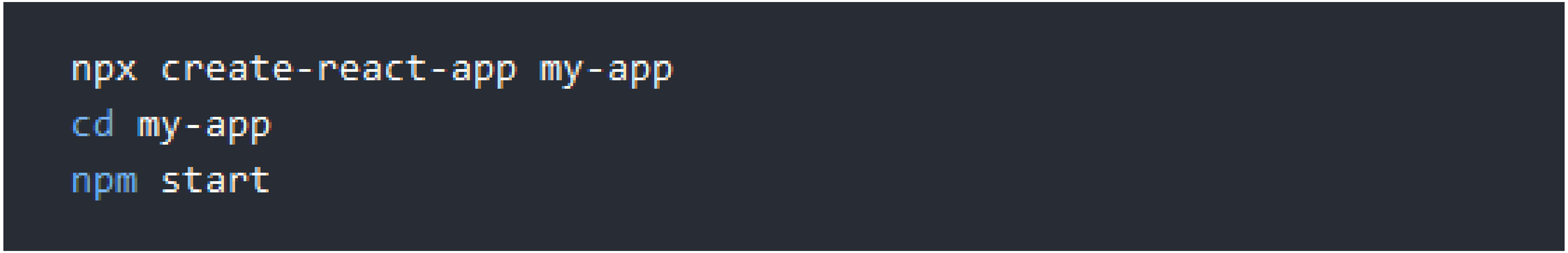
React

AJ

Create React App

Create React App is a comfortable environment for **learning React**, and is the best way to start building **a new single-page application** in React.

It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You’ll need to have Node >= 8.10 and npm >= 5.6 on your machine. To create a project, run:



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

Create-react-app

npx

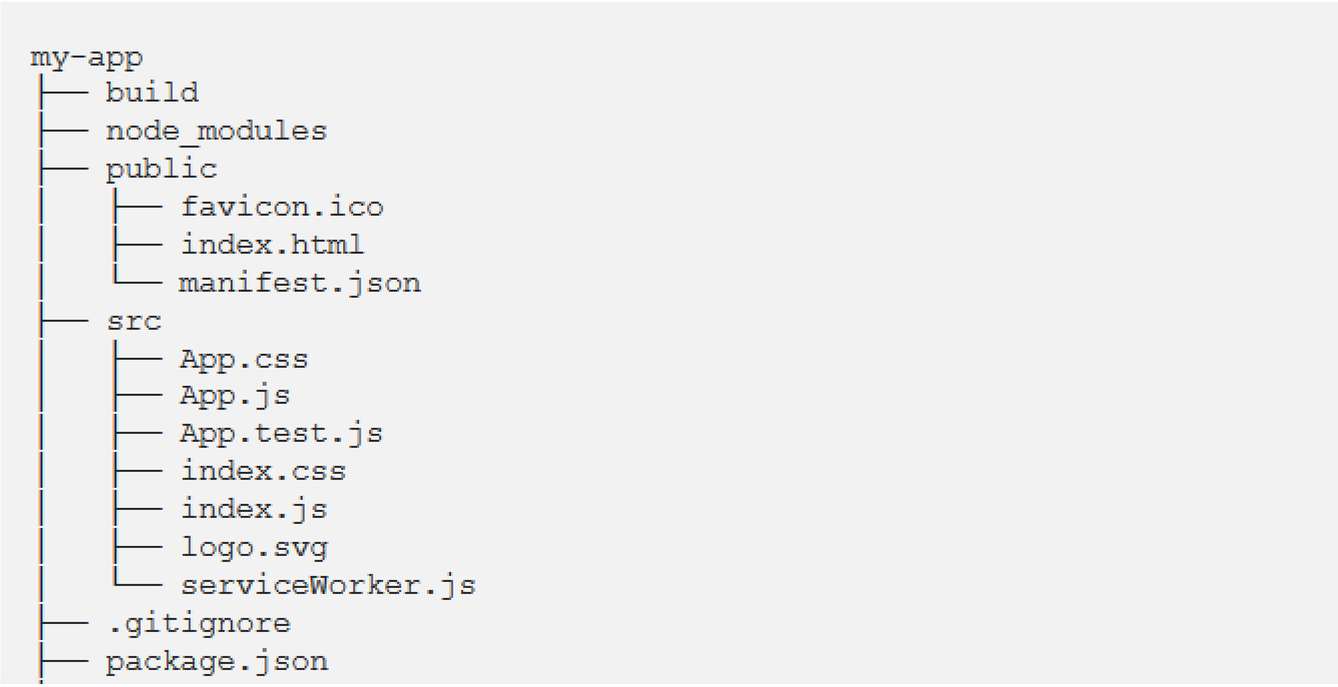
```
npx create-react-app <project_name>
```

