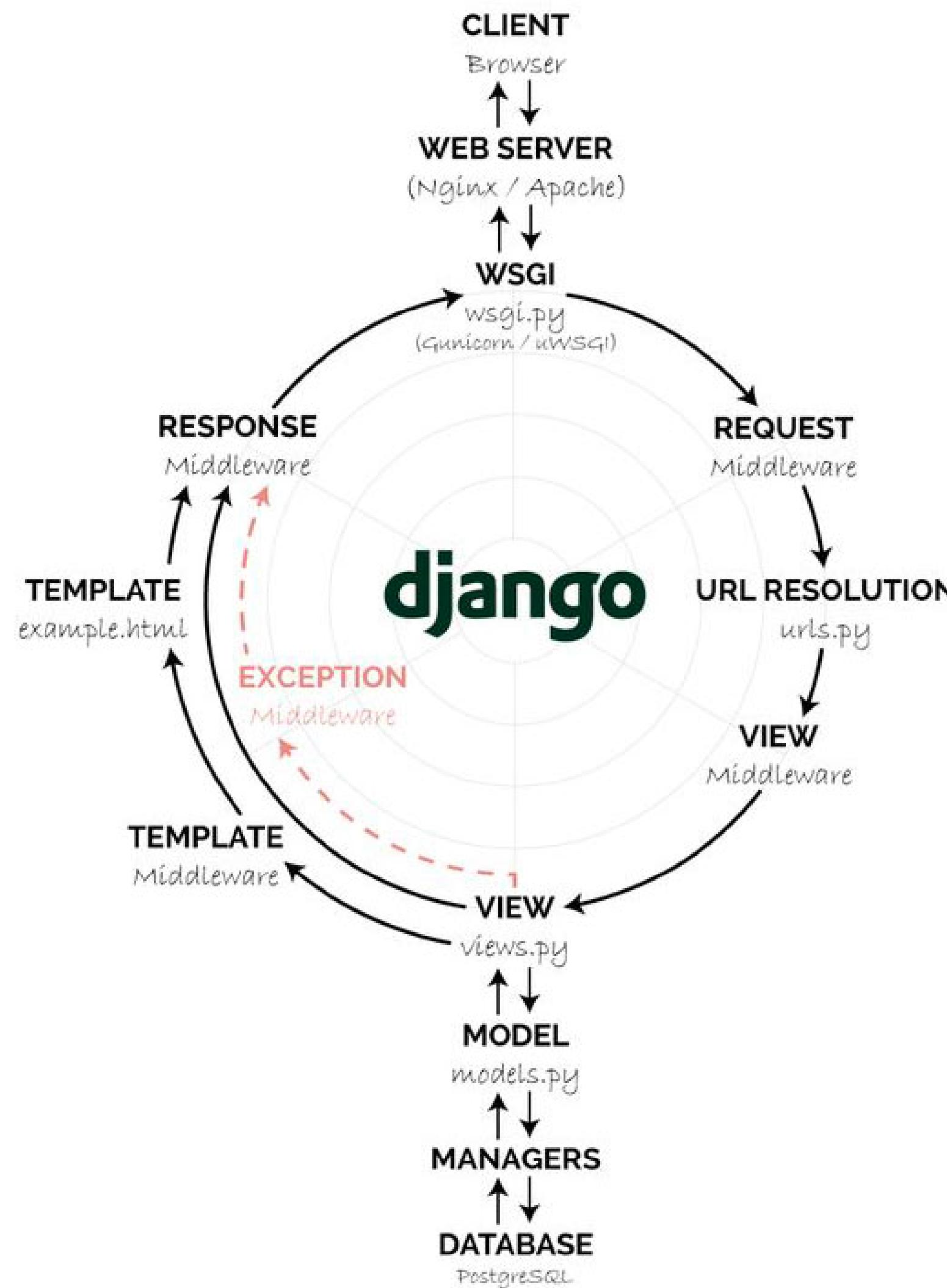
VS

CLIENT
SIDE
RENDERING

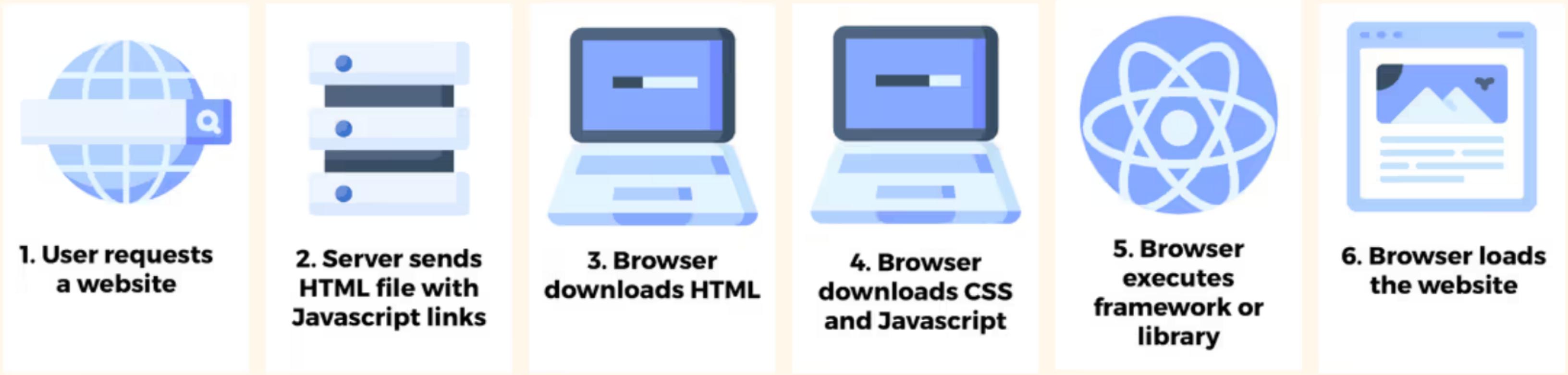
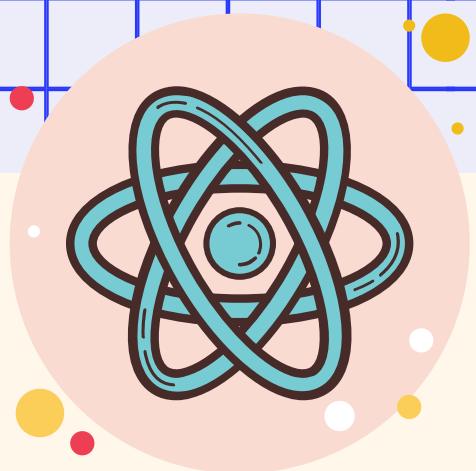




Server side rendering



Client Side Rendering



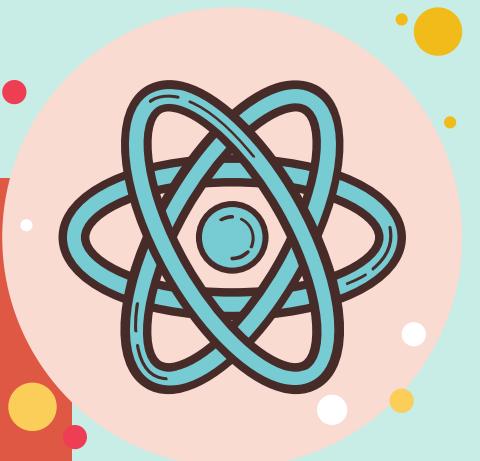


SSR

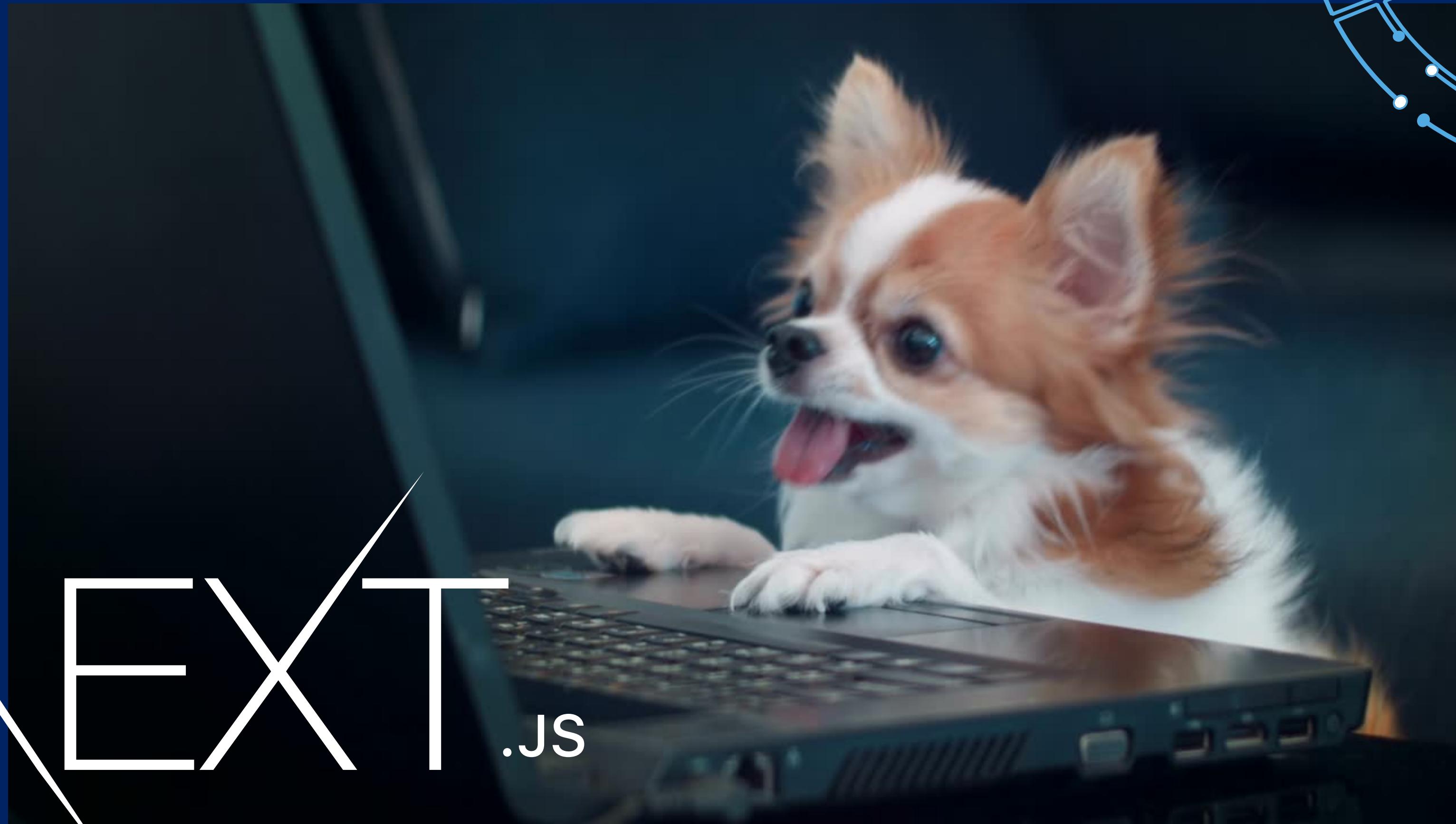
- + Ideal for static sites
- + Quick initial page access
- + No JS dependency
- + Easy-to-crawl site for better SEO
- Multiple server requests
- Full page reloads
- Non-rich site interactions
- Higher latency, prone to vulnerability

CSR

- + Ideal for web apps
- + Fast site rendering after initial load
- + Rich site interactions
- + Robust JS library selection
- Negative SEO for incorrect rendering & API response delays
- Slow initial load time
- Requires external library
- Higher memory consumption



~~NEXT~~.JS





OK COMPUTER!

