

# React

◆ 프로퍼티

정수아

# Contents

**01** 프로퍼티란?

**02** 프로퍼티 값 전달 및 조회

**03** 불리언 프로퍼티

**04** defaultProps

**05** props.children

**06** propTypes



01

**프로퍼티(properties)란?**

# 프로퍼티(properties)는 왜 필요한가?

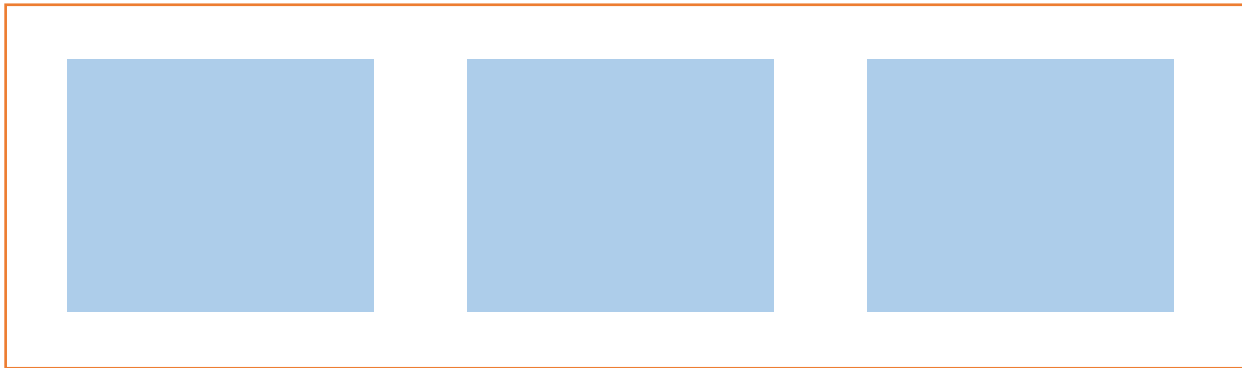
## ❖ 영화 목록을 컴포넌트로 표현한다고 가정

- 1. Movie 컴포넌트 생성 후 영화 정보 입력

영화 제목 : 공조2: 인터내셔널  
개봉 : 2022.09.07

Movie 컴포넌트

- 2. App.js 컴포넌트에서 Movie 컴포넌트 3번 사용



App.js

# 프로퍼티(properties)는 왜 필요한가?

## ❖ 실행 결과

영화 제목 : 공조2: 인터내셔널  
개봉 : 2022.09.07

영화 제목 : 공조2: 인터내셔널  
개봉 : 2022.09.07

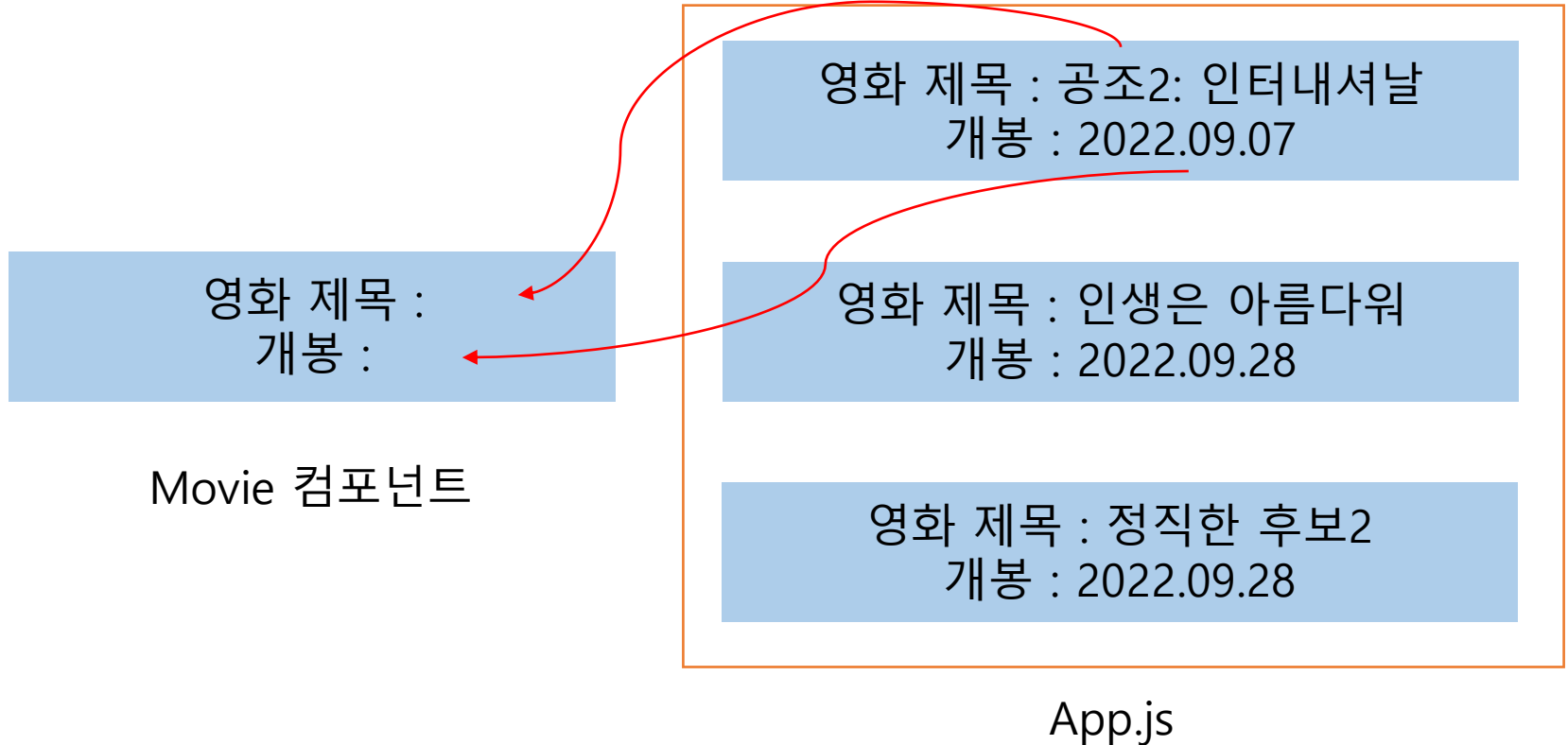
영화 제목 : 공조2: 인터내셔널  
개봉 : 2022.09.07

App.js

# 프로퍼티(properties)는 왜 필요한가?

## ❖ 해결 방법

- Movie 컴포넌트는 1개만 생성
- App.js에서 Movie 컴포넌트에 필요한 데이터를 전달



# 프로퍼티(properties)

## ❖ 프로퍼티란?

- 상위 컴포넌트에서 하위 컴포넌트로 전달되는 읽기 전용 데이터
  - 단방향 데이터 흐름
- 프로퍼티 값은 수정이 불가능 함
- properties를 줄여서 props라고도 함
- 전달 방법

```
<하위컴포넌트명 전달하려는_속성명 = 전달하려는_값 />
```

# 프로퍼티 값 자료형

## ❖ 문자열

- 큰따옴표(“) 사용

```
