

Web Programming

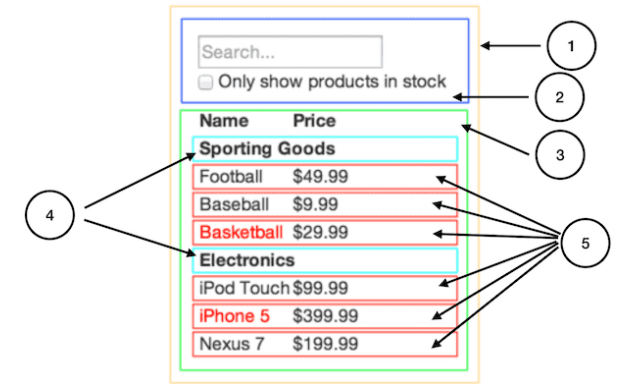
React programming



Component

컴포넌트

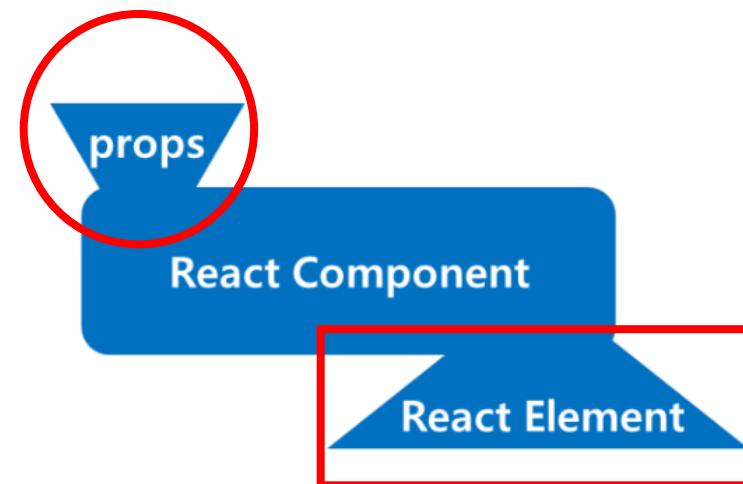
- Component : 입력(props)을 받아 출력(Element) 하는 역할
- 리엑트는 Component 기반의 구조라는 특징을 가지고 있다.
- 리엑트는 모든 페이지가 Component로 구성되어 있고 하나의 Component는 또 다른 여러 개의 Component의 조합으로 구성될 수 있다.
- 이러한 Component들을 마치 레고 블록을 조립하듯 끼워 맞춰 새로운 Component를 만들 수 있다.
- 프로그래밍에 있어 재사용이 가능한 각각의 독립된 모듈을 뜻한다
- 리엑트로 화면을 구성하게 되면, 사용자가 볼 수 있는 여러 가지 컴포넌트로 구성되어 있다. 사용자에게 보여지는 UI 요소를 컴포넌트 단위로 구분하여 구현할 수 있다.
- Components are like functions that return HTML elements.
- 웹 페이지를 만드는 퍼즐 조각
- 개발자는 컴포넌트 조각을 설계하고 만든 컴포넌트를 조합해서 사용자 인터페이스(user interface, UI)를 구축한다. UI 조각인 컴포넌트를 모으면 전체 퍼즐 그림인 웹 페이지를 만들 수 있다.
- 자바스크립트의 함수처럼 작동해서 리엑트 엘리먼트를 반환한다.
- 어떤 데이터 집합을 사용하든 같은 컴포넌트를 사용하면 모두 동일한 DOM 구조가 반환된다.





Component

자바 스크립트 컴포넌트와 리액트 컴포넌트



리액트 컴포넌트

- React Component에서의 입력은 props며 출력은 React Element가 된다.
- 결국 React Component가 해주는 역할은 어떠한 속성들을 입력으로 받아서 그에 맞는 React Element를 생성하여 리턴해 주는 것이다.
- React Element는 리액트 앱을 구성하는 가장 작은 빌딩 블록들이다.
- 자바스크립트 객체 형태로 존재하며 화면에 보이는 것을 기술한다.
- React Component를 만들고자 하는 대로 props, 즉 속성을 넣으면 해당 속성에 맞춰 화면에 나타날 Element를 만들어준다.

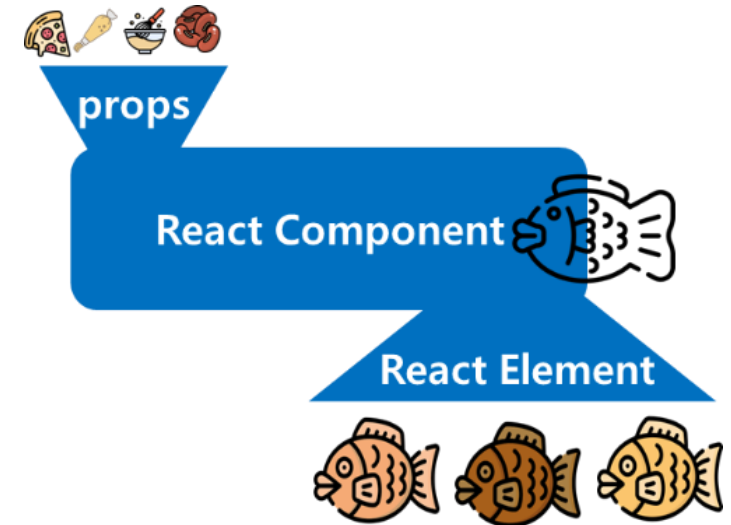
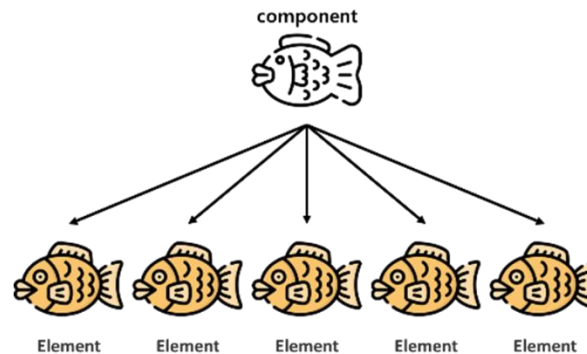