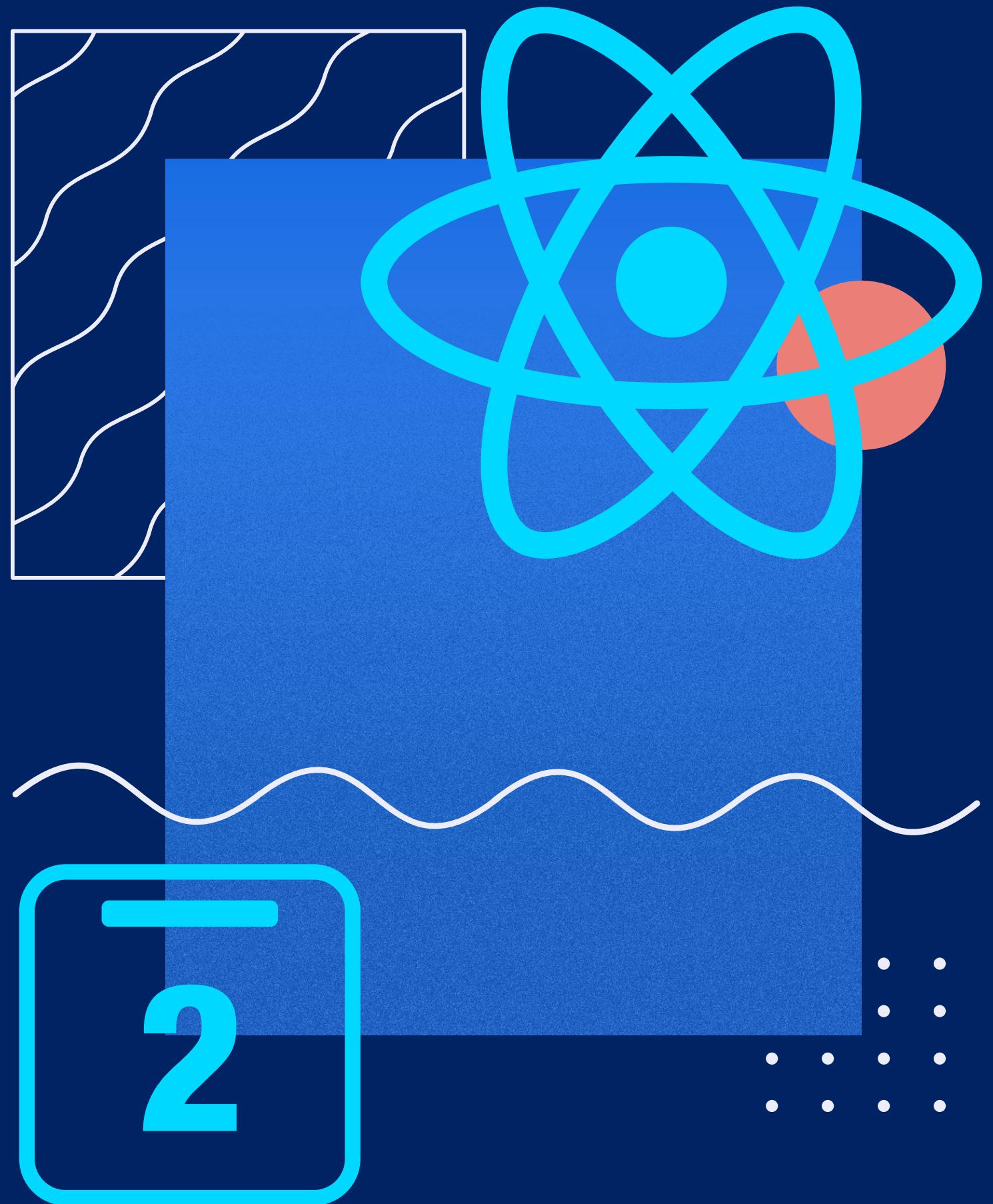
FUNCTIONAL COMPONENTS

Build a newsfeed.

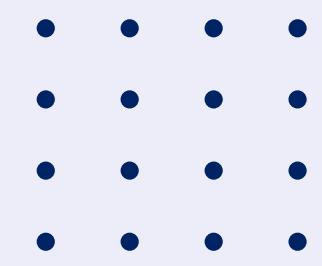
CREATE-REACT-APP

Stateful Components

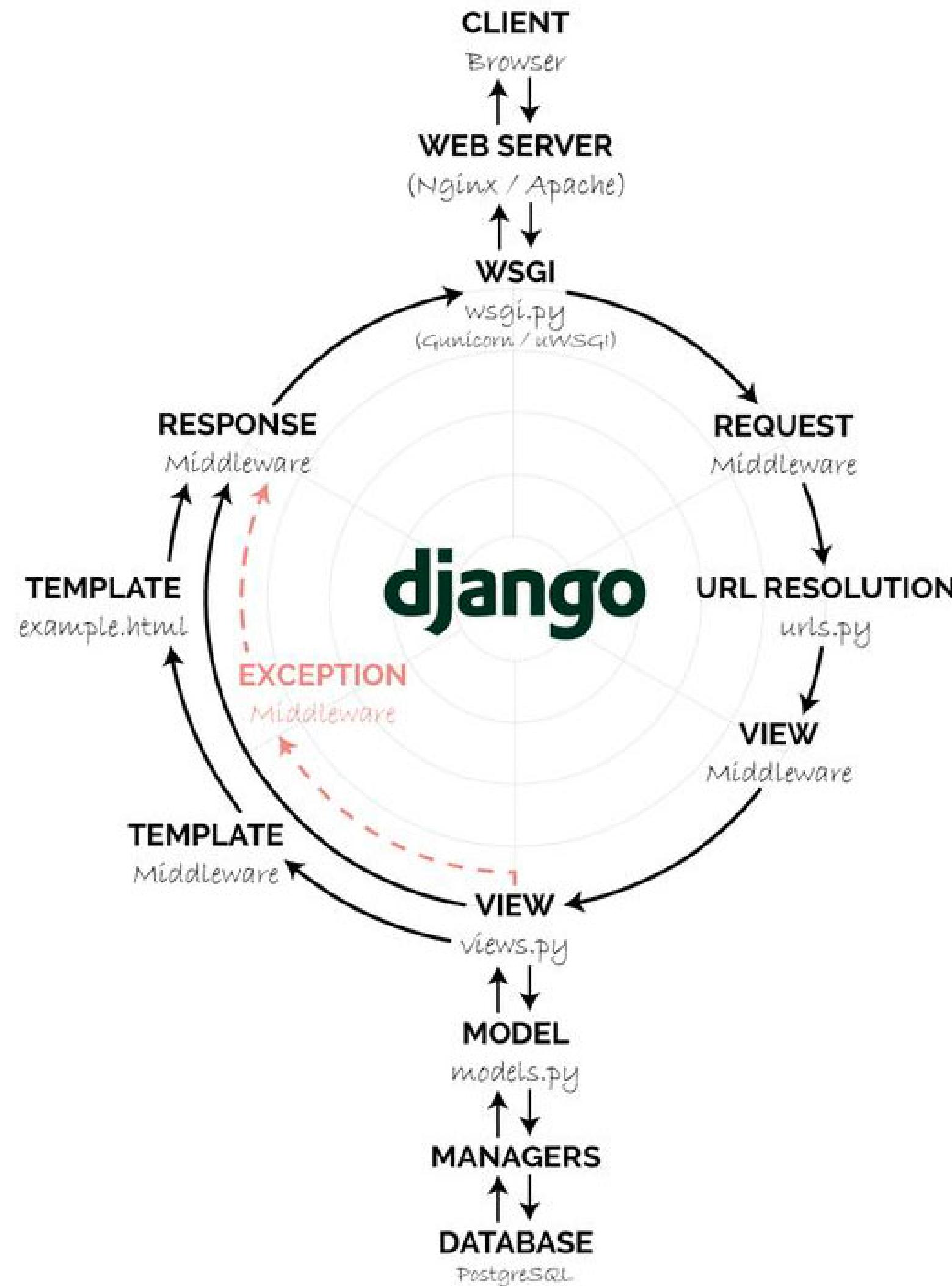
BUILDING A FORM IN REACT.



