

프로젝트로 배우는 React 웹 애플리케이션 개발

Alex Kwon

리액트 입문 11/17



- **리액트 입문, 11/17** - ES6, 리액트 기본, 컴포넌트 작성/분리 하기
- **리액트 익숙해지기, 11/24** - props, state 다루기, 이벤트 핸들링
- **본격 컴포넌트 다루기, 12/1** - lifecycle, styling, propTypes, API 데이터 렌더링
- **애플리케이션 제작하기, 12/8** - JS 비동기 통신, routing, db 연동하기
- **리덕스 활용하기, 12/15** - flux & redux, redux로 리팩토링
- **나만의 애플리케이션 만들기, 12/22** - graphql 서버 제작하기, 리액트 앱 서버와 연동하기, 배포하기

- 슬라이드
- 예제 및 실습 코드
- 과제
- 교재



- 프론트엔드 라이브러리/프레임워크
- 리액트 소개
- DOM 그리고 Virtual DOM
- ES6 기본
- JSX
- 컴포넌트 작성하기
- 컴포넌트 분리하기

프론트엔드 라이브러리/프레임워크

- 프로그래밍을 할 때 자주 쓰는 다양한 기능을 한 데 모아놓은 것
- 라이브러리와 프레임워크의 차이 => 제어성의 차이
- 라이브러리 - 이곳 저곳에 사용할 수 있다! ex) jQuery, react.js, vue.js
- 프레임워크 - 프로젝트의 구성과 컨벤션이 Strict 함. ex) Angular

프론트엔드 라이브러리/프레임워크

```
<!DOCTYPE HTML>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<!--Google Webmaster-->
<meta name="google-site-verification"
content="ZAUbZzsdsIgyaQMMml6c6GVpJ98NIXx3GbMorPJBDxM" />
<title>Albuquerque Web Design + Print Design - Rody Studios</title>
<link href="css/rs5.css" rel="stylesheet" type="text/css">

<!--Starting Google Analytics-->
<script type="text/javascript">

    var _gaq = _gaq || [];
    _gaq.push(['_setAccount', 'UA-11755188-1']);
    _gaq.push(['_trackPageview']);

    (function() {
        var ga = document.createElement('script'); ga.type =
'text/javascript'; ga.async = true;
        ga.src = ('https:' == document.location.protocol ? 'https://ssl' :
'http://www') + '.google-analytics.com/ga.js';
        var s = document.getElementsByTagName('script')[0];
s.parentNode.insertBefore(ga, s);
    })();

</script>
</head>
```

HTML



```
body {
    font-family: "Helvetica Neue", Helvetica, Arial, sans-
font-size: 100%;
    line-height: 1.5;
    color: #000;
    background-image: url('../images/bkgd.jpg');
}
#logo-header {
    width: 100%;
    text-align: center;
}
#logo-header img {
    max-width: 80%;
    height: auto;
}
#descrip-header {
    width: 100%;
    text-align: center;
}
.worktile img {
    margin: 10px;
    box-shadow: 0px 0px 7px #aaa;
    -moz-box-shadow: 0px 0px 7px #aaa;
    -webkit-box-shadow: 0px 0px 7px #aaa;
}
```