Component

리액트 컴포넌트

- props는 Component의 입력으로 들어간다.
- prop는 property의 의미로 재산이라는 뜻도 있지만 속성, 특성이라는 뜻도 가지고 있다.
- property를 줄여 prop이 되었고 여러 개라는 의미로 복수형 props가 되었다.
- 리액트에서 props는 리액트 컴포넌트의 속성이다.

	붕어빵 틀	Component
	붕어빵	Element
	붕어빵 재료	props



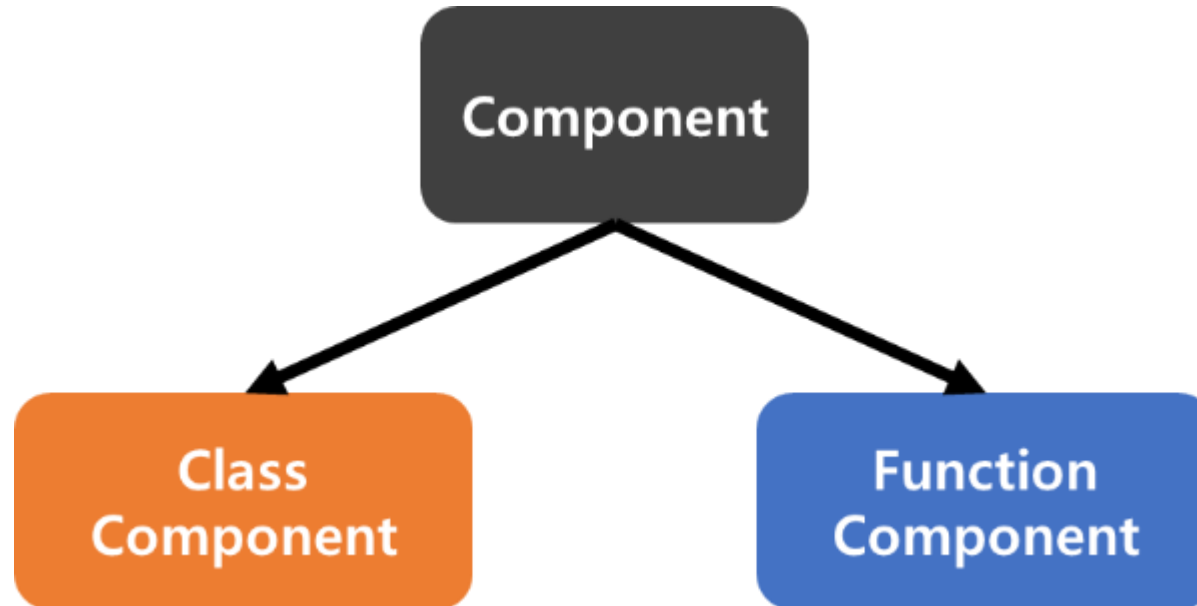
- 같은 붕어빵이라도 넣는 재료에 따라 다른 맛이 난다.
- 이처럼 props는 같은 React Component에서 눈에 보이는 글자나 색깔 등의 속성을 바꾸고 싶을 때는 사용하는 컴포넌트의 속 재료이다.
- 결국 React Component가 해주는 역할은 어떠한 속성들을 입력으로 받아서 그에 맞는 React Element를 생성하여 리턴해 주는 것이다.



Component

리액트 컴포넌트

- React에서 Component는 Class Component와 Function Component로 나뉜다.





Component

Function Component

- React Component는 pure 함수 같은 역할을 해야 한다.
- 즉, 리액트의 Component를 일종의 함수라고 생각한다.

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

- 이 함수는 하나의 props객체를 받아 인사말이 담긴 React element를 리턴하기 때문에 React Component라고 할 수 있다.
- 이런 것을 Function Component라고 한다.
- Function Component는 코드가 간단하다는 장점을 가지고 있다.



Component

Class Component

- 자바스크립트 ES6의 class라는 것을 사용해서 만들어진 형태의 Component다.
- Class Component의 경우 Function Component에 비해 몇 가지 추가적인 기능을 가지고 있다.

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

- 이 함수는 Function Component를 Class Component로 만든 것이다.



