



리액트와 함께 장고 시작하기 / 리액트

# 속성값 타입 및 디폴트값 정의하기

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

# 각 속성값에 대한 타입 명시 및 필수 여부 지정하기 → 생산성

TypeScript와 같은 정적 언어를 사용하거나,  
prop-types 패키지를 통해 속성값에 타입을 지정할 수 있습니다.

설치> yarn add prop-types

지원 타입

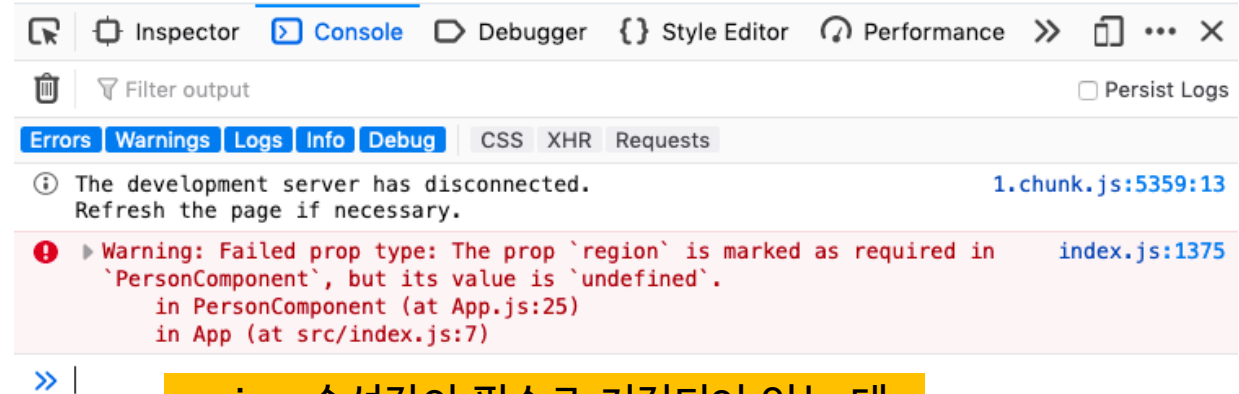
<https://github.com/facebook/prop-types#usage>

# 각 속성값에 타입 및 디폴트값 지정

```
class Person extends React.Component {
  static propTypes = {
    name: PropTypes.string.isRequired,
    age: PropTypes.number.isRequired,
    region: PropTypes.string
  };

  static defaultProps = {
    region: 'Seoul'
  };

  render() {
    const {name, age, region} = this.props;
    return (
      <div>
        안녕. 나는 {name}이야. {age}살이지.
        {region}에 살아.
      </div>
    );
  }
}
```



>> | region 속성값이 필수로 지정되어 있는 데, 지정되지 않는다면, 위와 같은 경고 발생

# 함수형 컴포넌트에서의 적용

```
function Person({ name, age, region }) {  
  return (  
    <div>  
      안녕. 나는 {name}이야. {age}살이지.  
      {region}에 살아.  
    </div>  
  );  
}
```

```
Person.defaultProps = {  
  region: 'Seoul'  
};
```

```
Person.propTypes = {  
  name: PropTypes.string.isRequired,  
  age: PropTypes.number.isRequired,  
  region: PropTypes.string  
};
```

## 또 다른 예시

```
function ThemedButton({ theme, label, ...restProps }) {  
  return (  
    <button className={`btn btn-${theme}`} {...restProps}>  
      {label}  
    </button>  
  );  
}
```

```
ThemedButton.defaultProps = {  
  theme: 'primary',  
  label: 'Empty Label'  
};
```

# recompose의 고차 컴포넌트 활용

Higher-Order Components → HOC

설치> yarn add recompose

```
import { defaultProps } from 'recompose';

const withDefaultProps = defaultProps({
  theme: 'primary',
  label: 'Empty Label'
});

const ThemedButton = withDefaultProps((props) => {
  const { theme, label, ...restProps } = props;
  return (
    <button className={`btn btn-${theme}`} {...restProps}>
      {label}
    </button>
  );
});
```

Life is short.  
You need Python and Django.

I will be your pacemaker.