CSS



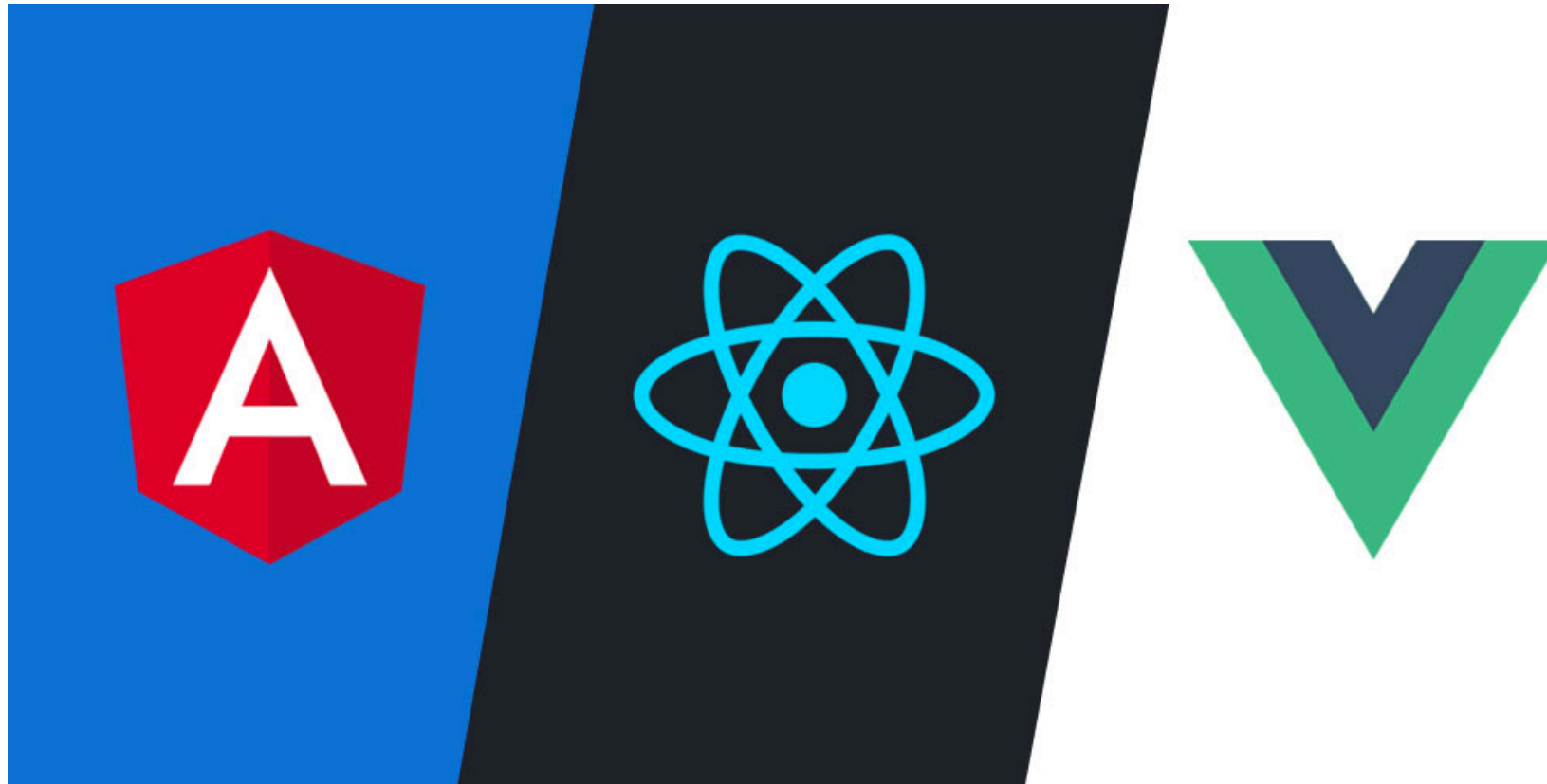
- HTML 및 CSS 만으로 개발



- jQuery 전성시대!



- 프론트엔드 프레임워크 대란
- react, angular, vue, ember, backbone..



- React, Angular, Vue로 정리되는 중!