Component

Component name

- Component의 이름은 항상 대문자로 시작해야 한다.
- 리엑트는 소문자로 시작하는 Component를 DOM 태그로 인식하기 때문이다.
- div, span과 같이 사용하는 것은 내장 컴포넌트라는 것을 뜻하며 div나 span과 같은 문자열 형태로 `react.createElement`에 전달된다.
- Component 이름이 소문자로 시작했다면 리엑트는 내부적으로 이것을 Component가 아닌 DOM 태그로 인식한다.
- 그러므로 항상 컴포넌트의 이름은 대문자로 시작해야 한다.

```
class Car extends React.Component {  
  render() {  
    return <h2>Hi, I am a Car!</h2>;  
  }  
}
```

```
function Car() {  
  return <h2>Hi, I am a Car!</h2>;  
}
```

모든 HTML 컴포넌트



Component 만들기

Function Component

```
> npx create-react-app comp
```

```
> cd comp
```

VS Code – Open Folder... : "comp"

src – New Folder – "funcComp"

src – funcComp – New File – "Header.js"

src – funcComp – New File – "Main.js"

src – funcComp – New File – "Footer.js"

- 폴더는 소문자로 시작하는 카멜케이스로 작성하고, 컴포넌트를 만드는 파일은 대문자로 시작하는 카멜케이스로 이름을 지어야 한다.



Component 만들기

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

```
function Header(props) {  
  return (  
    <div>  
      <header>  
        <h1> 헤더입니다.</h1>  
      </header>  
    </div>  
  );  
}
```

```
export default Header; // 다른 JS 파일에서 불러올 수 있도록 내보내기
```

Header.js

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

```
function Main(props) {  
  return (  
    <div>  
      <main>  
        <h1>안녕하세요! 메인입니다.</h1>  
      </main>  
    </div>  
  );  
}
```

```
export default Main;
```

Main.js

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

```
function Footer(props) {  
  return (  
    <div>  
      <footer>  
        <h1>푸터입니다.</h1>  
      </footer>  
    </div>  
  );  
}
```

```
export default Footer;
```

Footer.js

rsf (React Stateless Functional Component의 약자)



Component 만들기

```
import React from 'react'; // 리액트를 구현할 수 있는 플러그인을 연결
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
// JS파일에 외부 파일을 불러오는 것이기 때문에 "import" 키워드를 사용한다.
// 같은 JS파일은 확장자를 사용하지 않는다.
```

```
function App() {
  return (
    <div>
      <Header />
      <Main />
      <Footer />
    </div>
  );
}

export default App;
```

App.js

헤더입니다.

안녕하세요!! 메인입니다.

푸터입니다.



Component 만들기

Class Component

src – New Folder – “classComp”

src – classComp – New File – “Header.js”

src – classComp – New File – “Main.js”

src – classComp – New File – “Footer.js”



Component 만들기

```
import React, { Component } from 'react'

export default class Header extends Component {
  render() {
    return (
      <div>
        <header>
          <h1>헤더입니다.</h1>
        </header>
      </div>
    )
  }
}
```

Header.js

```
import React, { Component } from 'react'

export default class Main extends Component {
  render() {
    return (
      <div>
        <main>
          <h1>안녕하세요!! 메인입니다.</h1>
        </main>
      </div>
    )
  }
}
```

Main.js

```
import React, { Component } from 'react'

export default class Footer extends Component {
  render() {
    return (
      <div>
        <footer>
          <h1>푸터입니다.</h1>
        </footer>
      </div>
    )
  }
}
```

Footer.js

rcc (React Class Component의 약자)



Component 만들기

App.js

```
import React from 'react'; // 리액트를 구현할 수 있는 플러그인을 연결
import Header from './classComp/Header';
import Footer from './classComp/Footer';
import Main from './classComp/Main';
// JS파일에 외부 파일을 불러오는 것이기 때문에 "import" 키워드를 사용한다.
// 같은 JS파일은 확장자를 사용하지 않는다.

function App() {
  return (
    <div>
      <Header />
      <Main />
      <Footer />
    </div>
  );
}

export default App;
```

헤더입니다.

안녕하세요!! 메인입니다.

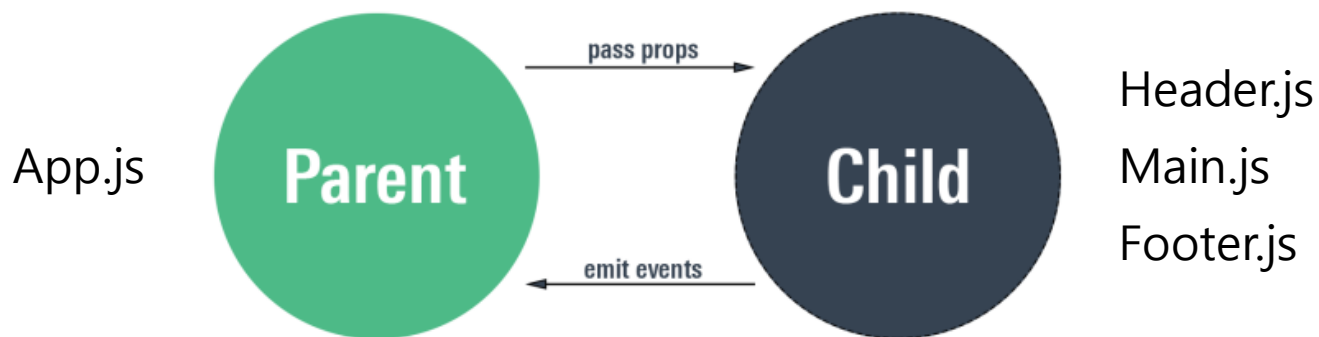
푸터입니다.