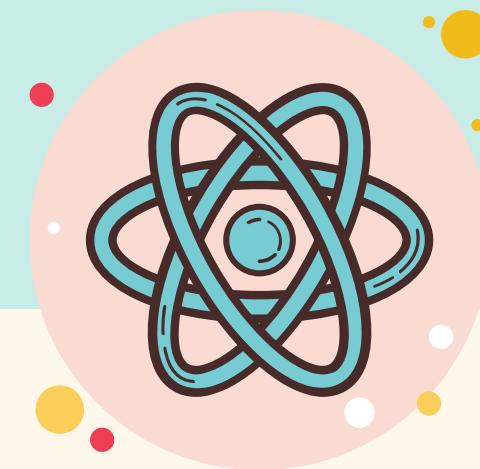
REVIEW



- What is React: front-end framework for creating modular, component-based UI.
- Docker + React
- How React integrates with Django
- Writing functional components



Front End



Client Side Rendering



Back End

Dockerfile

```
# Use the latest version of Node.js
FROM node:latest

# Use the /app directory as the work directory
WORKDIR /app

# Install the tool to create React apps
RUN npm install create-react-app
```

Terminal

```
docker build . -t react-docker
```

```
docker run -it -v "$(pwd):/app" -p 3000:3000 react-docker bash
```

```
npx create-react-app example  
cd example  
npm start
```

create-react-app

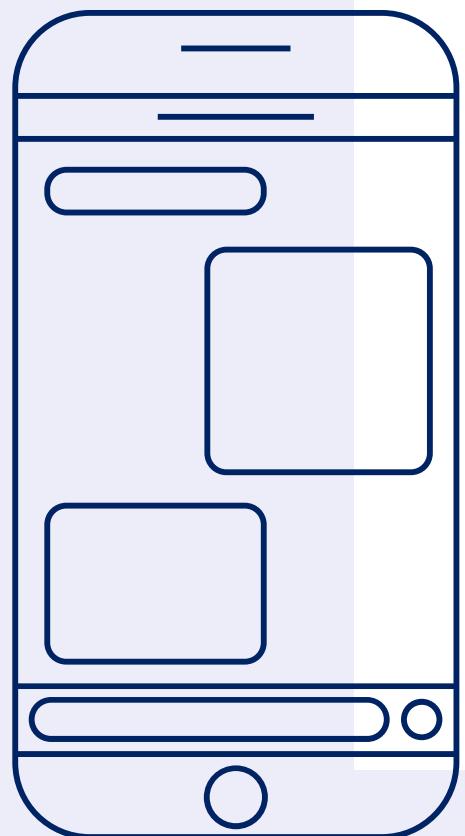
npm run build

npm run start

A screenshot of a file explorer window titled "DEMO". The tree view shows the following structure:

- > build
- ✓ public
 - ★ favicon.ico
 - ↳ index.html
 - 🖼 logo192.png
 - 🖼 logo512.png
 - { manifest.json
 - ≡ robots.txt
- ✓ src
 - # App.css
 - JS App.js
 - JS App.test.js
 - # index.css
 - JS index.js
 - ♫ logo.svg
 - JS setupTests.js
 - ↳ .gitignore
 - { package-lock.json
 - { package.json
 - ⓘ README.md

The News Feed



facebook Welcome Matt!

home search browse share invite help logout

Requests 1 group invitation

My Status edit

Keep your friends updated on your current status.

Shared with You see all

ShowBizSpy - Movies From Eddie Lim

Microsoft launches the Zune! - Engadget From Soleio

Drew From Soleio

Birthdays see all

Today's Birthdays Kyle

October 7th Birthdays Joe Alisha Nick

October 8th Birthdays

News Feed

Sponsored: iTunes

Athalie and Meredith commented on the note tab energy: an anomaly in my life. 3:59pm

Carrie joined the group Marvel: Ultimate Alliance – The World's Largest Army of Super Heroes. 3:39pm

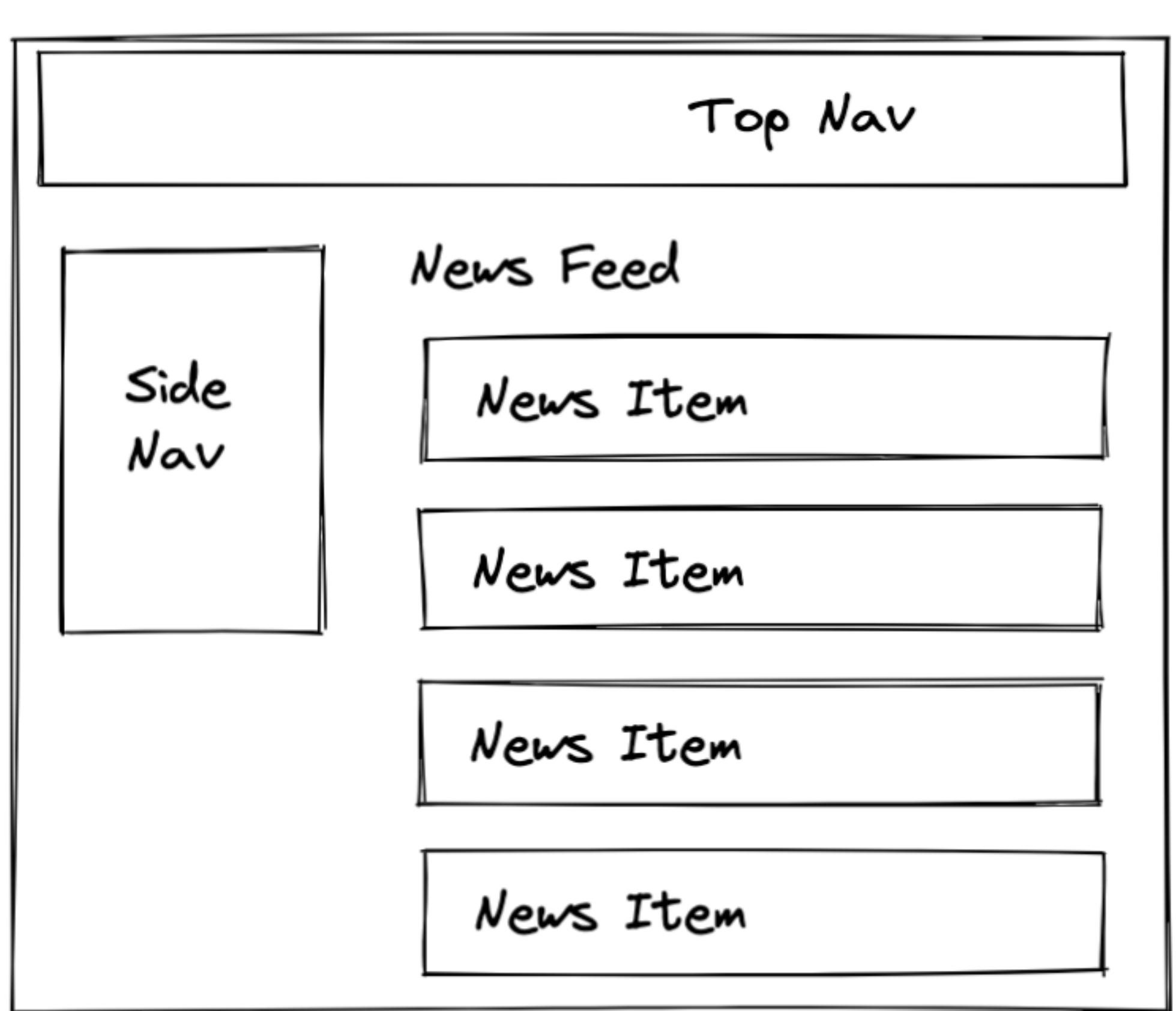
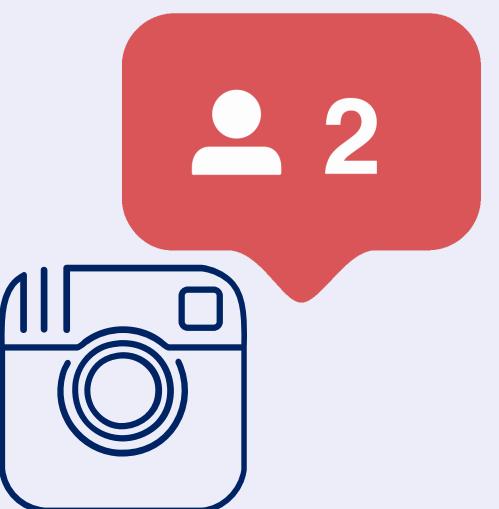
Payam Imani added new photos. 3:30pm Share Save

Gotham City: Chicago – 58 photos by Payam Location: Chicago, IL In this album: Payam

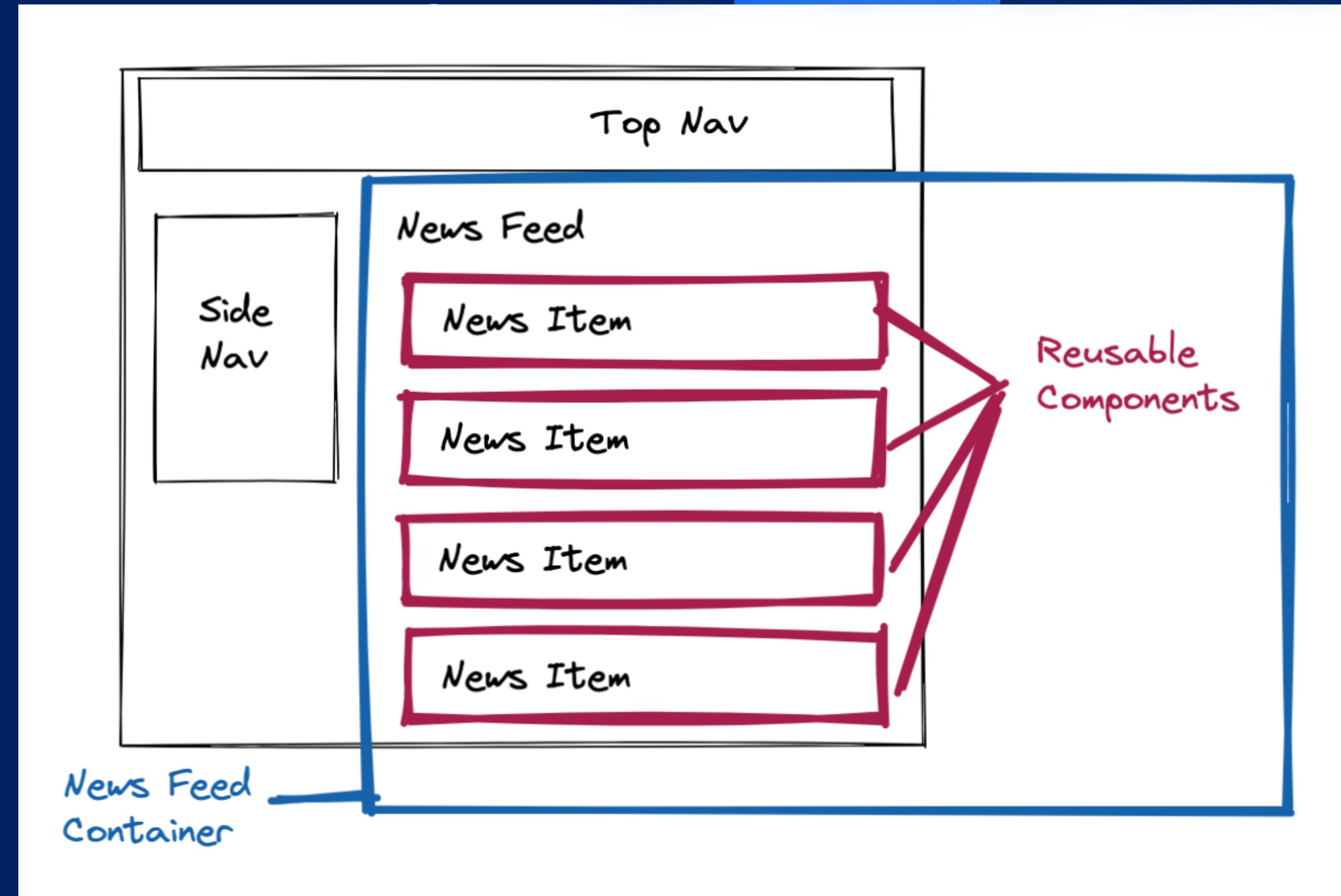
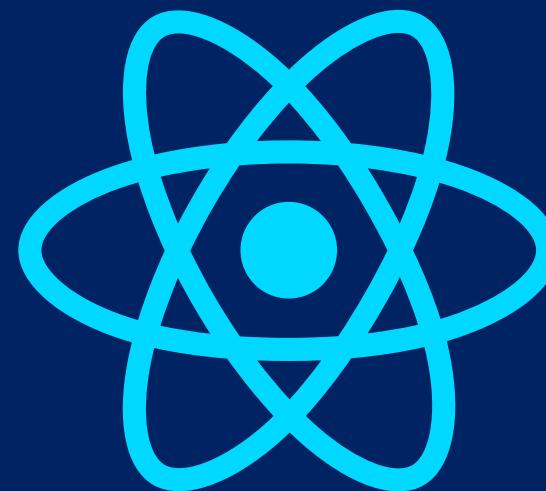
Steve created a group. 2:59pm