<MyComponent name = "정수아" />
```

## ❖ 문자열 이외의 값

- 숫자, 불리언, 배열, 객체, 함수 등
- 중괄호 사용

```
<MyComponent age = {20} />
```





02

## 프로퍼티 값 전달 및 조회

# props를 통해 하나의 값 전달하기

## ❖ 데이터 조회하기

- FirstProps.js

```
import React from 'react';

function FirstProps(props){
  return (
    <div>
      안녕하세요. 제 이름은 {props.name}입니다.
    </div>
  );
}

export default FirstProps;
```

# props를 통해 하나의 값 전달하기

## ❖ 데이터 전달하기

- App.js

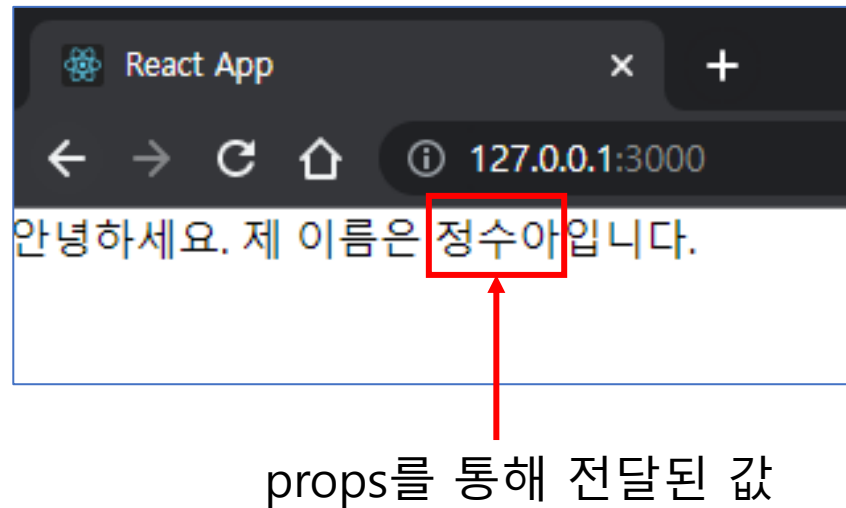
```
import React from 'react';
import FirstProps from './FirstProps';

function App(){
  return (
    <div>
      <FirstProps name="정수아" />
    </div>
  );
}

export default App;
```

# props를 통해 하나의 값 전달하기

## ❖ 실행 결과



# props를 통해 여러 개의 값 전달하기

## ❖ 데이터 조회하기

- FirstProps.js

```
function FirstProps(props){  
  return (  
    <div>  
      <h1>  
        안녕하세요. 제 이름은 {props.name}입니다.  
      <br/>  
      제가 좋아하는 색은 {props.color}입니다.  
      </h1>  
    </div>  
  );  
}
```