Props

프로퍼티(속성)

- 프로퍼티, props(properties의 줄임말)이다.
- 상위 컴포넌트가 하위 컴포넌트에 값을 전달할때 사용한다.(단방향 데이터 흐름 갖는다.)
- 프로퍼티는 수정할 수 없다는 특징이 있다.(자식입장에선 읽기 전용인 데이터이다.)
- React 컴포넌트는 props를 이용해 서로 통신한다.
- 모든 부모 컴포넌트는 props를 줌으로써 몇몇의 정보를 자식 컴포넌트에게 전달할 수 있다.
- 부모 컴포넌트가 자식 컴포넌트에게 물려준 데이터를 의미한다.





Props

프로퍼티 예제

- `<MyComponent name="React"/>` 에서 `name` 부분이 `props` 를 설정하는 부분이다.
- `name` 이라는 `props`를 지정하고, 그 값은 `React`가 된다.
- `App.js`는 `MyComponent`를 불러오는 부모 컴포넌트이다.

```
/* App.js */
import React from 'react';
import MyComponent from './MyComponent';

function App() {
  return (
    <MyComponent name="React"/>
  );
}
export default App;
```

```
/* MyComponent.js */
import React from 'react';

const MyComponent = (props) => {
  return <div> {props.name}로 만드는 테스트 페이지</div>;
};

export default MyComponent;
```

HTML로 만드는 테스트 페이지
Javascript로 만드는 테스트 페이지
React로 만드는 테스트 페이지



Props

1개의 프로퍼티 넘기기

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
```

```
function App() {
  return (
    <div>
      <Header />
      <Main name="홍길동"/>
      <Footer />
    </div>
  );
}
```

```
export default App;
```

App.js

```
import React from 'react';

function Main(props) {
  return (
    <div>
      <main>
        <h1>안녕하세요. {props.name} 입니다.</h1>
      </main>
    </div>
  );
}

export default Main;
```

Main.js



Props

2개의 프로퍼티 넘기기

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
```

```
function App() {
  return (
    <div>
      <Header />
      <Main name="갯대희" color="blue"/>
      <Footer />
    </div>
  );
}
```

```
export default App;
```

App.js

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

```
function Main(props) {
  return (
    <div>
      <main>
        <h1 style={{color: props.color}}>안녕하세요. {props.name} 입니다.</h1>
      </main>
    </div>
  );
}
```

```
export default Main;
```

Main.js

```
<h1 style={{color}}>안녕하세요. {name} 입니다.</h1>
```

Props 생략 가능



Props

숫자 프로퍼티 넘기기

- App.js (문자열 이외에는 중괄호({ }) 사용)

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
```

```
function App() {
  return (
    <div>
      <Header />
      <Main name={9} color="blue"/>
      <Footer />
    </div>
  );
}
```

```
export default App;
```

App.js

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

```
function Main(props) {
  return (
    <div>
      <main>
        <h1 style={{color: props.color}}>안녕하세요. {props.name} 입니다.</h1>
      </main>
    </div>
  );
}
```

```
export default Main;
```

Main.js

```
<h1 style={{color}}>안녕하세요. {name} 입니다.</h1>
```

props 생략 가능



Props

프로퍼티의 자료형, 타입정의

- 프로퍼티의 자료형을 미리 선언할 수 있다.
- 리액트 엔진이 프로퍼티로 전달하는 값을 효율적으로 알 수 있고, 버그예방에도 도움이 된다.
- 리액트에서 제공하는 prop-types를 이용하여 각각의 자료형을 선언하면 된다.

```
import React from 'react';
import PropTypes from 'prop-types' // 프로퍼티 타입을 지정해주기 위해 사용 한다.

function Main({name, color}) {
  return (
    <div>
      <main>
        <h1 style={{color}}>안녕하세요. {name} 입니다.</h1>
      </main>
    </div>
  );
}

// 프로퍼티 타입 지정
Main.propTypes = {
  name: PropTypes.string
}

export default Main;
```



Props

프로퍼티의 기본값 설정

- 기본값 설정
- 컴포넌트에 props 기본값을 설정하고 싶은 경우 defaultProps를 설정하면 된다.
ex) Main.js (name 프로퍼티가 없는 경우, '디폴트'라는 값을 사용 하게 처리)

```
import React from 'react';
import PropTypes from 'prop-types' // 프로퍼티 타입을 지정해주기 위해 사용 한다.
function Main({name, color}) {
  return (
    <div>
      <main>
        <h1 style={{color}}>안녕하세요. {name} 입니다.</h1>
      </main>
    </div>
  );
}
// 프로퍼티 타입 지정
Main.propTypes = {
  name: PropTypes.string
}
// 프로퍼티 기본값 지정
Main.defaultProps = {
  name: '디폴트'
}
export default Main;
```

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';

function App() {
  return (
    <div>
      <Header />
      <Main color="blue"/>
      <Footer />
    </div>
  );
}

export default App;
```