Infinite Scroll

React.js created by
Facebook engineer
Jordan Walke (2011)



React Components



Component Hierarchy

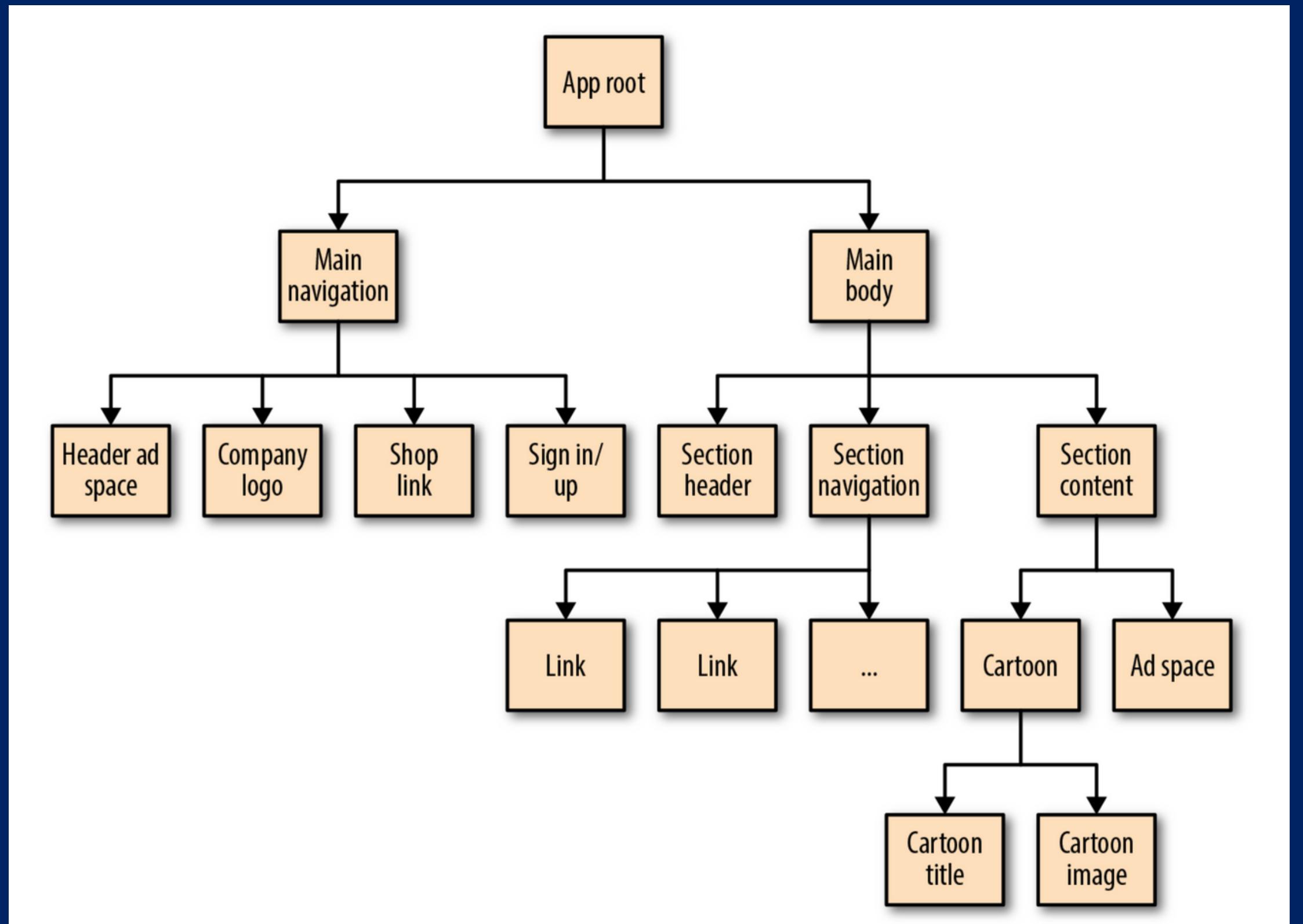


Image Source: **O'Reilly**
(<https://www.oreilly.com/library/view/what-react-is/9781491996744/ch01.html>)



Passing Props

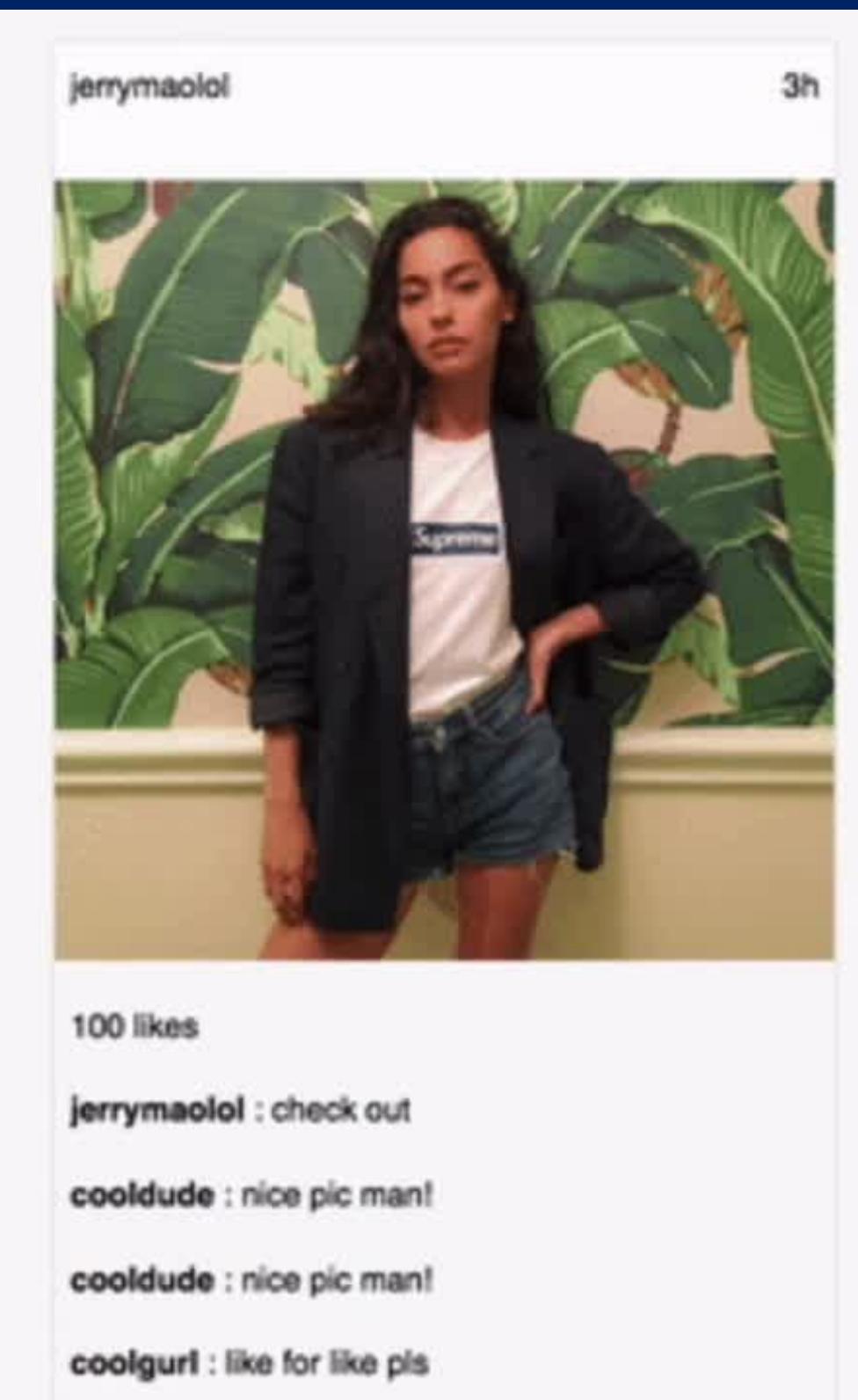
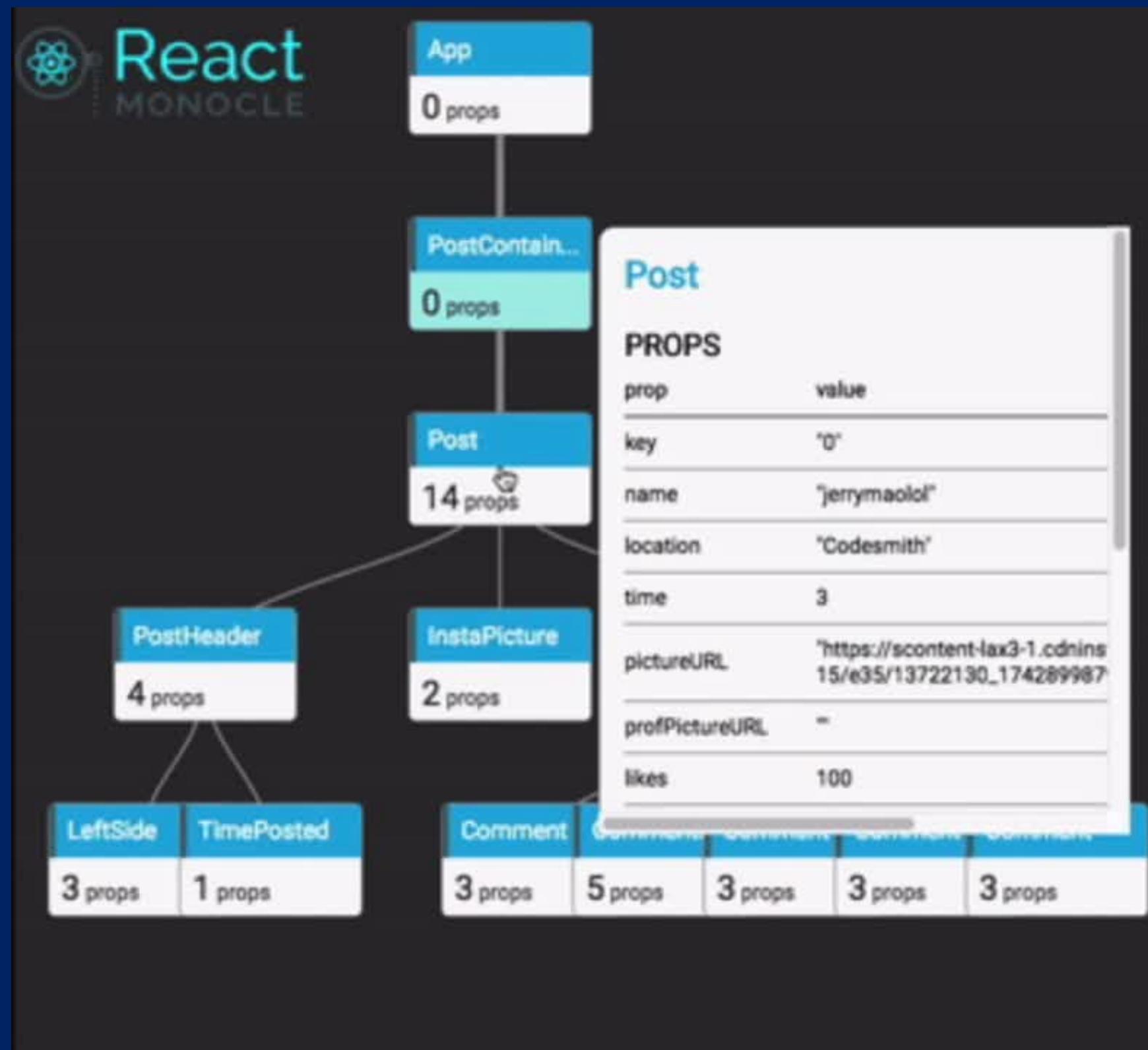