# props를 통해 여러 개의 값 전달하기

## ❖ 데이터 전달하기

- App.js

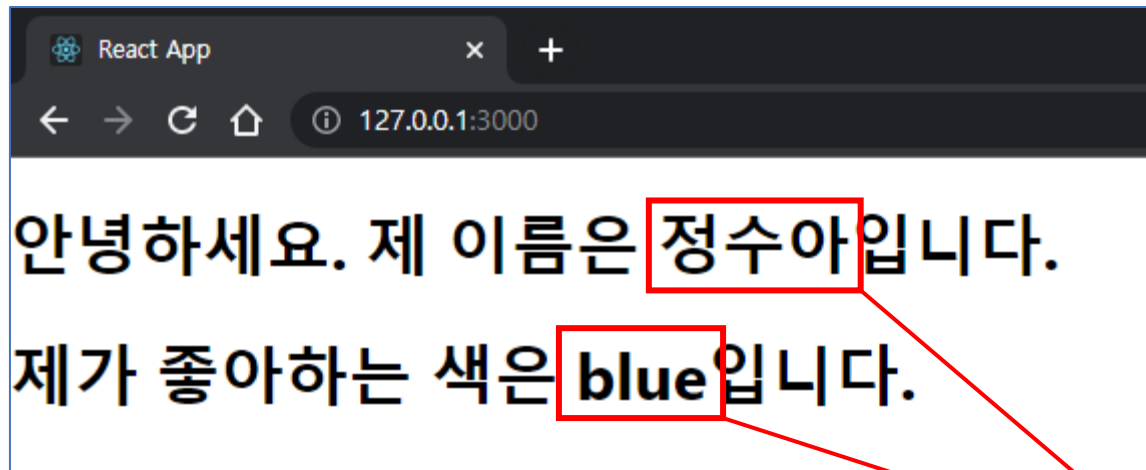
```
import React from 'react';
import FirstProps from './FirstProps';

function App(){
  return (
    <div>
      <FirstProps name="정수아" color="blue"/>
    </div>
  );
}

export default App;
```

# props를 통해 여러 개의 값 전달하기

## ❖ 실행 결과



props를 통해 전달된 값

# props 값 간결하게 조회하기

## ❖ 비구조화 할당(=구조 분해) 문법 사용

```
function FirstProps({name, color}){  
  return (  
    <div>  
      <h1>  
        안녕하세요. 제 이름은 {name}입니다.  
        <br/>  
        제가 좋아하는 색은 {color}입니다.  
      </h1>  
    </div>  
  );  
}
```



# props 값 간결하게 조회하기

## ❖ 비구조화 할당(=구조 분해) 문법 사용

```
function FirstProps(props){  
  const {name, color} = props;  
  
  return (  
    <div>  
      <h1>  
        안녕하세요. 제 이름은 {name}입니다.  
      <br/>  
      제가 좋아하는 색은 {color}입니다.  
    </h1>  
  </div>  
  );  
}
```

# 클래스형 컴포넌트에서 props 사용

## ❖ 데이터 전달하기

- App.js

```
import React, { Component } from 'react';
import ClassProps from './ClassProps';

class App extends Component {
  render() {
    return (
      <div>
        <ClassProps name="정수아" color="blue"/>
      </div>
    );
  }
}

export default App;
```

# 클래스형 컴포넌트에서 props 사용

## ❖ 데이터 조회하기

- ClassProps.js

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

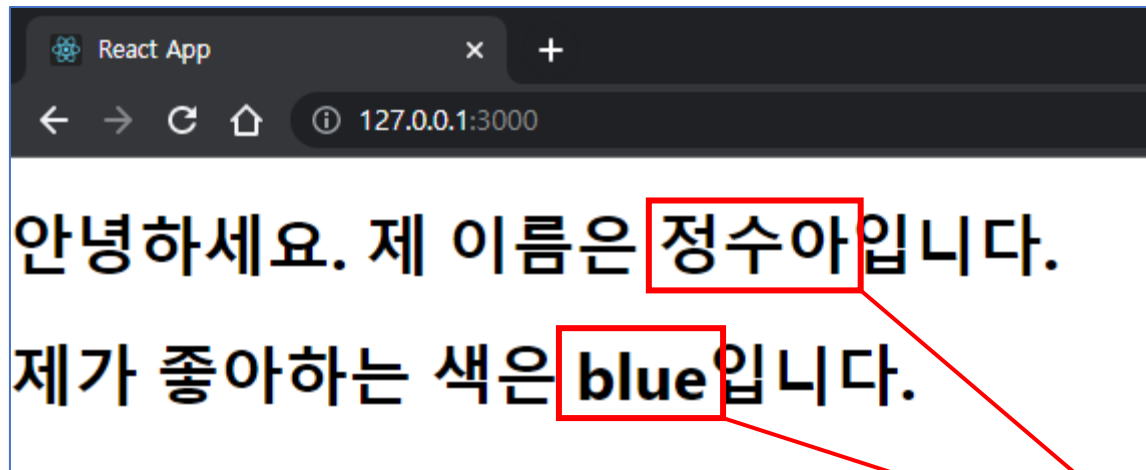
class ClassProps extends Component {
  render() {
    const {name, color} = this.props;

    return (
      <div>
        <h1>
          안녕하세요. 제 이름은 {name}입니다.
          <br/>
          제가 좋아하는 색은 {color}입니다.
        </h1>
      </div>
    );
  }
}

export default ClassProps;
```

# 클래스형 컴포넌트에서 props 사용

## ❖ 실행 결과



props를 통해 전달된 값



03

## 불리언 프로퍼티

# 불리언 프로퍼티

## ❖ 불리언 프로퍼티 사용하기

- 불리언 값은 특별한 방법으로 전달 가능
  - true는 프로퍼티의 이름만 선언함