Props

프로퍼티의 필수값 설정

- 디폴트 설정을 하지 않는 경우 해당 프로퍼티를 필수 프로퍼티로 선언 할 수도 있다.
ex) Main.js (isRequired 를 통한 필수값 설정)

```
import React from 'react';
import PropTypes from 'prop-types' // 프로퍼티 타입을 지정해주기 위해 사용 한다.

function Main({name, color}) {
  return (
    <div>
      <main>
        <h1 style={{color}}>안녕하세요. {name} 입니다.</h1>
      </main>
    </div>
  );
}

// 프로퍼티 타입 지정 및 필수값 설정
Main.propTypes = {
  name: PropTypes.string.isRequired,
}

export default Main;
```



Props

불리언 프로퍼티 사용하기

- true, false만 정의 가능한 자료형
- 중괄호로 감싸 전달할 필요 없이 프로퍼티의 이름만 선언하면 된다.

```
import React from 'react';
import PropTypes from 'prop-types'
function Main({color, name, maleYn}) {
  const msg = maleYn ? '남자' : '여자'; // 불리언 사용
  return (
    <div>
      <main>
        <h1 style={{color}}>안녕하세요. {name} 입니다. ({msg})</h1>
      </main>
    </div>
  );
}
Main.propTypes = {
  name: PropTypes.string
}
Main.defaultProps = {
  name: '디폴트'
}
export default Main;
```

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';

function App() {
  return (
    <div>
      <Header />
      <Main name="홍길동" color="blue" maleYn/>
    <Footer />
  </div>
  );
}

export default App;
```

App.js에서 mailYn을 생략하면 false로 처리한다.
<Main name= " 홍길동" color="blue"/>



Props

props.children 활용하기

- children을 사용하여 내부에 있는 내용을 표현할 수 있다.

Wrapper.js

```
import React from 'react';

function Wrapper(props) {
  const style = {
    backgroundColor: 'yellow',
  };

  return (
    <div style={style}>
      </div>
  );
}

export default Wrapper;
```

App.js

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
import Wrapper from './funcComp/Wrapper';

function App() {
  return (
    <div>
      <Header />
      <Wrapper>
        <Main color="blue"/>
      </Wrapper>
      <Footer />
    </div>
  );
}

export default App;
```

Wrapper.js

```
import React from 'react';

function Wrapper(props) {
  const style = {
    backgroundColor: 'yellow',
  };

  return (
    <div style={style}>
      {props.children} {children}
    </div>
  );
}

export default Wrapper;
```

왼쪽의 Wrapper.js는 Main이 보이지 않으나 오른쪽 Wrapper.js는 Main이 보임



Props

props.children 활용하기

- 여러 개의 자식을 사용할 수도 있다.

App.js

```
import React, { Component } from 'react';
import Header from './funcComp/Header';
import Footer from './funcComp/Footer';
import Main from './funcComp/Main';
import Wrapper from './funcComp/Wrapper';

function App() {
  return (
    <div>
      <Header />
      <Wrapper>
        <Main name="홍길동" color="blue"/>
        <Main name="이순신" color="blue"/>
      </Wrapper>
      <Footer />
    </div>
  );
}

export default App;
```



Props

이미지와 텍스트를 출력하는 컴퍼넌트

FPhotoText.js

```
import React from 'react';

function FPhotoText(props) {
  const url = "img/" + props.image + ".png"
  const label = props.label
  const boxStyle = {
    border: "1px solid silver",
    margin: "8px",
    padding: "4px"
  }
  return (
    <div style={boxStyle}>
      <img src={url} width="128" alt="computer"/>
      <span>{label}</span>
    </div>
  );
}

export default PhotoText;
```

App.js

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

function App() {
  return (
    <div>
      <FPhotoText image="desktop" label = "Desktop" />
      <FPhotoText image="notebook" label = "NoteBook" />
      <FPhotoText image="pad" label = "Pad" />
    </div>
  );
}

export default App;
```



public – img 폴더 생성
Iconfinder에서 무료 이미지 다운로드(128)



Props

이미지와 텍스트를 출력하는 컴퍼넌트

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script src="https://unpkg.com/react@15/dist/react.min.js">
  </script>
  <script src="https://unpkg.com/react-dom@15/dist/react-dom.min.js">
  </script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.38/browser.min.js">
  </script>
</head>
<body>
  <div id="root"> </div>
  <script type="text/babel">    // 컴포넌트를 사용합니다.
  const dom = <div>
    <PhotoText image="pic1" label="데스크탑" />
    <PhotoText image="pic2" label="노트북" />
    <PhotoText image="pic3" label="패드" />
  </div>    // 리액트로 DOM의 내용을 변경합니다.
```

index.js

```
ReactDOM.render(dom,
document.getElementById('root'))
// 함수로 컴포넌트를 정의합니다.
function PhotoText (props) {
  const url = "img/" + props.image + ".jpeg"
  const label = props.label
  const boxStyle = {
    border: "1px solid silver",      margin: "8px",
    padding: "4px"
  }
  return <div style={boxStyle}>
    <img src={url} width="128"/>
    <span>{label}</span>
  </div>
}
</script>
</body>
</html>
```



Props

이미지와 텍스트를 출력하는 컴퍼넌트

CPhotoText.js

```
import React, { Component } from 'react'
export default class CPhotoText extends Component {
  getImageURL () {
    const id = this.props.image
    const url = "img/" + id + ".png"
    return url;
  }
  render() {
    const label = this.props.label
    const url = this.getImageURL()
    const boxStyle = {
      border: "1px solid silver",
      margin: "8px",
      padding: "4px"
    }
    return (
      <div style={boxStyle}>
        <img src={url} width="128" alt="computer" />
        <span> {label} </span>
      </div>
    )
  }
}
```