npm package runner

npm

```
npm install create-react-app -g
```

```
create-react-app<project_name>
```



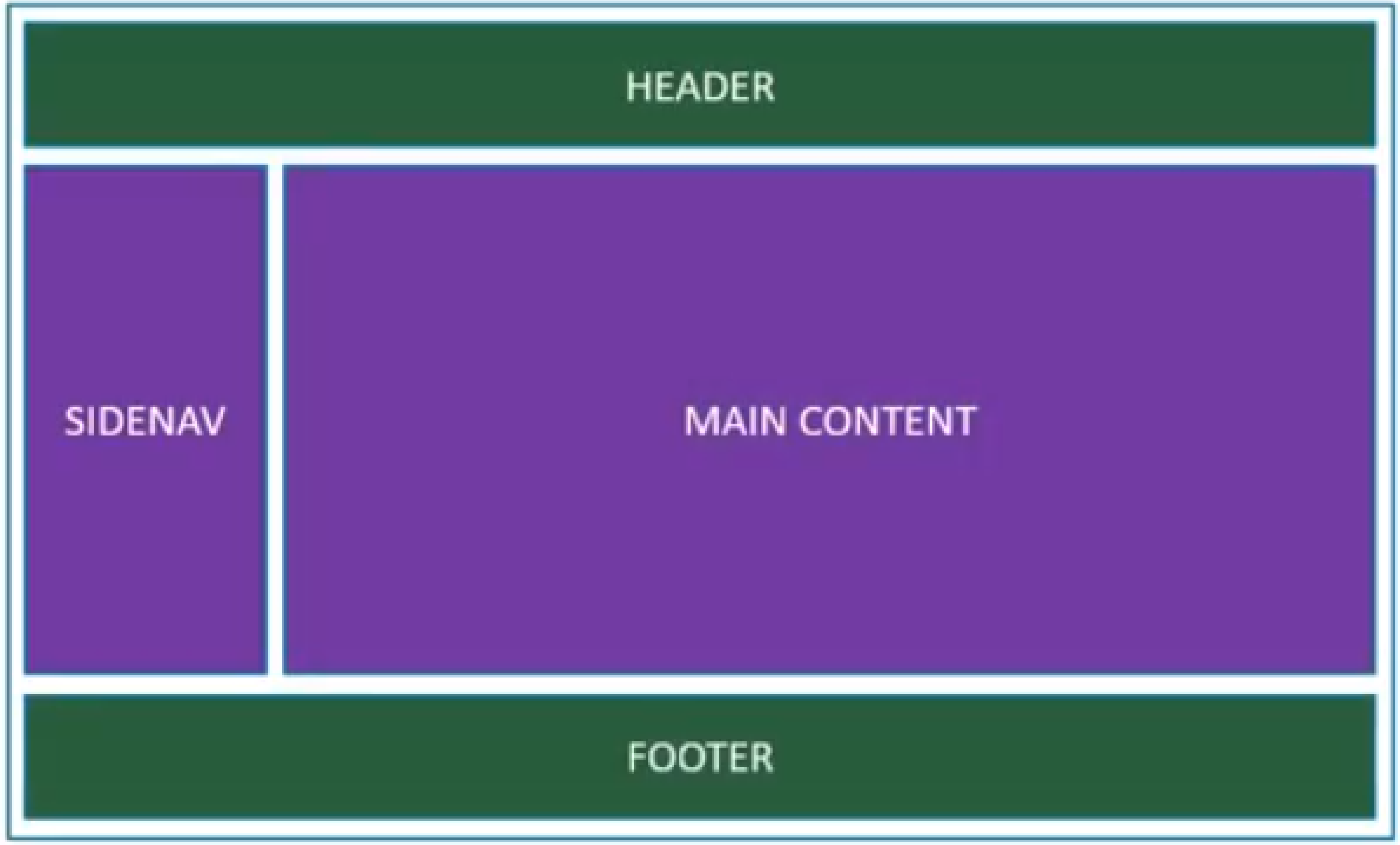
```
my-app
├── build
├── node_modules
├── public
│   ├── favicon.ico
│   ├── index.html
│   └── manifest.json
├── src
│   ├── App.css
│   ├── App.js
│   ├── App.test.js
│   ├── index.css
│   ├── index.js
│   ├── logo.svg
│   └── serviceWorker.js
├── .gitignore
├── package.json
```