```
<BooleanComponent boolValue />
```

- false는 프로퍼티의 이름을 생략함

```
<BooleanComponent />
```

# 불리언 프로퍼티

## ❖ BooleanComponent.js

```
import React from 'react';

function BooleanComponent({bored}){
  const message = bored ? '놀러 가자' : '하던 일 열심히  
마무리하기'

  return (
    <div>
      {message}
    </div>
  );
}

export default BooleanComponent;
```

# 불리언 프로퍼티

## ❖ App.js

```
import React from 'react';
import BooleanComponent from './BooleanComponent';

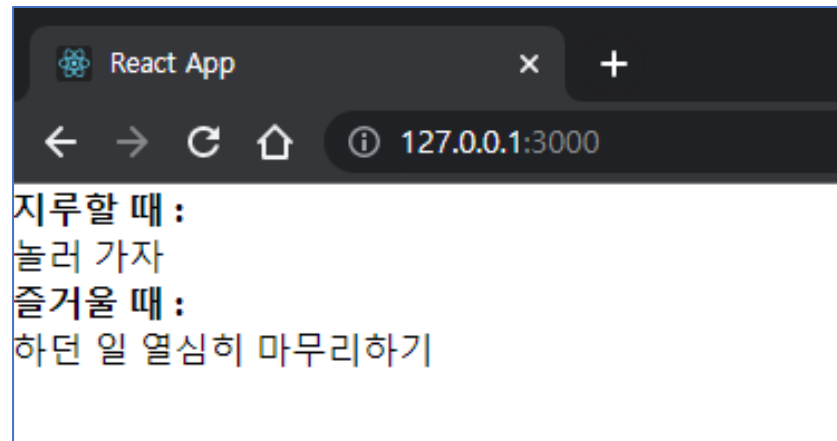
function App(){
  return (
    <div>
      <div>
        <b>지루할 때 : </b>
        <BooleanComponent bored/>
      </div>
      <div>
        <b>즐거울 때 : </b>
        <BooleanComponent/>
      </div>
    </div>
  );
}

export default App;
```



# 불리언 프로퍼티

## ❖ 실행 결과





04

**defaultProps**

# defaultProps

## ❖ defaultProps

- props를 지정하지 않았을 때 기본적으로 사용할 값을 설정
- DefaultProps.js

```
import React from 'react';

function DefaultProps({name}){
  return(
    <div>
      안녕하세요. 제 이름은 {name}입니다.
    </div>
  );
}

// 컴포넌트이름.defaultProps
DefaultProps.defaultProps = {
  name : "이름없음"
}

export default DefaultProps;
```

# defaultProps

- App.js

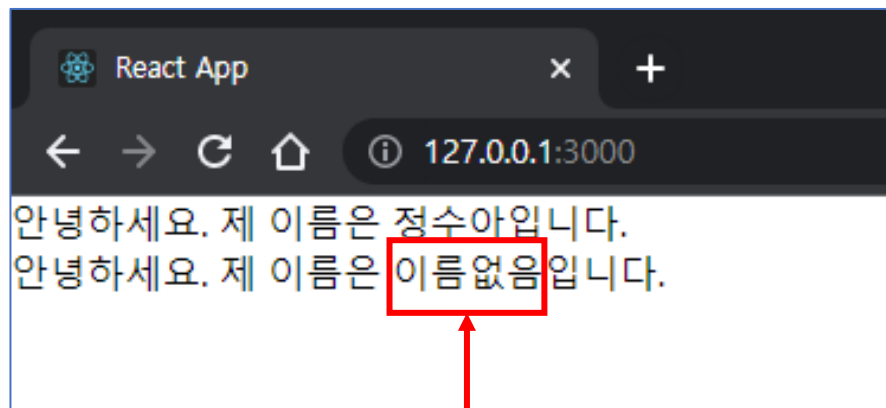
```
import React from 'react';
import DefaultProps from './DefaultProps';

function App(){
  return (
    <div>
      <DefaultProps name="정수아"/>
      <DefaultProps/>
    </div>
  );
}

export default App;
```

# defaultProps

## ❖ 실행 결과



defaultProps를 통해 전달된 값



**05**

**props.children**

# props.children

## ❖ props.children

- 컴포넌트 태그 사이에 넣은 값을 조회할 때 사용

```
<ParentComponent>  
  안녕하세요  
  <childComponent/>  
</ParentComponent>
```

# props.children

## ❖ Wrapper.js

```
import React from 'react';

function Wrapper(){
  const style = {
    border : '3px solid red',
    padding : '20px'
  }

  return (
    <div style={style}>

    </div>

  );
}

export default Wrapper;
```



# props.children

## ❖ App.js

```
import React from 'react';
import Wrapper from './Wrapper';
import HeaderComponent from './CreateHeader';
import BodyComponent from './CreateBody';
import FooterComponent from './CreateFooter';