프론트엔드 라이브러리/프레임워크



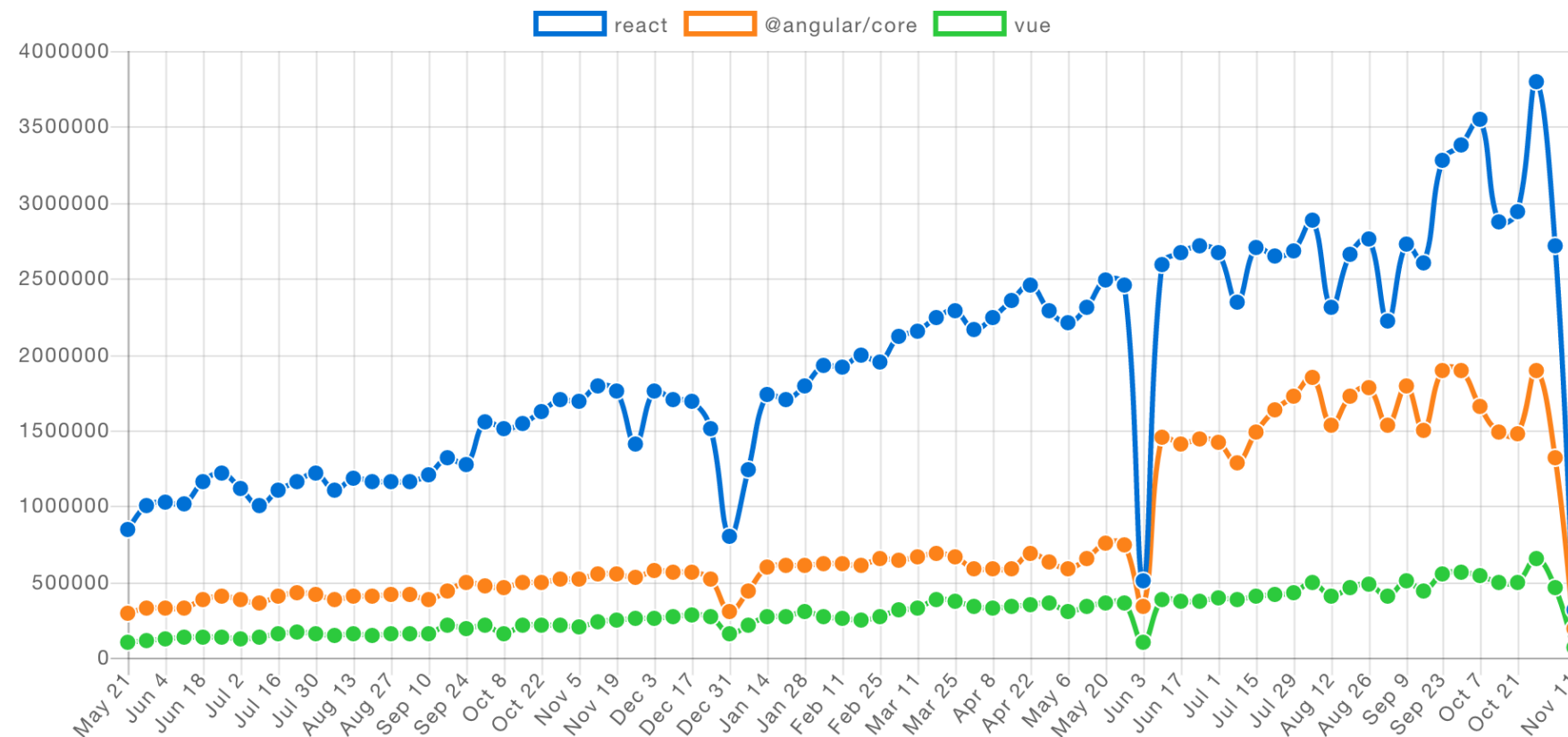
react vs @angular/core vs vue

react x

@angular/core x

vue x

Downloads in past 2 Years ▾



“React is a declarative, efficient, and flexible JavaScript library for building **user interfaces**.”

“리액트는 **유저 인터페이스**를 작성하기 위한 선언적이고 효율적이며 유연한 자바스크립트 라이브러리입니다.”

리액트 소개 - UI? 유저 인터페이스?

- 사용자 인터페이스는 사람(사용자)과 사물 또는 시스템, 특히 기계, 컴퓨터 프로그램 사이에서 의사소통을 할 수 있도록 일시적 또는 영구적인 접근을 목적으로 만들어진 물리적, 가상적 매개체.
- 인간이 프로그램과 상호작용하기 위한 것들!
- 웹앱, 모바일앱, 데스크톱 앱, VR 등등..

