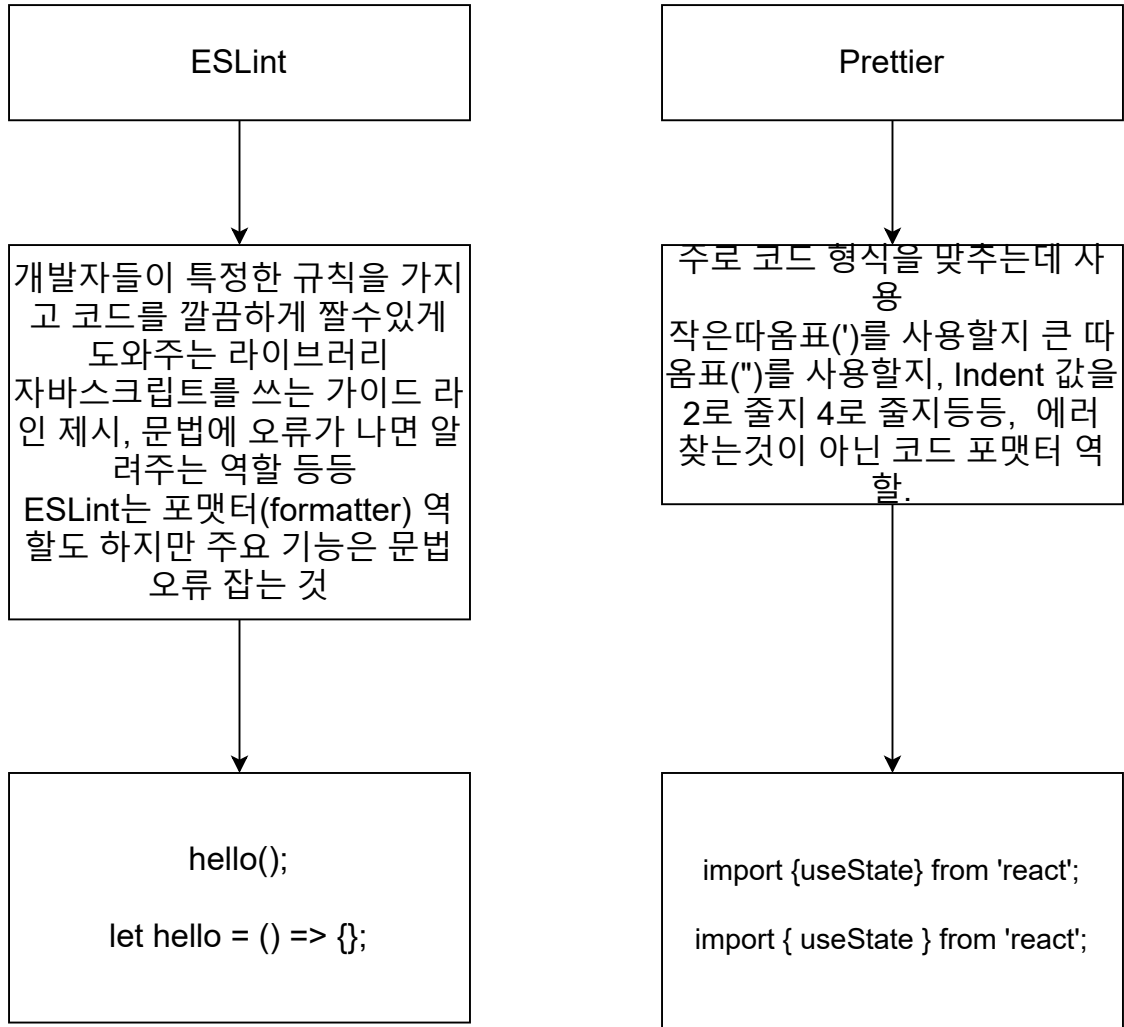


테스팅에 도움을 주는 모듈 설치

테스팅할 때 `matcher`를 알맞게 쓰는지 확신이 들지 않을 때가 있으며, 코드의 형식이나 자바스크립트 문법 등을 올바르게 쓰지 못할 때가 있습니다. 그러한 부분을 도와주는 모듈을 설치해주는 시간을 갖겠습니다.



먼저 ESLint 설정

ESLint 익스텐션 설치

이미 Create React App 으로 리엑트를 설치할 때 기본 eslint가 설정되어 있습니다. 하지만 이 상태로는 VS Code에서 바로 에러 확인 할 수 없고 앱을 시작했을 때 터미널 상에서 볼 수 있습니다.

eslint 설정 파일 생성,
package.json에 eslintConfig 부
분 지우고 .eslintrc.json 파일 생
성

이렇게 하면 PROBELMS 탭에
서 ESLint 에서 주는 기본적인
경고들을 확인 할 수 있습니다.

```
{  
  "extends": [  
    "react-app",  
    "react-app/jest"  
  ]  
}
```

이제는 Testing을 도와주는
ESLint를 설치해보겠습니다.

ESLint Testing Plugins 설치

Plugins 란 ?

eslint에서 기본으로 제공하지 않는 다양한 규칙을 플러그인을 통해 사용할 수 있습니다.
예를 들어서 react에 관련된 린트설정을 위해서는 eslint-plugin-react를 사용하면 되며, react hooks에 관련된 규칙을 적용시켜주려면 eslint-plugin-react-hooks를 사용하면 됩니다.

npm install
eslint-plugin-testing-library
eslint-plugin-jest-dom

testing-library
render로 Dom 그리는 부분

jest-dom
expect-matcher로 테스트

내부 설정해주기

plugins 항목 : 플러그인 추가
추가할 때,
eslint-plugin- 부분 생략가능

extends 항목 :
플러그인을 추가 한 후에 **규칙을 정해줘야**
사용가능합니다.
그래서 extends 항목에 사용하고자 하는 규
칙을 설정합니다.
vue, angular, react 중에 react 를 위한 규칙
recommended는 추천이 되는 걸 사용
만약 **규칙을 변경**하고자 할 때는 rule 항목
추가

```
{
  "plugins": [
    "testing-library",
    "jest-dom"
  ],
  "extends": [
    "react-app",
    "react-app/jest",
    "plugin:testing-library/react",
    "plugin:jest-dom/recommended"
  ]
}
```

lint가 잘 작동하는지 확인

```
const lintTest = screen.getByRole('button', {
  name: 'lintTest',
});

expect(lintTest.textContent).toBe('lintTest');
```

Prettier 설치 및 설정

Prettier는 코드 형식을 맞추는데 사용을 합니다. 테스트를 위해서 특화된 것은 아니지만 ESLint와 함께 자주 사용하기에 Prettier도 설치해보겠습니다.