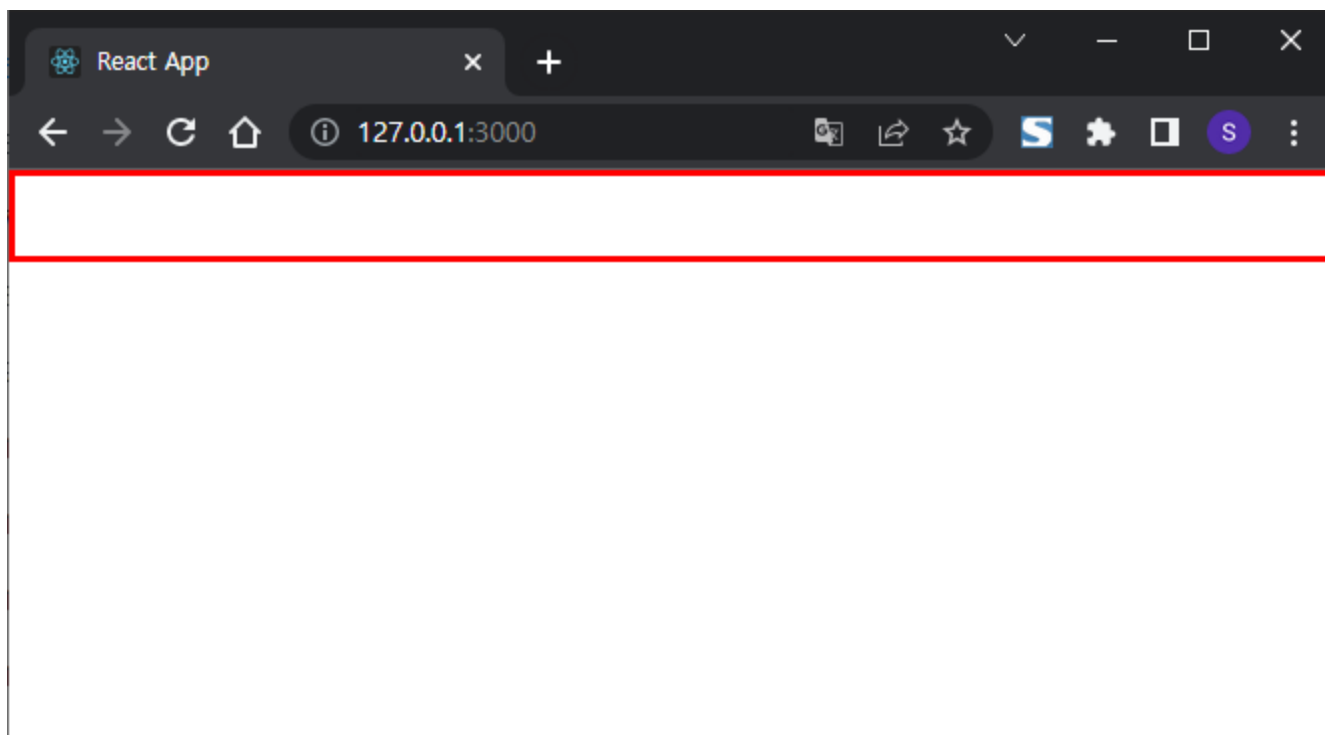
function App(){
  return (
    <Wrapper>
      <HeaderComponent/>
      <BodyComponent/>
      <FooterComponent/>
    </Wrapper>
  );
}

export default App;
```

# props.children

## ❖ 실행 결과

- Wrapper 컴포넌트만 화면에 출력
- HeaderComponent, BodyComponent, FooterComponent는 화면에 출력 안됨



# props.children

## ❖ Wrapper.js 수정

```
import React from 'react';

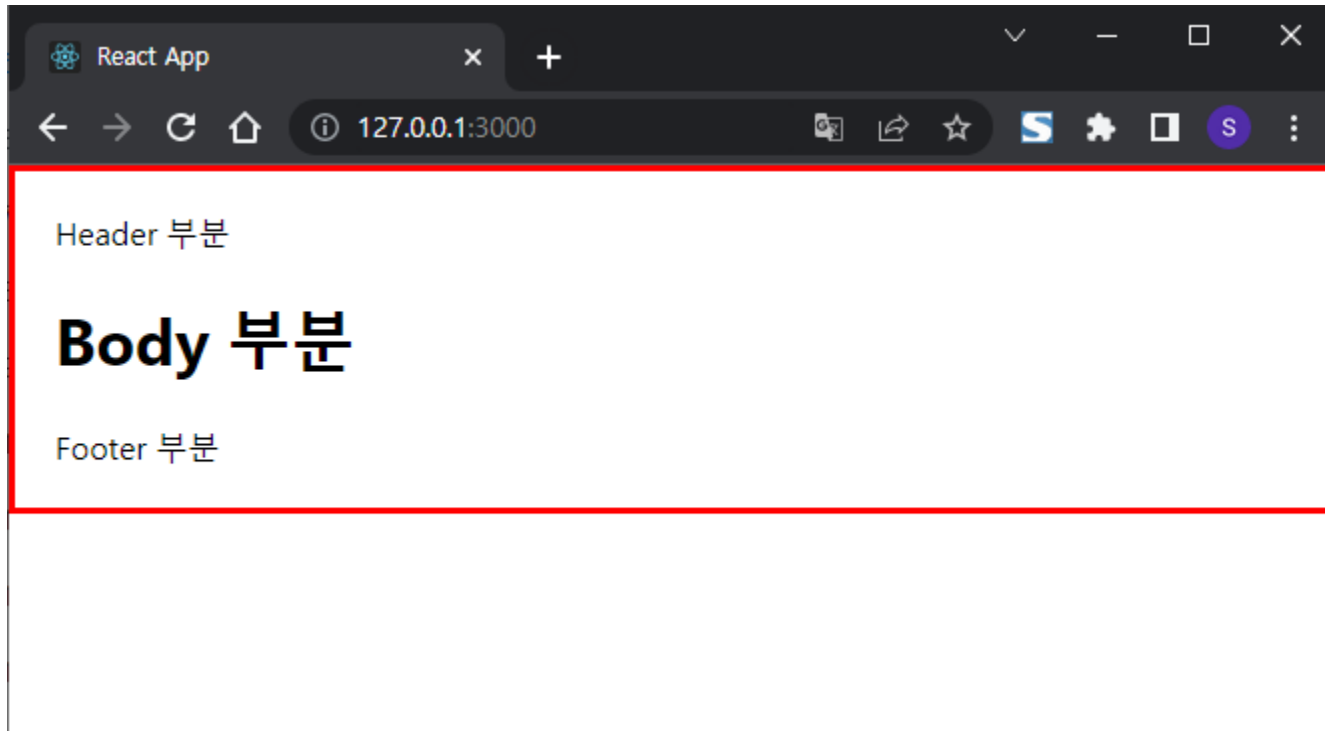
function Wrapper({children}){
  const style = {
    border : '3px solid red',
    padding : '20px'
  }