리액트 소개 - 리액트로 다 해먹기

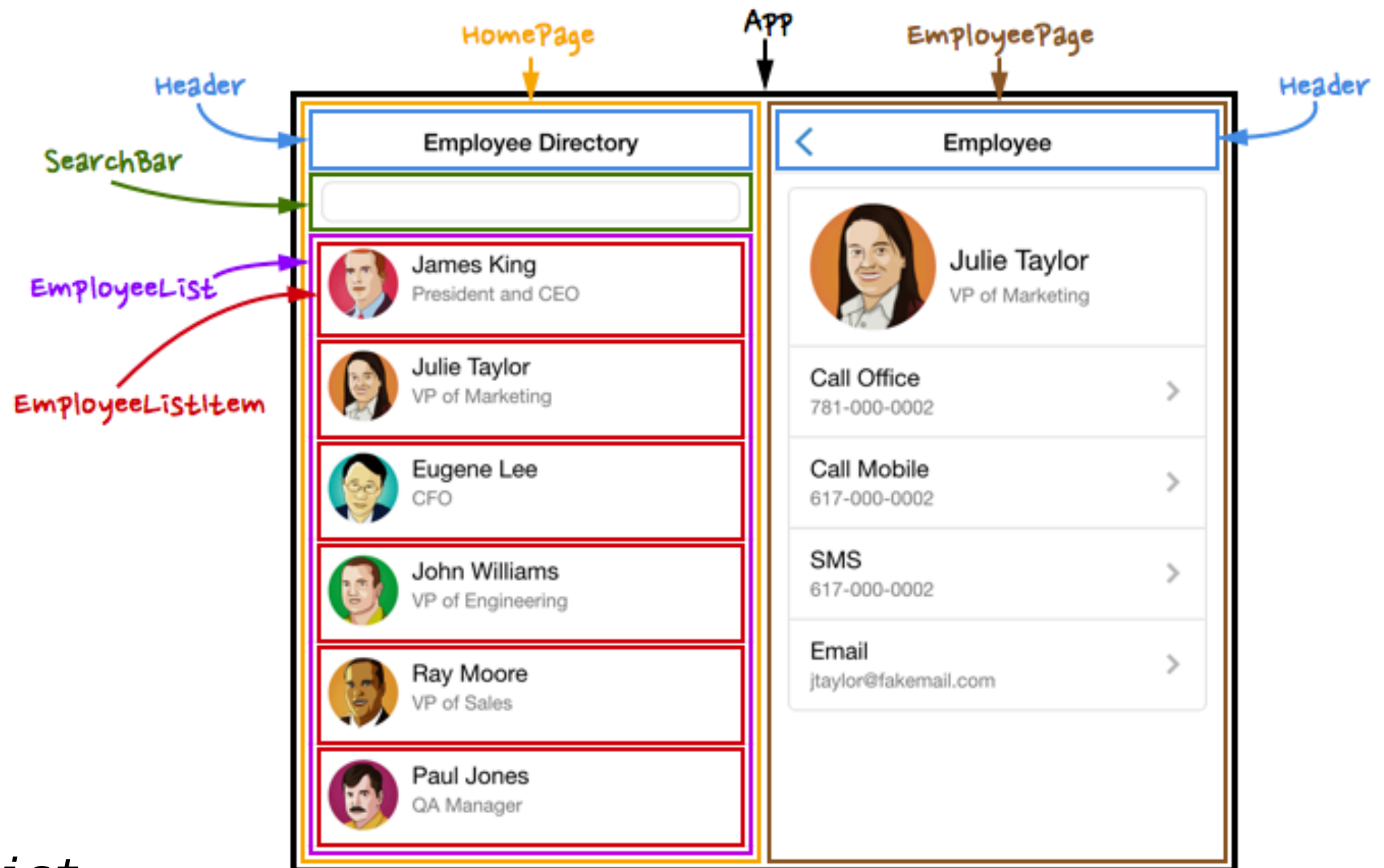
- React DOM - 웹 앱
- React Native - 모바일 앱
- React + Electron - 데스크탑 앱
- React360 - VR 앱

- 컴포넌트 기반
- JSX
- Virtual DOM

리액트 특징 - 컴포넌트란?