Image Source: **React Monocle**
(<https://github.com/team-gryff/react-monocle>)

PROPS

Props are data passed from a parent component to a child component.

Props are passed via HTML attributes and are used in React components as arguments.

Props are immutable.

Passing Props

Parent component (NewsFeed.jsx)

```
<div className='NewsFeed'>
  {data.map((item, index) => {
    return(
      <NewsItem key={index} name={item['name']} action={item['action']} />
    )
  )}
</div>
```

Passing Props

Child component (NewsItem.jsx)

```
1  function NewsItem(props) {
2      const {name, action} = props; <----- Javascript Object Destructuring
3      return(
4          <div className='NewsItem'>
5              <h3>{name}</h3>
6              <p>{action}</p>
7          </div>
8      );
9  }
10
11 export default NewsItem;
```

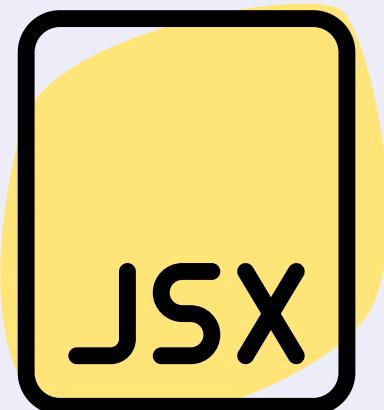
FUNCTIONAL COMPONENT

```
src > components > example.jsx > ...
1 import React from 'react'
2
3 export default function example() {
4     return (
5         <div>
6             <h1>Hello World</h1>
7         </div>
8     )
9 }
10
```



CLASS-BASED COMPONENT

```
class Clock extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {date: new Date()};  
  }  
  
  render() {  
    return (  
      <div>  
        <h1>Hello, world!</h1>  
        <h2>It is  
        {this.state.date.toLocaleTimeString()}</h2>  
      </div>  
    );  
  }  
}
```



STATE

The `state` object is a mutable store of data inside the component.

To change a value in the state object, use the `this.setState()` method.

When a value in the state object changes, the component will re-render.

Setting State

```
class HelloWorld extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      name: 'Woof! 🐶'  
    };  
  }  
  render() {  
    return (  
      <>  
        <h1>Hello, {this.state.name}</h1>  
        <input onChange={event => this.setState({name: event.currentTarget.value})}/>  
      </>  
    );  
  }  
  export default HelloWorld;
```

Event Handling

.onClick()
.onSubmit()

```
<div>
  <button onClick={this.showAlert}>Click Me</button>
</div>
```

Event **listener** names are camelCase instead of lowercase, and we pass a function in as the event handler, rather than a string.

```
showAlert(e) {
  e.preventDefault();
  alert("hello");
}
```

Binding "This"

```
this.handleChange = this.handleChange.bind(this);  
this.handleSubmit = this.handleSubmit.bind(this);
```

The default binding for "this" is the browser window. In order to avoid this possible problem at runtime, we explicitly bind the method to the object.

READ MORE HERE:

<https://www.freecodecamp.org/news/this-is-why-we-need-to-bind-event-handlers-in-class-components-in-react-f7ea1a6f93eb/>

Lifecycle Methods

A component “mounts” when it renders for the first time.

```
componentDidMount() {  
  fetchPosts().then(response => {  
    this.setState({  
      posts: response.posts  
    });  
  });  
  
  fetchComments().then(response => {  
    this.setState({  
      comments: response.comments  
    });  
  });  
}
```

1. Mounting
2. Updating
3. Unmounting

componentDidMount()
componentWillMount()
componentDidUpdate()
componentWillUnmount()

CLASS / FUNCTIONAL

```
import React from "react";

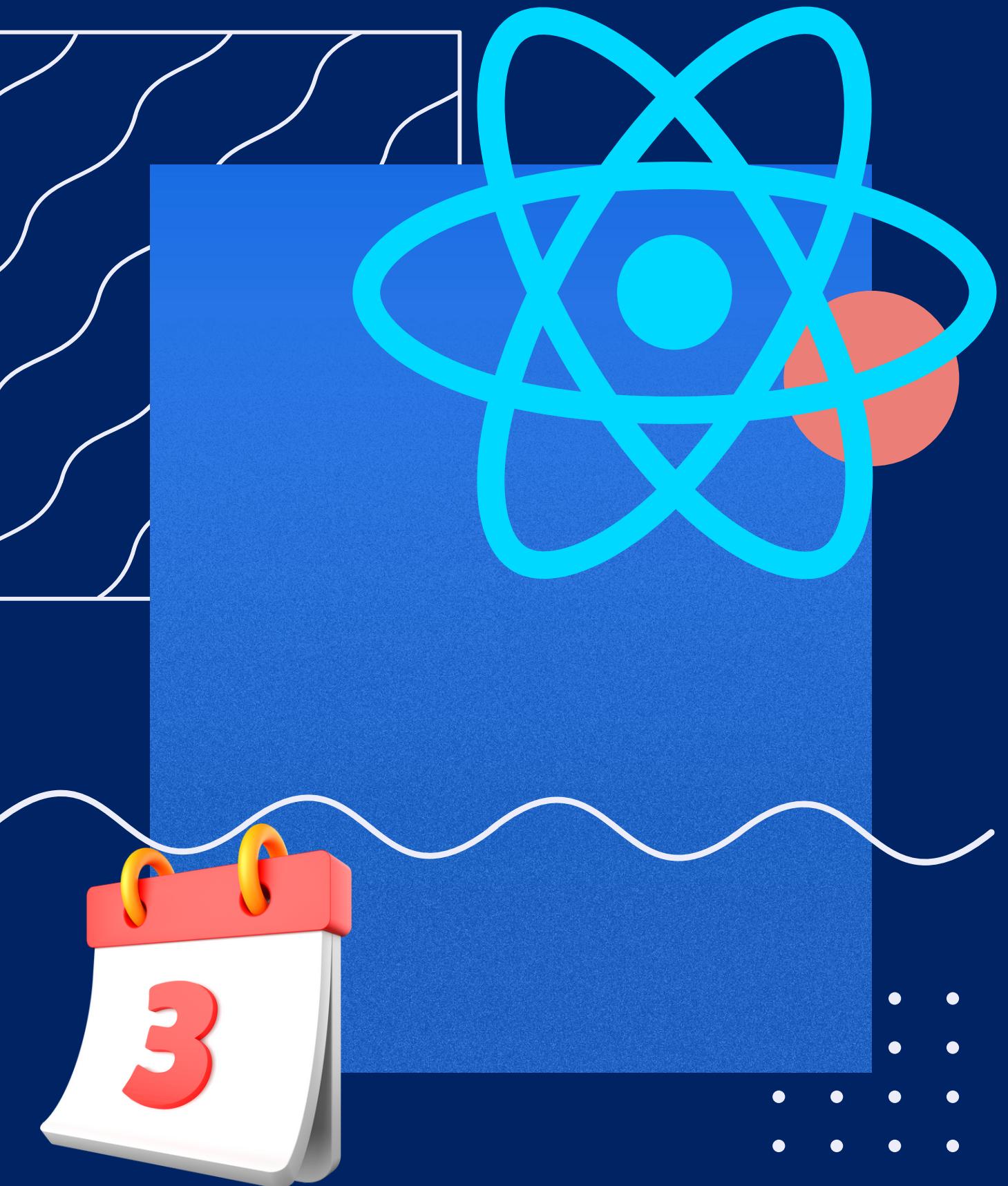
class ClassHelloWorld extends React.Component {
  render() {
    return <h1>Hello, World</h1>;
  }
}

const FunctionHelloWorld = function () {
  return <h1>Hello, World</h1>;
};

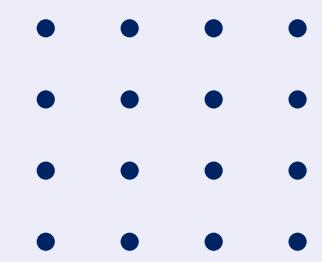
const ArrowHelloWorld = () => <h1>Hello, World</h1>;
```


React Router

BUILDING A SINGLE PAGE APPLICATION.