App.js

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

function App() {
  return (
    <div>
      <CPhotoText image="desktop" label = "Desktop" />
      <CPhotoText image="notebook" label = "NoteBook" />
      <CPhotoText image="pad" label = "Pad" />
    </div>
  );
}

export default App;
```



public – img 폴더 생성
Iconfinder에서 무료 이미지 다운로드(128)



Props

이미지와 텍스트를 출력하는 컴퍼넌트

index.js

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <script src="https://unpkg.com/react@15/dist/react.min.js"> </script>
  <script src="https://unpkg.com/react-dom@15/dist/react-dom.min.js"> </script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-core/5.8.38/browser.min.js">
  </script>
</head>
<body>
  <div id="root"> </div>
  <script type="text/babel">
    // 클래스로 PhotoText 컴포넌트를 정의합니다
    class PhotoText extends React.Component {
      getImageURL () {
        const id = this.props.image
        const url = "img/" + id + ".jpeg"
        return url;
      }
      render () {
        const label = this.props.label
        const url = this.getImageURL()
```

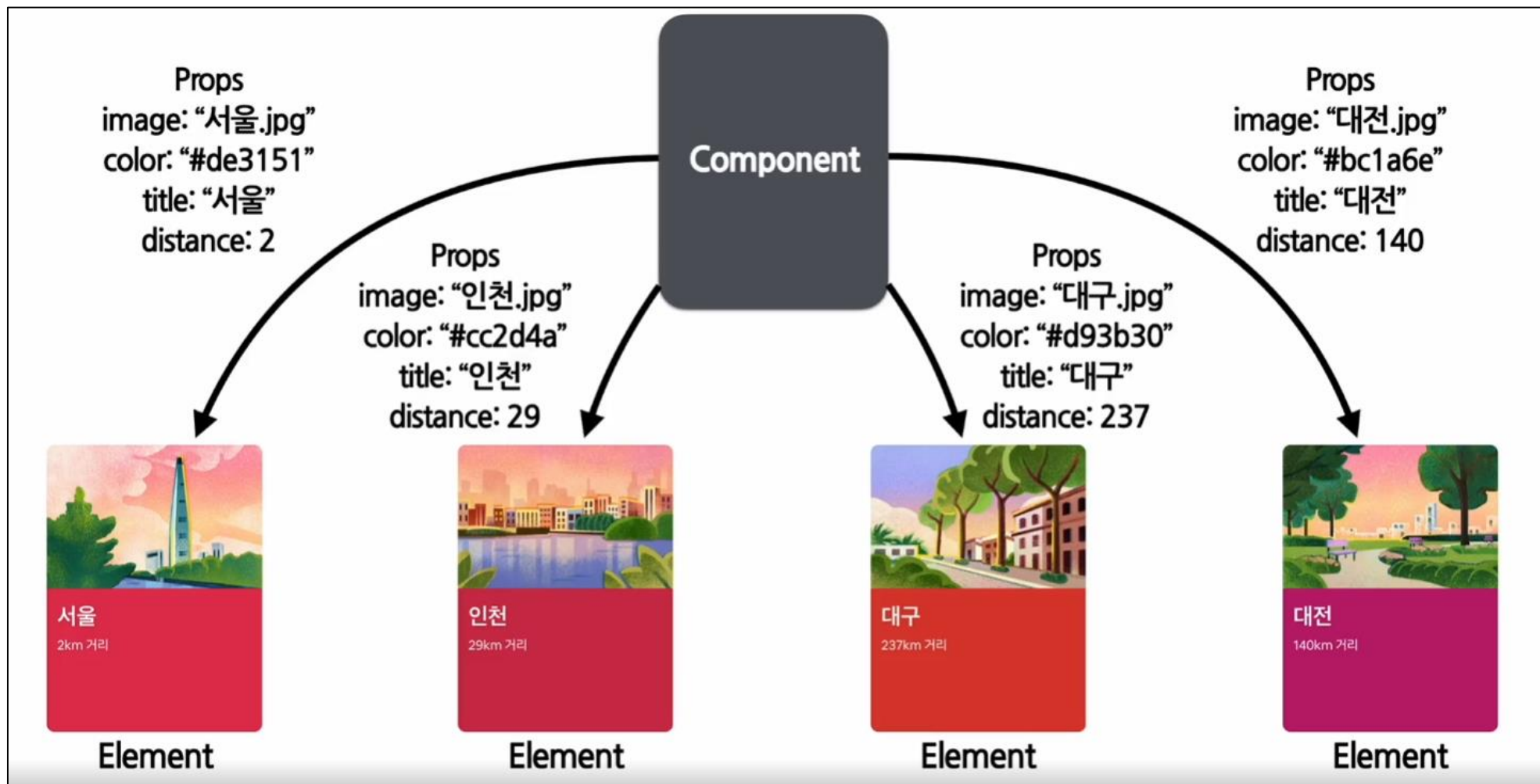
```
      const boxStyle = {
        border: "1px solid silver",
        margin: "8px",          padding: "4px"
      }
      return (<div style={boxStyle}>
        <img src={url} width="128"/>
        <span> {label} </span>
      </div>)
    }
  }
  // 컴포넌트를 사용합니다
  const dom = <div>
    <PhotoText image="pic1" label="데스크탑" />
    <PhotoText image="pic2" label="노트북" />
    <PhotoText image="pic3" label="패드" />
  </div>
  // 리액트로 DOM의 내용을 변경합니다.
  ReactDOM.render(dom,
    document.getElementById('root')) </script>
</body>
</html>
```