- UI 상에서의 최소 단위
- 컴포넌트의 집합
- 재사용 가능한 코드 블록
- View 와 로직의 덩어리

리액트 특징 - 컴포넌트란?



- App
 - HomePage
 - Header
 - SearchBar
 - EmployeeList
 - EmployeeListItem
 - EmployeePage
 - Header
 - EmployeeDetails

리액트 특징 - 컴포넌트 코드!



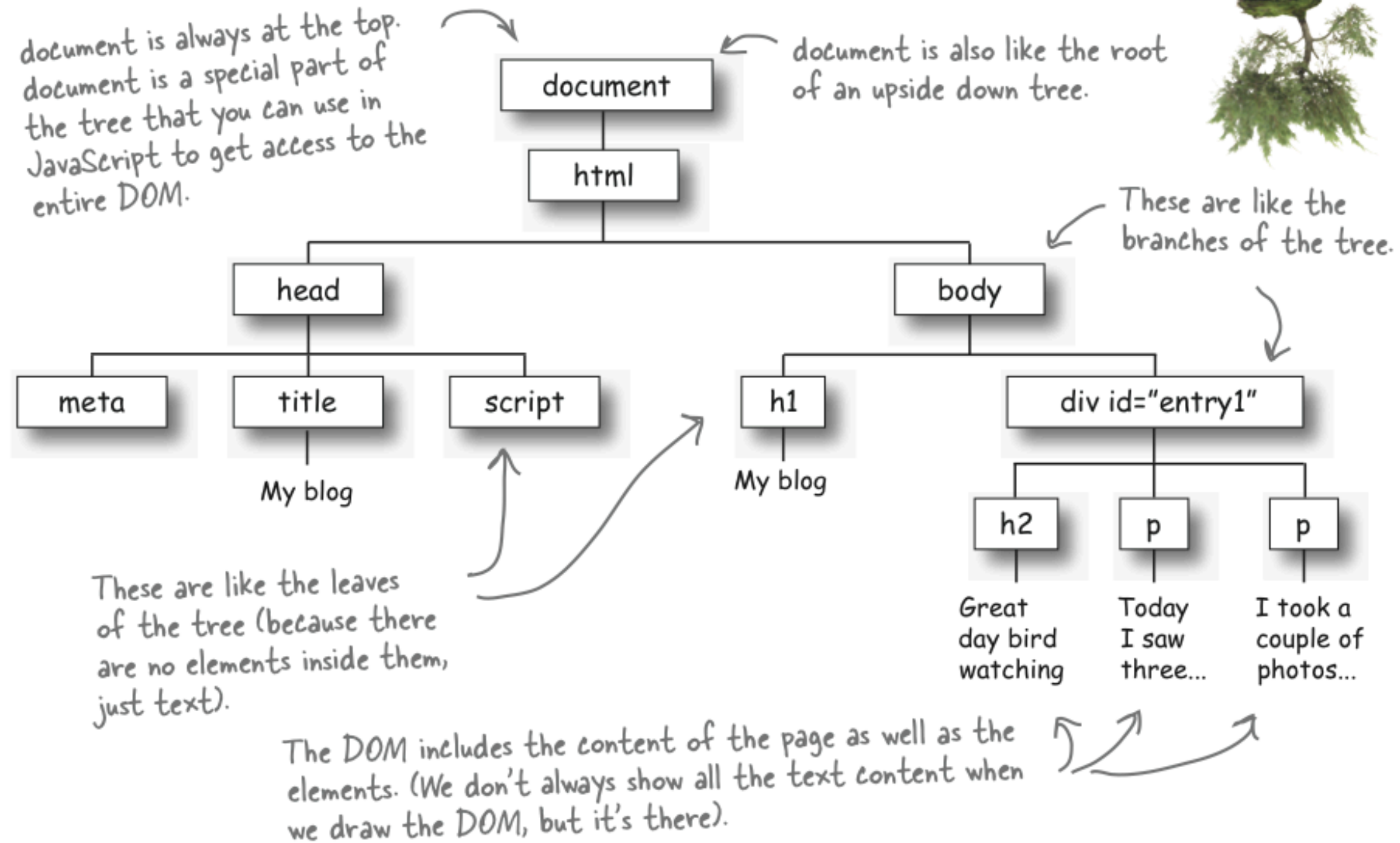
```
class HomePage extends React.Component {  
  
  searchHandler = (key) => {  
    alert('Search key: ' + key);  
  },  
  
  render() {  
    const employees = [  
      {firstName: 'Christophe', lastName: 'Coenraets'},  
      {firstName: 'Lisa', lastName: 'Jones'},  
      {firstName: 'John', lastName: 'Smith'}  
    ];  
  
    return (  
      <div>  
        <Header text="Employee Directory"/>  
        <SearchBar searchHandler={this.searchHandler}/>  
        <EmployeeList employees={employees}/>  
      </div>  
    );  
  }  
};
```