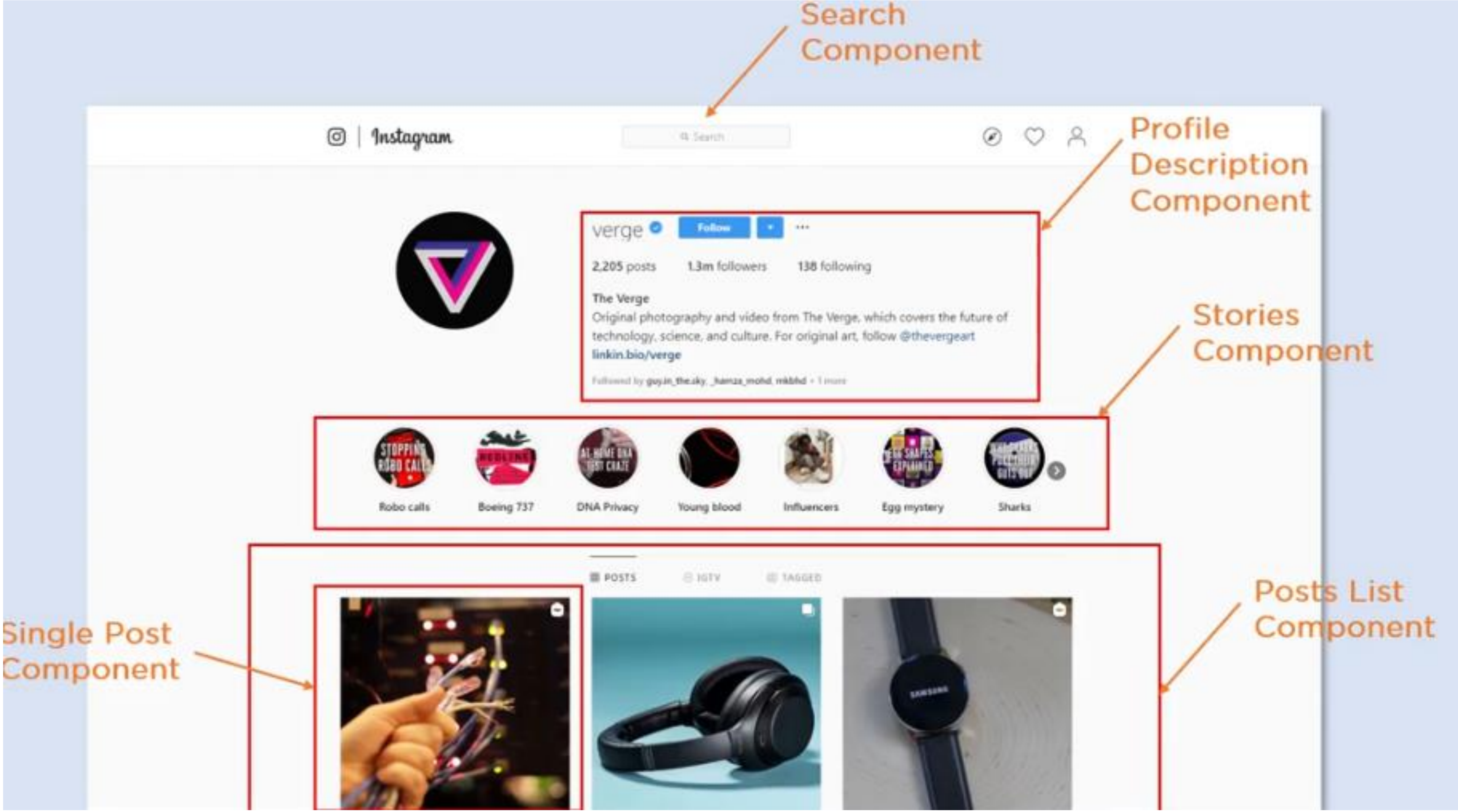
- `build` is the location of your final, production-ready build. This directory won't exist until you run `npm build` or `yarn build`. The contents of this folder should be ready-to-ship without any interaction on your part.