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

export default Wrapper;
```

# props.children

## ❖ 실행 결과





06

**propTypes**

# propTypes

## ❖ propTypes

- 특정 컴포넌트에 꼭 전달되어야 하는 프로퍼티를 필수 프로퍼티로 지정하거나 프로퍼티의 타입을 지정할 때 사용
- 사용 방법
  - prop-types를 import 해야 함

```
import PropTypes from 'prop-types';
```

# propTypes - 타입 지정

## ❖ PropTypesComponent.js

```
import React from 'react';
import PropTypes from 'prop-types';

function PropTypesComponent({name}){
  return(
    <div>
      안녕하세요. 제 이름은 {name}입니다.
    </div>
  );
}

PropTypesComponent.propTypes = {
  name : PropTypes.string
}

export default PropTypesComponent;
```

# propTypes - 타입 지정

## ❖ App.js

```
import React from 'react';
import PropTypesComponent from './PropTypesComponent';

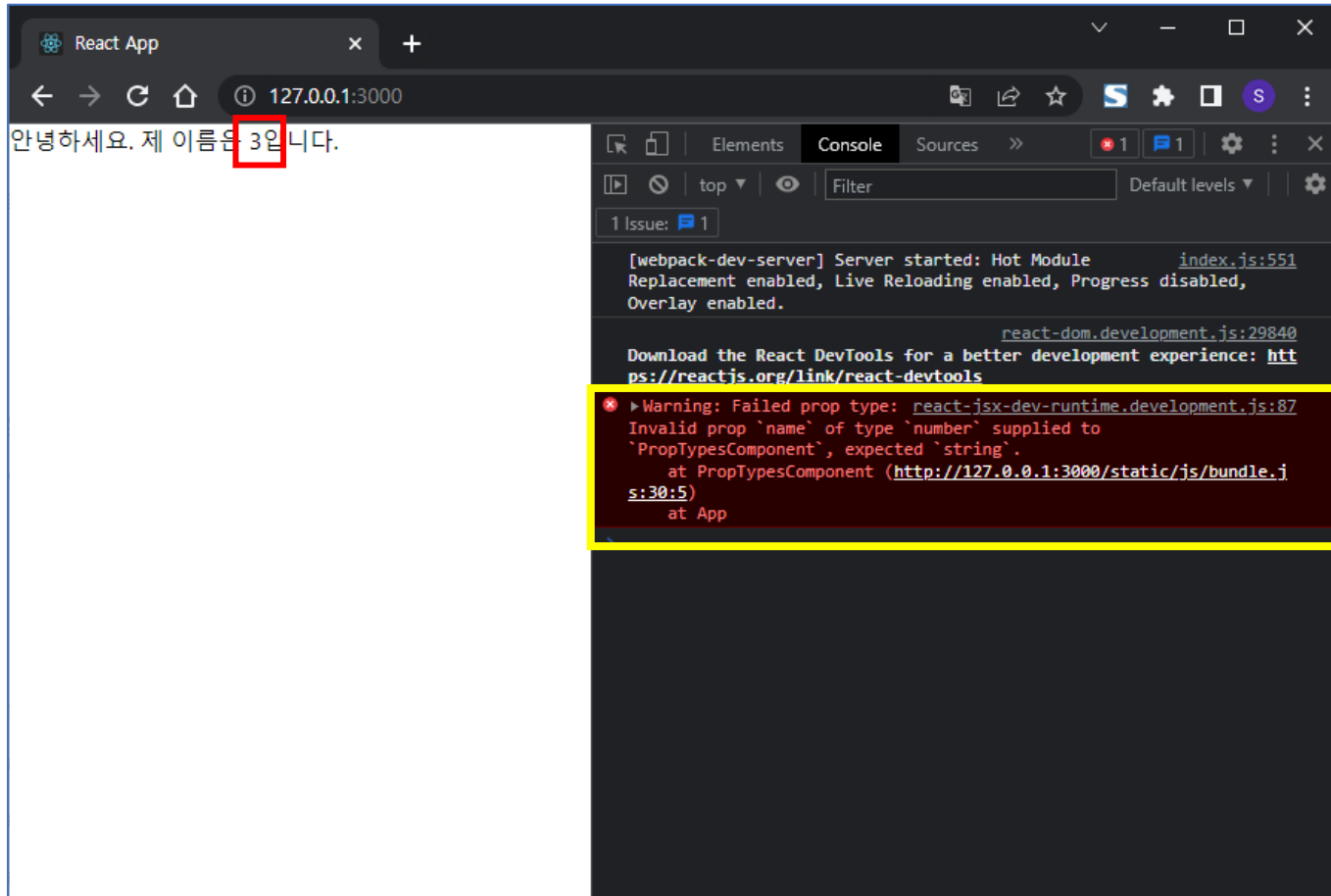
function App(){
  return (
    <PropTypesComponent name={3}/>
  );
}