리액트 특징 - Virtual DOM

- DOM의 추상화 레이어
- JS로 관리되는 DOM 트리
- 빠른 성능

- Document Object Model
- HTML을 제어하고 조작할 수 있는 API => 프로그램으로 HTML을 다룰 수 있게 함
- HTML 엘리먼트 구조, 엘리먼트 속성, 엘리먼트 메소드, 엘리먼트 이벤트를 정의함

HTML DOM



1. 크롬 브라우저 열기
2. <https://google.com> 접속하기
3. 개발자도구 > Console 탭
4. `document.querySelector('#hplogo').remove();`

리액트 특징 - Virtual DOM의 장점

- DOM 요소의 업데이트/변경(mutation)의 비용이 높음
- 리액트는 유저 이벤트에 따른 변화를 Virtual DOM으로 관리
- 변경된 부분만 HTML DOM을 업데이트 => DOM 변경을 최소화

- let, const
- forEach, map
- 화살표 함수(arrow functions)
- class
- 전개 연산자(spread operator)
- 비구조화 할당(destructuring assignment)
- 객체 줄여쓰기

- 리액트에서 HTML 구조를 표현하기 위한 확장 문법
- HTML과 유사. 하지만 Javascript!
- Babel 을 사용하여 JS로 변환(transpile)



```
class HelloWorld extends React.Component {  
  render() {  
    return (  
      <h1 className='large'>Hello World</h1>  
    );  
  }  
}
```

JSX - Babel로 컴파일

```
class HelloWorld extends React.Component {  
  render() {  
    return (  
      React.createElement(  
        'h1',  
        {className: 'large'},  
        'Hello World'  
      )  
    );  
  }  
}
```

JSX - HTML 렌더링



```
<html>
  <head></head>
  <body>
    <h1 class="large">Hello World</h1>
  </body>
</html>
```



컴포넌트 작성하기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Hello world</title>
  <!-- Script tags including React -->
  <script crossorigin src="https://unpkg.com/react@16/umd/
react.production.min.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-
dom.production.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/
babel.js"></script>
</head>
<body>
  <div id="app"></div>
  <script type="text/babel">
    const app = <h1>Hello world</h1>
    const mountComponent = document.querySelector('#app');
    ReactDOM.render(app, mountComponent);
  </script>
</body>
</html>
```

컴포넌트 작성하기 - ReactDOM.render()