REVIEW



- Functional & Class-Based components
- Passing Props
- Setting State
- Event-handling
- Component Lifecycle

CLASS / FUNCTIONAL

```
import React from "react";

class ClassHelloWorld extends React.Component {
  render() {
    return <h1>Hello, World</h1>;
  }
}

const FunctionHelloWorld = function () {
  return <h1>Hello, World</h1>;
};

const ArrowHelloWorld = () => <h1>Hello, World</h1>;
```

PROPS

Props are data passed from a parent component to a child component.

Props are **immutable**.

Props are passed via **HTML attributes** and are used in React components as **arguments**.

STATE

The **State** object is a **mutable** store of data inside the component.

To change a value in the state object, use the **asynchronous** **this.setState()** method.

When a value in the state object changes, the component will re-render: **Component Lifecycle**.

Component Hierarchy

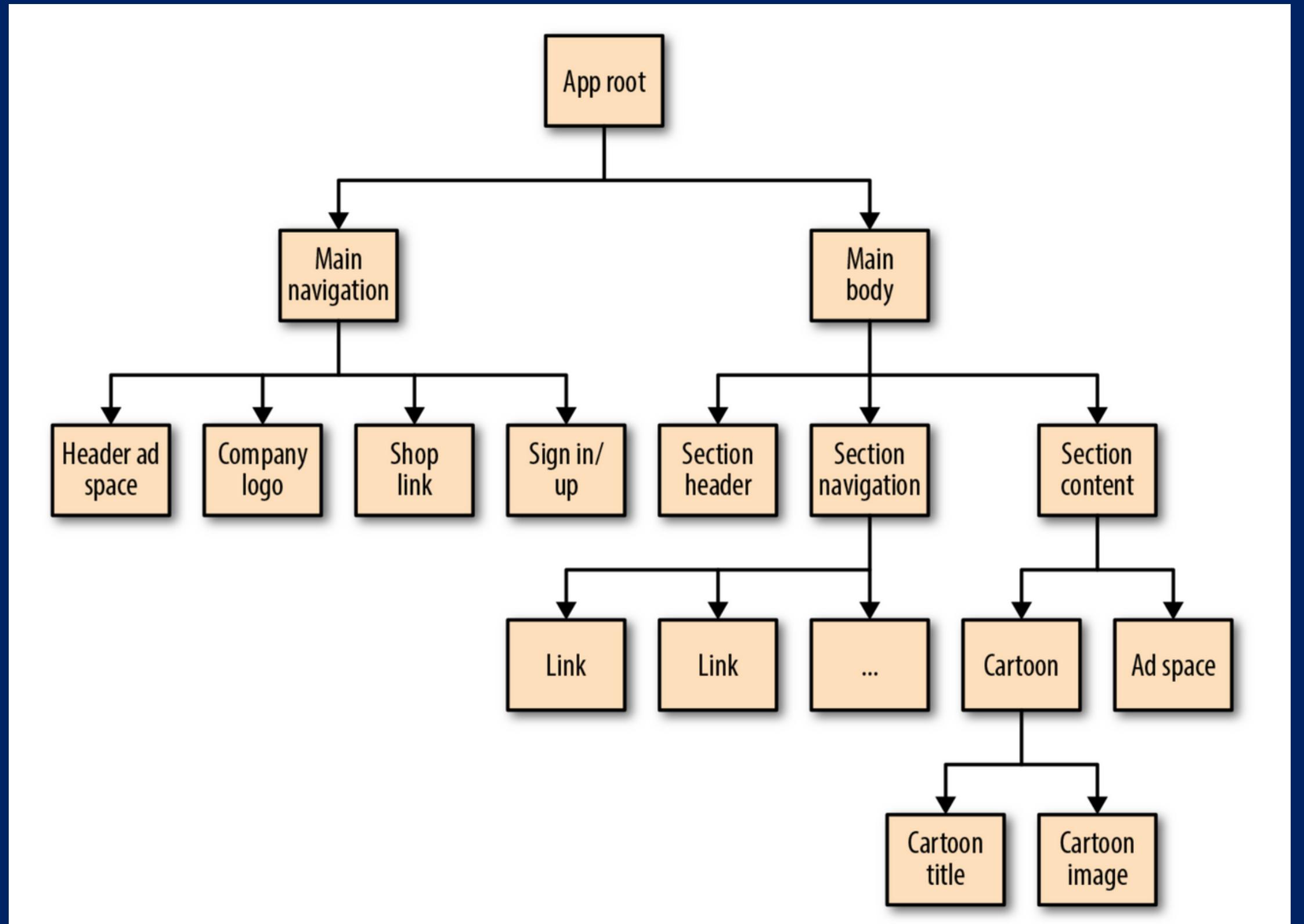


Image Source: **O'Reilly**
(<https://www.oreilly.com/library/view/what-react-is/9781491996744/ch01.html>)



Lifecycle Methods

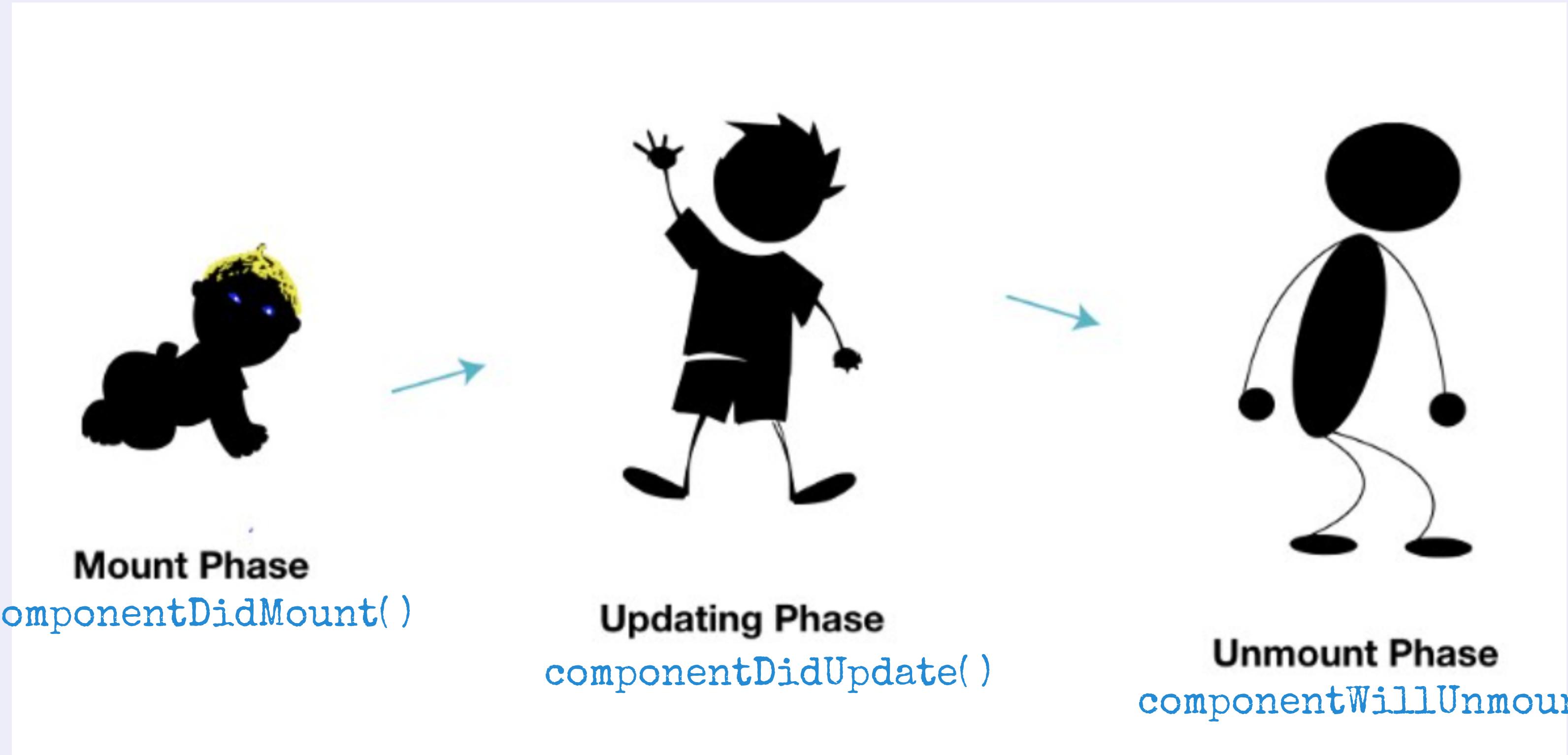
A component “**mounts**” when it renders for the first time.

```
componentDidMount() {  
  fetchPosts().then(response => {  
    this.setState({  
      posts: response.posts  
    });  
  });  
  
  fetchComments().then(response => {  
    this.setState({  
      comments: response.comments  
    });  
  });  
}
```

1. Mounting
2. Updating
3. Unmounting

componentDidMount()
componentWillMount()
componentDidUpdate()
componentWillUnmount()

Component Lifecycle



1

Mount Phase

When a component mounts to the DOM for the first time, four methods are called in order:

1. **constructor()**
2. **getDerivedStateFromProps()**
3. **render()**
4. **componentDidMount()**



Mount Phase

2

Updating Phase

Whenever there is a change to a component's state or props, the component is updated & re-rendered:



Updating Phase

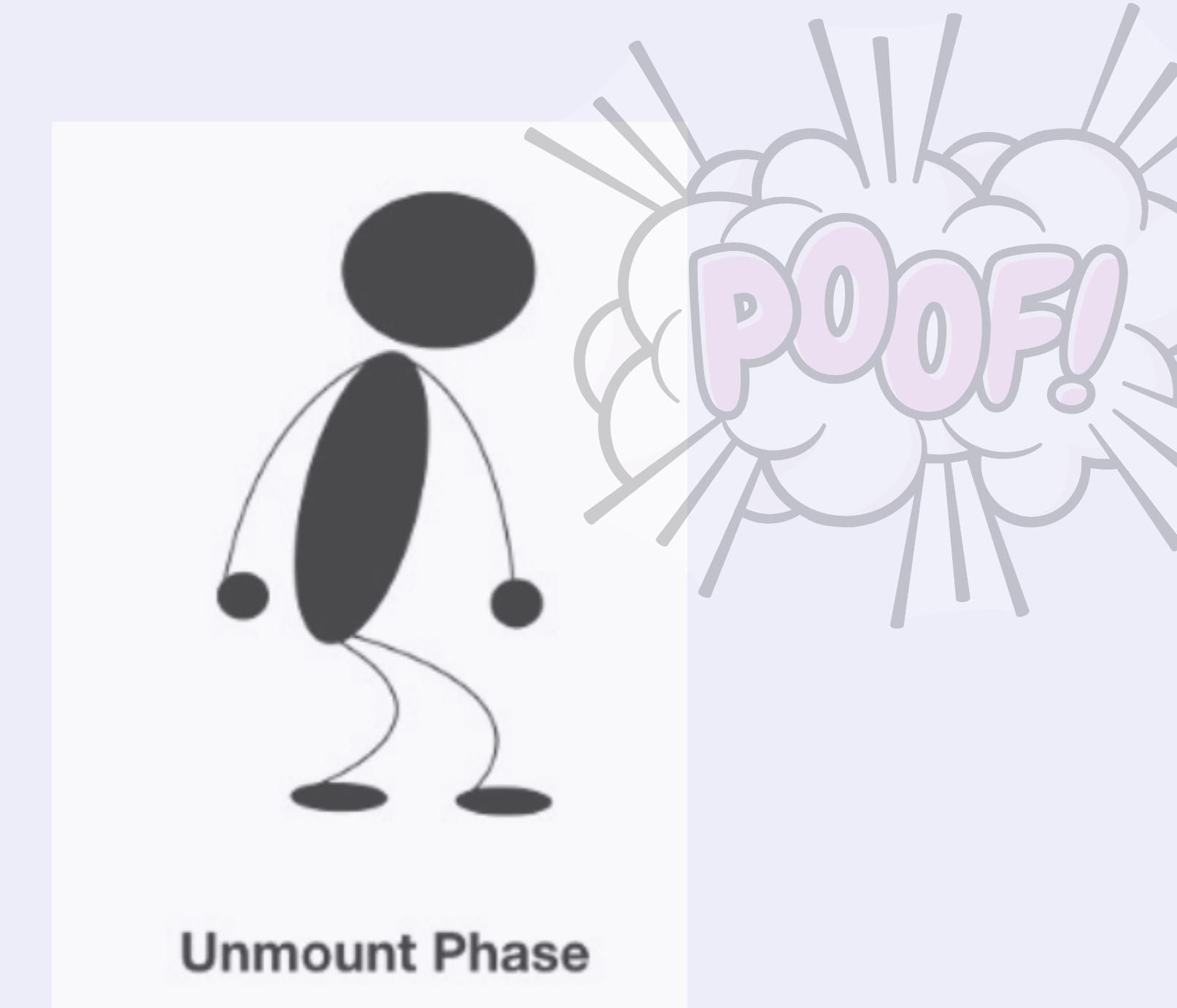
1. `getDerivedStateFromProps()`
2. `shouldComponentUpdate()`
3. `render()`
4. `getSnapshotBeforeUpdate()`
5. `componentDidUpdate()`

③

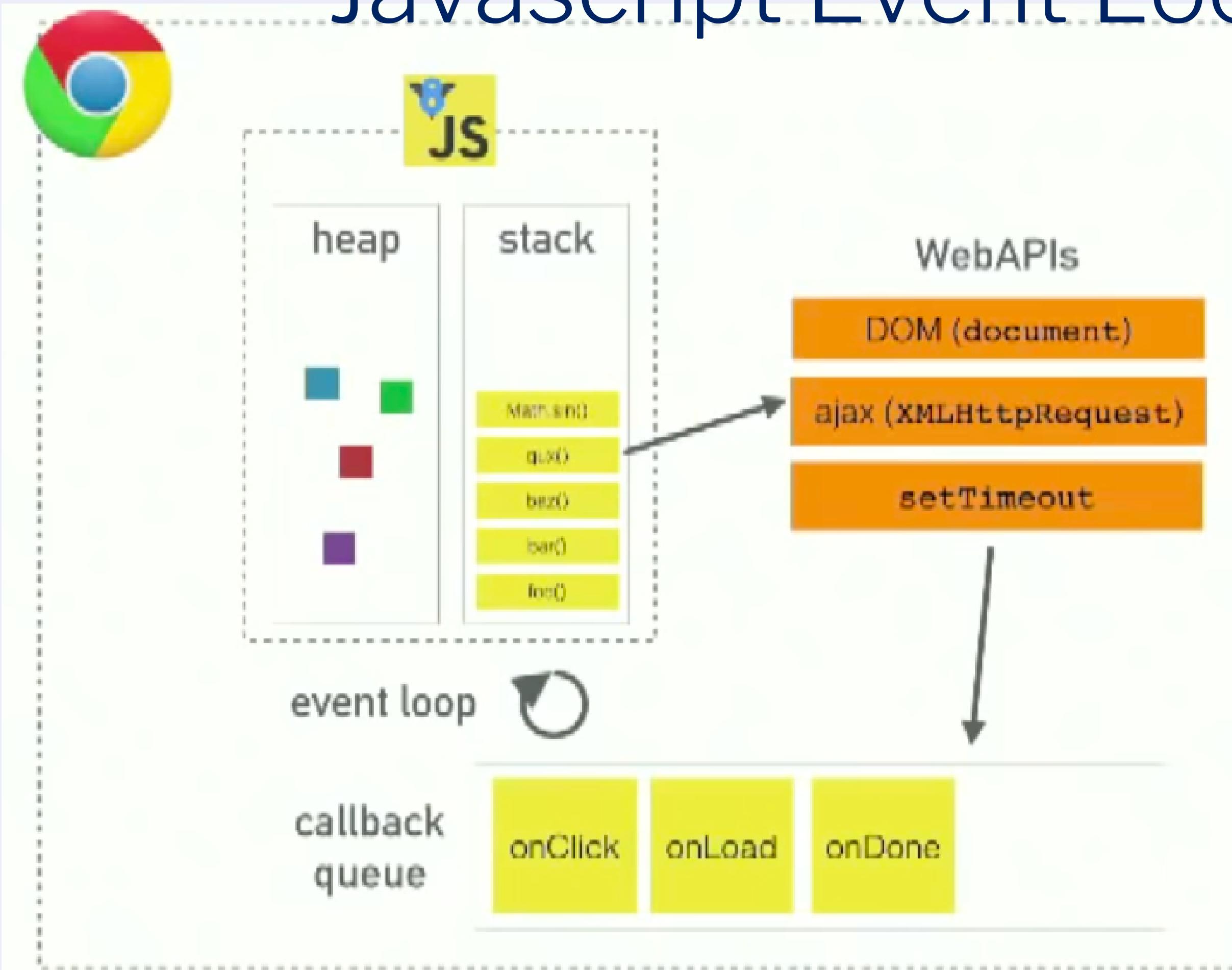
Unmount Phase

When a component is removed from the DOM, there's only one lifecycle method:

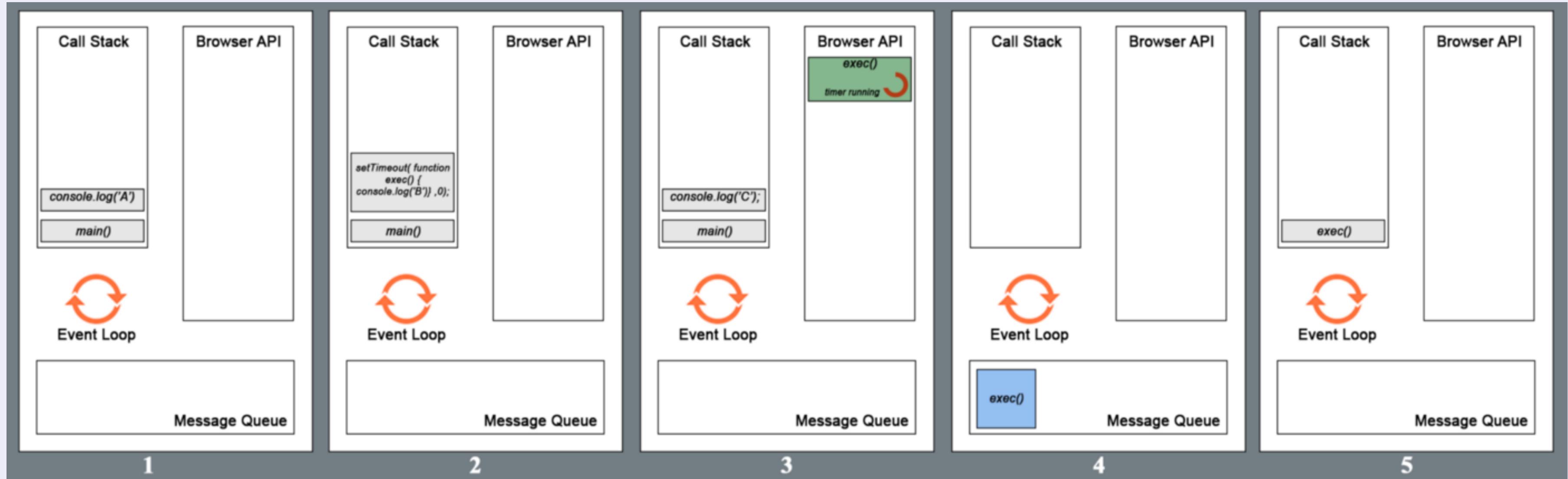
1. **componentWillUnmount()**



Javascript Event Loop



Javascript Execution Stack



.setState() is asynchronous!