Props

이미지, color, 텍스트, 숫자를 출력하는 컴퍼넌트

<https://www.airbnb.co.kr/>





Props

리스트 컴포넌트

RList.js

```
import React, { Component } from 'react'

export default class RList extends Component {
  render() {
    // items 속성에 지정한 items 배열을 사용합니다.
    const items = this.props.items.split(",")
    // 아이템 목록을 기반으로 li 요소를 생성합니다.
    const itemsObj = items.map(
      (e) => {
        return <li>{e}</li>
      })
    // 타이틀
    let title = this.props.title
    if (!title) title = "LIST"
    // 렌더링할 내용을 반환합니다.
    return (
      <div>
        <h3>{title}</h3>
        <ul>{itemsObj}</ul>
      </div>
    )
  }
}
```

App.js

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

function App() {
  return (
    <div>
      <RList title="Colors" items="Red,Green,Blue,White" />
    </div>
  );
}

export default App;
```

Colors

- Red
- Green
- Blue
- White

```
<div>
  <Rlist title="과일" items="바나나, 사과, 딸기" />
  <Rlist title="채소" items="무, 당근, 오이" />
</div>
```



Props

화살표 함수

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
// 화살표 함수로 컴포넌트를 정의합니다.
const TitleParts = (props) => (
  <div style={{backgroundColor: 'red', color: 'white'}}>
    <h3>{props.title}</h3>
  </div>
)
const ContentParts = (props) => (
  <div style={{border: '1px solid blue', margin: 15}}>
    <div>줄거리: {props.body}</div>
  </div>
)
```

```
// 메인 컴포넌트
const Book = (props) => (
  <div>
    <TitleParts title={props.title} />
    <ContentParts body={props.body} />
  </div>
)
root.render(

  <React.StrictMode>
    <div>
      <Book title='삼국지' body='옛날 중국 이야기' />
      <Book title='민수기' body='옛날 이스라엘 이야기' />
      <Book title='서유기' body='원숭이가 활약하는 이야기' />
    </div>
  </React.StrictMode>
);

reportWebVitals();
```

삼국지

줄거리: 옛날 중국 이야기

민수기

줄거리: 옛날 이스라엘 이야기

서유기

줄거리: 원숭이가 활약하는 이야기



Component

컴포넌트 추출

```
function Comment(props){
  return {
    <div className = "comment">
      <div className = "user-info">
        <img className = "avatar"
          src = {props.author.avatarUrl}
          alt = {props.author.name}
        />
        <div className = "user-info-name">
          {props.author.name}
        </div>
      </div>
      <div className = "comment-text">
        {props.text}
      </div>
      <div className = "comment-date">
        {formatDate(props.date)}
      </div>
    </div>
  );
}
```



Comment



Component

컴포넌트 추출

```
function Avatar(props){  
  return (  
    <img className = "avatar"  
      src = {props.user.avatarUrl}  
      alt = {props.user.name}  
    />  
  );  
}
```



```
function Comment(props){  
  return {  
    <div className ="comment">  
      <div className = "user-info">  
        <Avatar user={props.author} />  
        <div className = "user-info-name">  
          {props.author.name}  
        </div>  
      </div>  
      <div className = "comment-text">  
        {props.text}  
      </div>  
      <div className = "comment-date">  
        {formatDate(props.date)}  
      </div>  
    </div>  
  );  
}
```



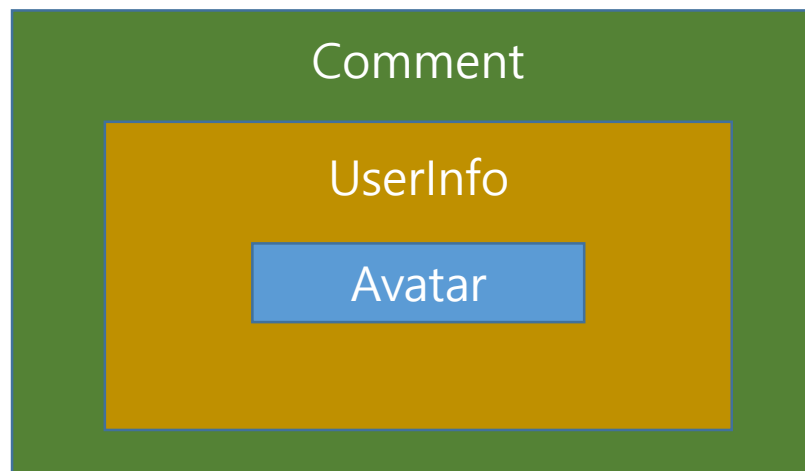
Component

컴포넌트 추출

```
function UserInfo(props) {  
  return (  
    <div className = "user-info">  
      <Avatar user = {props.user}  
      <div className = "user-info-name">  
        {props.user.name}  
      </div>  
    </div>  
  );  
}
```

```
function Avatar(props){  
  return (  
    <img className = "avatar"  
      src = {props.user.avatarUrl}  
      alt = {props.user.name}  
    />  
  );  
}
```

```
function Comment(props){  
  return {  
    <div className ="comment">  
      <UserInfo user = {props.author} />  
      <div className = "comment-text">  
        {props.text}  
      </div>  
      <div className = "comment-date">  
        {formatDate(props.date)}  
      </div>  
    </div>  
  );  
}
```



어느 수준까지 추출하라는 기준은 없음
기능 단위, 재사용 가능 여부로 추출



Component

재사용 가능한 Component를
많이 갖고 있을수록
개발 속도가 빨라진다!