export default App;
```



# propTypes - 타입 지정

## ❖ 실행 결과



# propTypes - 필수 프로퍼티 지정

## ❖ PropTypesComponent.js 수정

```
import React from 'react';
import PropTypes from 'prop-types';

function PropTypesComponent({name, favoriteNumber}){
  return(
    <div>
      안녕하세요. 제 이름은 {name}입니다.<br/>
      제가 좋아하는 숫자는 {favoriteNumber}입니다.
    </div>
  );
}

PropTypesComponent.propTypes = {
  name : PropTypes.string,
  favoriteNumber : PropTypes.number.isRequired
}

export default PropTypesComponent;
```

# propTypes - 필수 프로퍼티 지정

## ❖ App.js

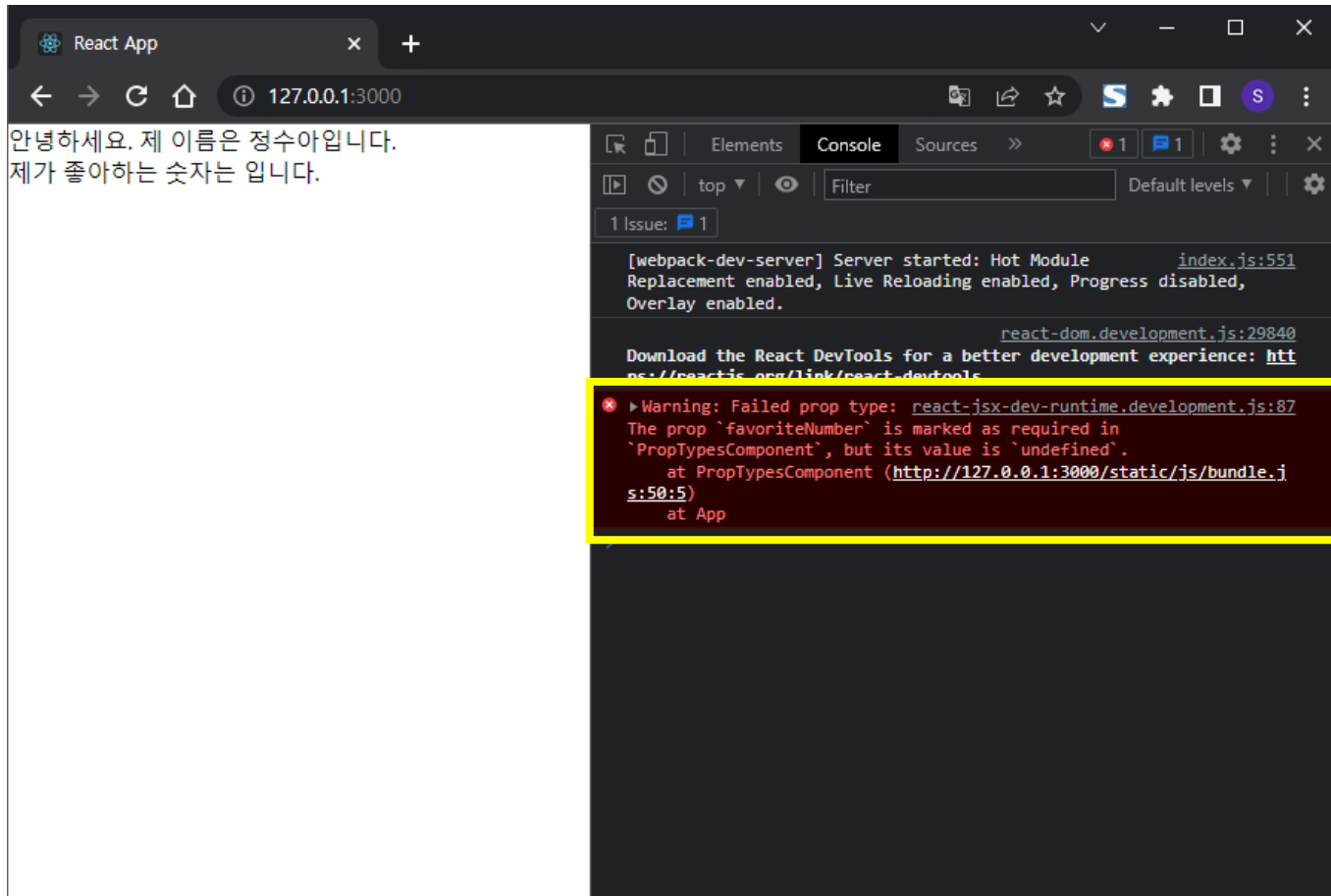
```
import React from 'react';
import PropTypesComponent from './PropTypesComponent';