- `node_modules` is where packages installed by NPM or Yarn will reside.
- `public` is where your static files reside. If the file is not imported by your JavaScript application *and* must maintain its file name, put it here. Files in the `public` directory will maintain the same file name in production, which typically means that they will be cached by your client and never downloaded again. If your file does not have a filename that matters — such as `index.html`, `manifest.json`, or `robots.txt` — you should put it in `src` instead.

- `src` is where your dynamic files reside. If the file is imported by your JavaScript application or changes contents, put it here. In order to make sure the client downloads the most up-to-date version of your file instead of relying on a cached copy, Webpack will give changed files a unique file name in the production build. This allows you to use simple, intuitive file names during development, such as `banner.png` instead of `banner-2019-03-01-final.png`. You never have to worry about your client using the outdated cached copy, because Webpack will automatically rename `banner.png` to `banner.unique-hash.png`, where the unique hash changes only when `banner.png` changes.

Component Based Architecture





Search Component

Profile Description Component

Stories Component

Posts List Component

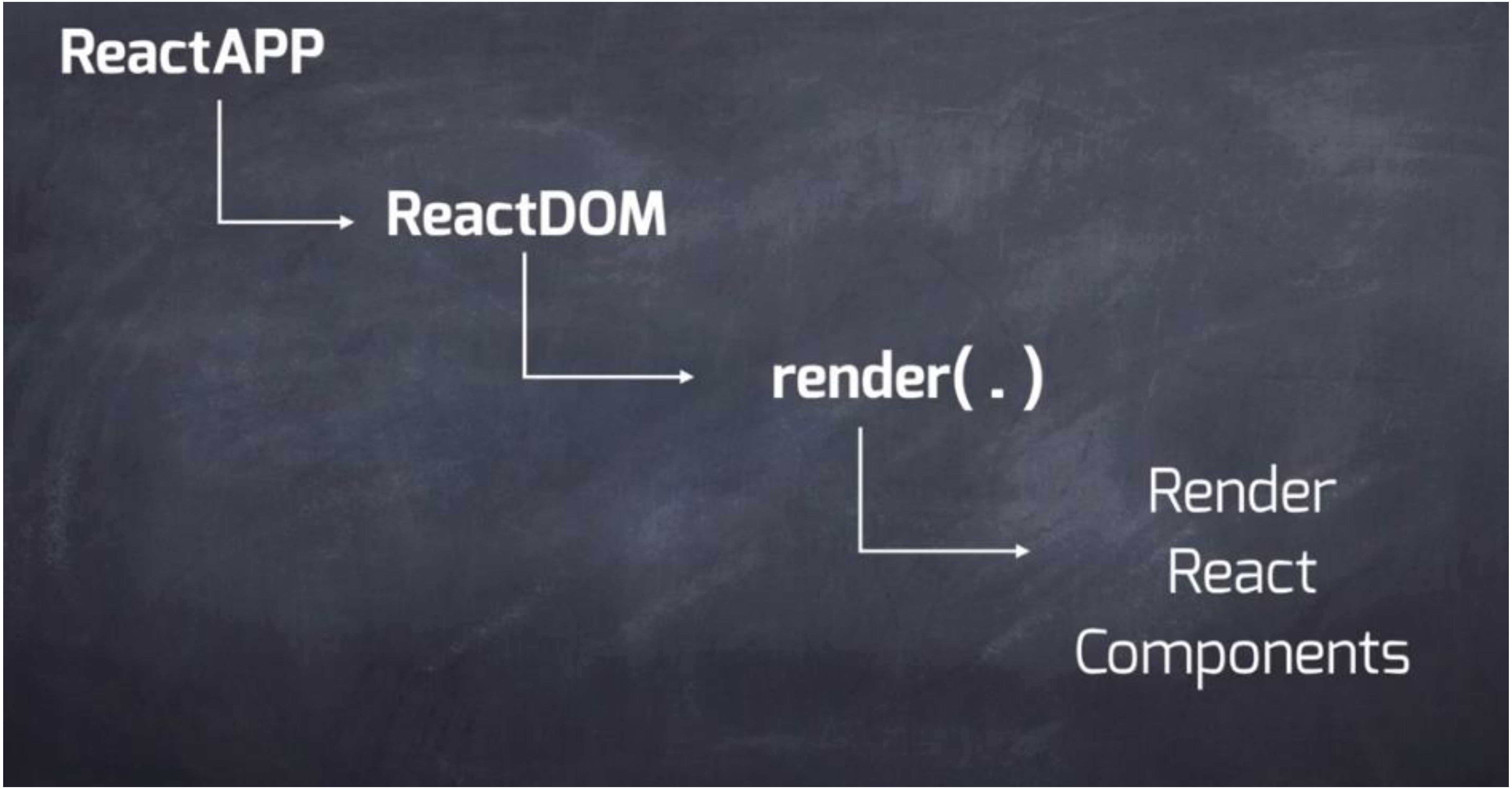
Single Post Component

The Verge

Original photography and video from The Verge, which covers the future of technology, science, and culture. For original art, follow @thevergeart

linkin.bio/verge

Followed by guy.in_the.sky, _hamza_mohd, mkbhd + 1 more



ReactAPP

ReactDOM

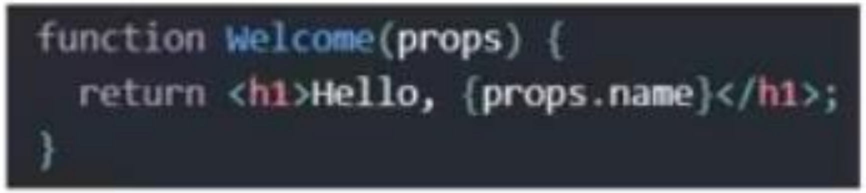
render(.)

Render
React
Components

Component Types

Stateless Functional Component

JavaScript Functions

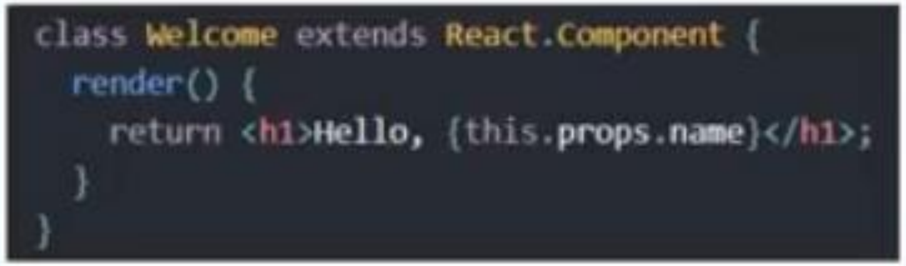


```
function Welcome(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

Stateful Class Component

Class extending Component class

Render method returning HTML



```
class Welcome extends React.Component {
  render() {
    return <h1>Hello, {this.props.name}</h1>;
  }
}
```