Component

댓글 컴포넌트 만들기

- > npx create-react-app comment
- > cd comment

```
Index.js
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import CommentList from './CommentList';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <CommentList />
  </React.StrictMode>,
);

// If you want to start measuring performance in your app, pass a
function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

```
Comment.js
import React from "react";

function Comment(props) {
  return (
    <div>
      <h1>홍길동이 만든 컴포넌트입니다.</h1>
    </div>
  );
}
export default Comment;
```

```
CommentList.js
import React from "react";
import Comment from "./Comment";

function CommentList(props) {
  return (
    <div>
      <Comment />
    </div>
  );
}
```



Component

스타일 입히기

```
import React from "react";

const styles = {
  wrapper: {
    margin: 8,
    padding: 8,
    display: "flex",
    flexDirection: "row",
    border: "1px solid grey",
    borderRadius: 16,
  },
  imageContainer: {},
  image: {
    width: 50,
    height: 50,
    borderRadius: 25,
  },
  contentContainer: {
    marginLeft: 8,
    display: "flex",
    flexDirection: "column",
    justifyContent: "center",
  },
};
```

Comment.js

```
function Comment(props) {
  return (
    <div style={styles.wrapper}>
      <div style={styles.imageContainer}>
        

```



홍길동

홍길동이 만든 컴포넌트입니다.

```
      nameText: {
        color: "black",
        fontsize: 16,
        fontWeight: "bold",
      },
      commentText: {
        color: "black",
        fontSize: 16,
      },
    },
  };
};
```

```
    </div>
    <div style={styles.contentContainer}>
      <span style={styles.nameText}>홍길동 </span>
      <span style={styles.commentText}>
        홍길동이 만든 컴포넌트입니다.
      </span>
    </div>
  </div>
);
}

export default Comment;
```



Component

props 추가하기

```
function Comment(props) {  
  return (  
    <div style={styles.wrapper}>  
      <div style={styles.imageContainer}>  
          
      </div>  
      <div style={styles.contentContainer}>  
        <span style={styles.nameText}>{props.name}</span>  
        <span style={styles.commentText}>{props.comment}</span>  
      </div>  
    </div>  
  );  
}  
export default Comment;
```

Comment.js



Component

props 추가하기

```
import React from "react";
import Comment from "../Comment";

function CommentList(props) {
  return (
    <div>
      <Comment name="홍길동" comment="홍길동이 만든 컴포넌트입니다."/>
    </div>
  );
}

export default CommentList;
```

CommentList.js



홍길동
홍길동이 만든 컴포넌트입니다.

```
<Comment name="홍길동" comment="홍길동이 만든 컴포넌트입니다."/>
<Comment name="유재석" comment=" 리액트 재밌어요~~. " />
```




Component

컴포넌트 데이터 객체로 분리하기

```
import React from "react";
import Comment from "./Comment";
const comments = [
  {
    name: "홍길동",
    comment: "안녕하세요! 홍길동입니다.",
  },
  {
    name: "유재석",
    comment: "리액트 재밌어요~~.",
  },
  {
    name: "최예진",
    comment: "안녕하세요! 프론트엔드 개발자입니다.",
  },
];
```

```
function CommentList(props) {
  return (
    <div>
      {comments.map((comment) => {
        return (
          <Comment name={comment.name}
            comment={comment.comment}/>
        );
      })}
    </div>
  );
}

export default CommentList;
```



홍길동
안녕하세요! 홍길동입니다.



유재석
리액트 재밌어요~~.



최예진
안녕하세요! 프론트엔드 개발자입니다.





Report

웹사이트를 5개 이상의 컴포넌트로 구성하세요
2개 이상의 props를 전달하는 컴포넌트를 하나 이상 포함하세요

발표주제

pure function , impure function
화살표 함수, 람다 함수
map() 함수



Reference

- <https://goddaehee.tistory.com/299>
 - <https://wikidocs.net/197615>
 - <https://ajdkfl6445.gitbook.io/study/react/1-object>
 - <https://javascript.plainenglish.io/creating-your-first-react-application-db9dfe93b4f5>
-
- 모던 자바스크립트 개발자를 위한 리액트 프로그래밍, 쿠지라히코우즈키에, 윤인성, 위키북스
 - 명품 웹 프로그래밍, 황기태, 생능출판사
 - 리액트 & 리액트 네이티브 통합 교과서, 아담 보두치, 강경일, 신희철, 에이콘
 - Do it! 리액트 모던 웹 개발 with 타입스크립트, 전예홍, 이지스퍼블리싱