```
ReactDOM.render(<What />, Where!);
```

컴포넌트 작성하기 - <App /> 작성하기

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Hello world</title>
  <!-- Script tags including React -->
  <script crossorigin src="https://unpkg.com/react@16/umd/
react.production.min.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@16/umd/react-
dom.production.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.26.0/
babel.js"></script>
</head>
<body>
  <div id="app"></div>
  <script type="text/babel">
    class App extends React.Component {
      render() {
        return <h1>첫번째 App 컴포넌트 입니다!</h1>
      }
    }
    ReactDOM.render(<App />, document.querySelector('#app'));
  </script>
</body>
</html>
```

컴포넌트 분리하기 - <MyPage /> 작성하기

```
class MyPage extends React.Component {  
  render() {  
    return (  
      <div>  
          
        <ul>  
          <li>이름: Alex Kwon</li>  
          <li>주소: 대한민국 서울</li>  
          <li>직업: 한량</li>  
          <li>취미: 코딩, 주짓수, 국궁</li>  
        </ul>  
      </div>  
    );  
  }  
}
```

컴포넌트 분리하기 - <Profile /> 작성하기

```
class Profile extends React.Component {  
  render() {  
    return (  
      <ul>  
        <li>이름: Alex Kwon</li>  
        <li>주소: 대한민국 서울</li>  
        <li>직업: 한량</li>  
        <li>취미: 코딩, 주짓수, 국궁</li>  
      </ul>  
    );  
  }  
}
```

```
class MyPage extends React.Component {  
  render() {  
    return (  
      <div>  
          
        <Profile />  
      </div>  
    );  
  }  
}
```

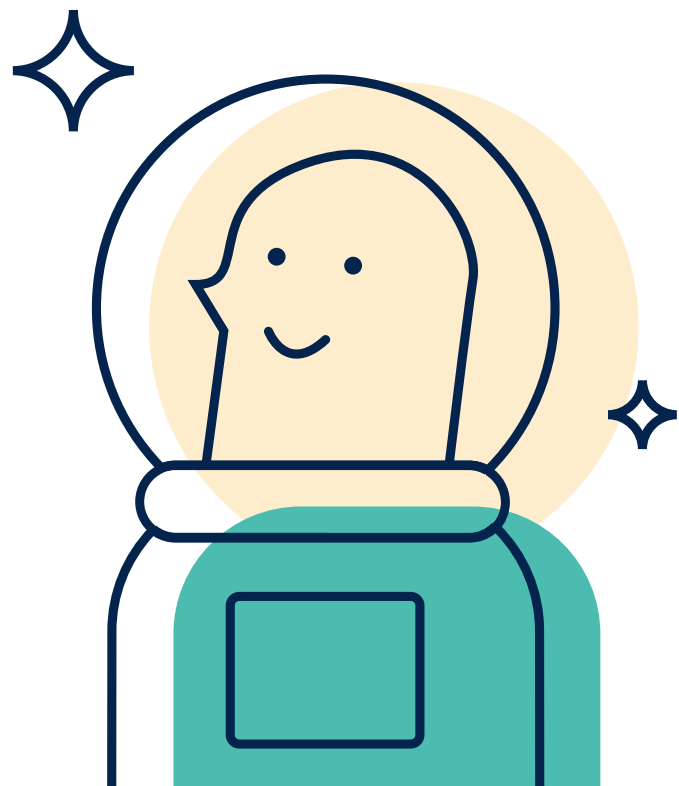

- 수많은 프론트엔드 라이브러리가 존재
- 리엑트는 유저 인터페이스를 작성하기 위한 라이브러리
- ES6 - 리엑트에서 사용하는 최신 자바스크립트 문법
- JSX를 활용한 컴포넌트 작성 및 분리하기

과제

- ES6 예제 코드 손으로 작성해보기
- 컴포넌트 작성하기, 분리하기 예제 작성해보기

복습

- ES6 문법 살펴보기 - <https://poiemaweb.com>
- 교재 읽기 - 챕터 2.1 부터 챕터 2.4 살펴보기



다음에
또!
같이!
만나요!