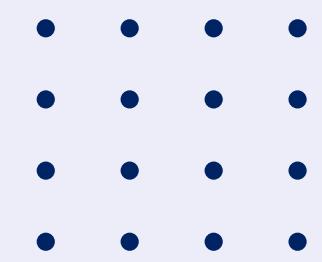
Pop Quiz!

```
class Counter extends Component {  
  constructor() {  
    this.state = {  
      counter: 0  
    }  
  }  
  render() { ... }  
}
```

What will we see in the console?

```
this.setState({ count: this.state.count + 1 });  
this.setState({ count: this.state.count + 1 });  
this.setState({ count: this.state.count + 1 });  
  
console.log(this.state.count);
```

REACT ROUTER

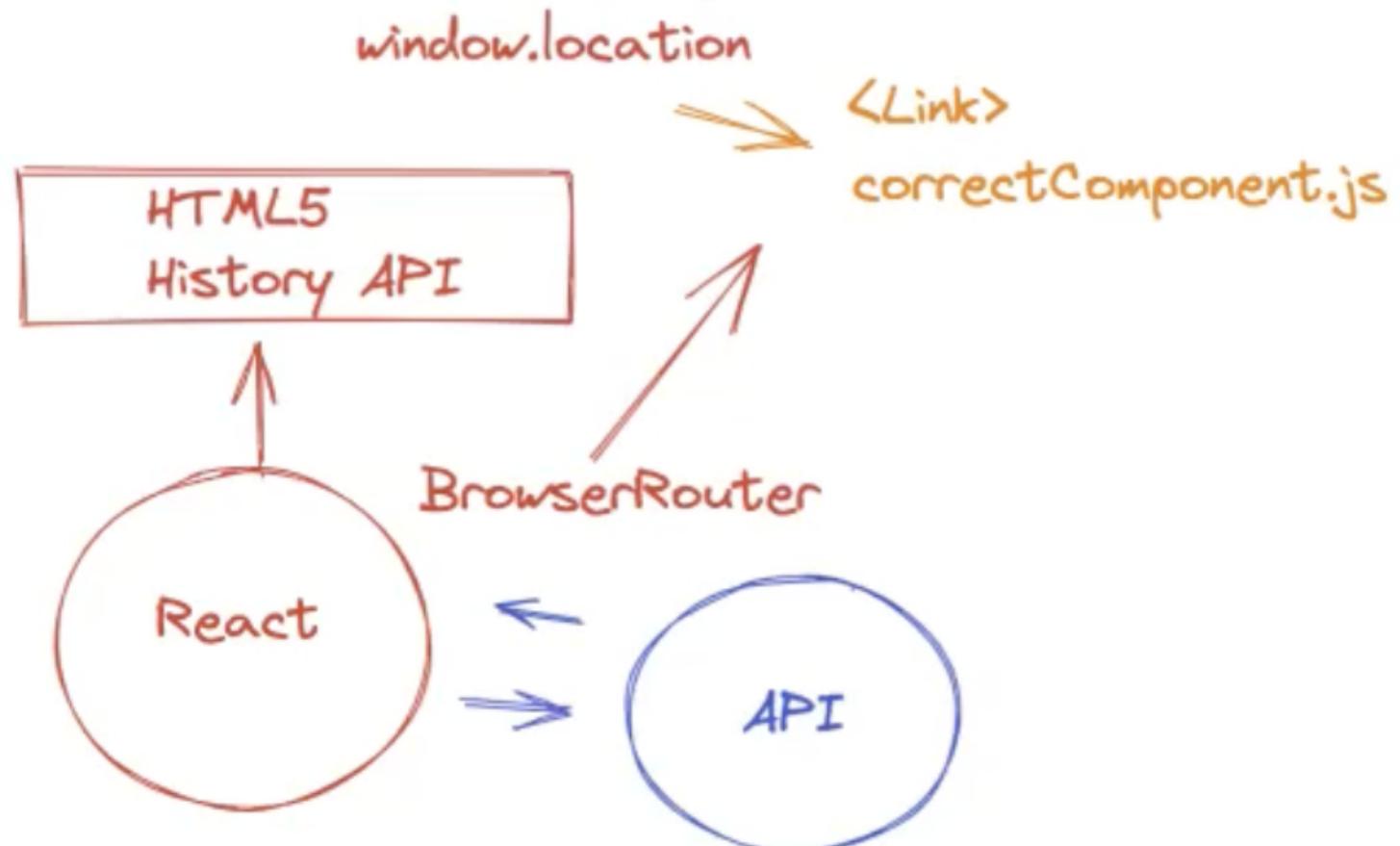


- Single Page Applications & Client-Side Rendering
- BrowserRouter
- Routes & Route
- Link
- NavLink
- Outlet

Routing!

SPA's - Single Page Applications(React.js)

Client Side Rendering!

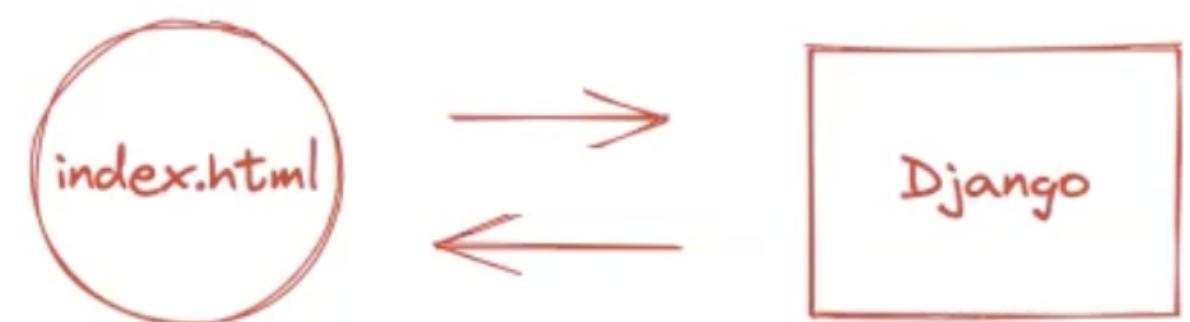


<Outlet> - needed
to render children
routes

<Routes>
<Route path=/home>
</Routes>

Server Side Rendering

Jinja templates
render(<h1>Hello World</h1>)

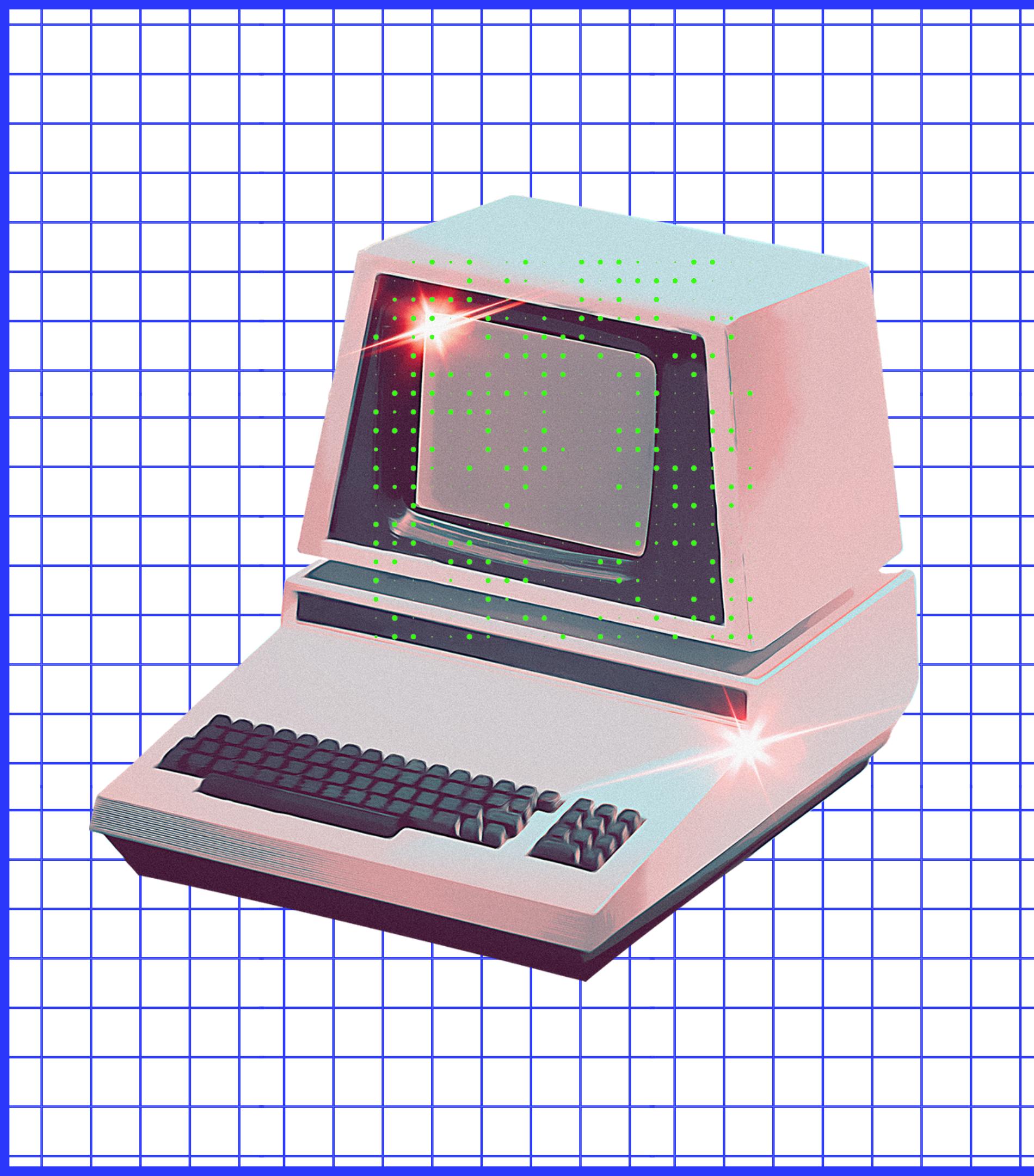


Ex-Calidraw: Paul Nnaji

React Router

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

Import libraries from
[react-router-dom](#)
v. 6.3.0



OK COMPUTER!

DEMO

Reading Through a New Code Base.

REACT ROUTER