function App(){
  return (
    <PropTypesComponent name="정수아"/>
  );
}

export default App;
```

# propTypes - 필수 프로퍼티 지정

## ❖ 실행 결과



# propTypes - 필수 프로퍼티 지정

## ❖ App.js 수정

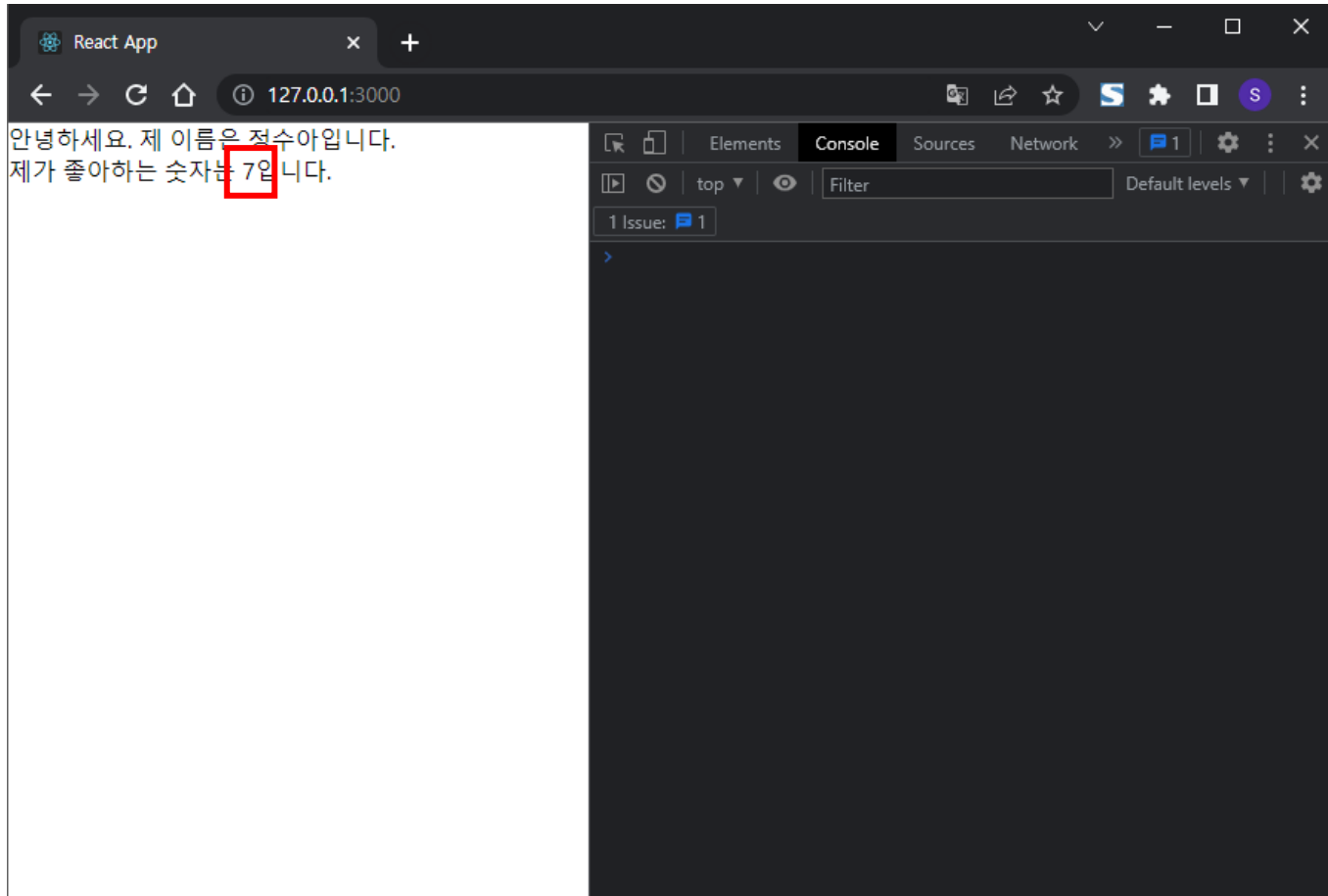
```
import React from 'react';
import PropTypesComponent from './03/PropTypesComponent';

function App(){
  return (
    <PropTypesComponent name="정수아" favoriteNumber={7}/>
  );
}

export default App;
```

# propTypes - 필수 프로퍼티 지정

## ❖ 실행 결과



**THANK 😊 YOU**