<Creating React Project>

3-1) Create React App

3-2) ESLint

3-3) Prettier

3-4) husky

3-5) lint-staged

3-6) 리액트 컴포넌트 디버깅

1. Create React App

- CRA(Create React App) : [https://create-react-app.dev](https://create-react-app.dev/)

\* npx create-react-app tic-tac-toe

- npx : npm 5.2.0 이상부터 함께 설치된 커맨드라인 명령어

\* npx serve -s build로 띄우기.

\* npx를 실행할 때 마다 새로운 서버 패키지를 받아서 실행함.

\* npm으로 실행하면 (로컬에서 실행하면) 작고 못생겨지게 나옴.

\* 하지만, npx에서 실행하면 소스코드를 수정하면 자동으로 반영이 되게끔 실행된다.

- npm test : Jest가 실행되면서 테스트 환경이 구성된다.

- npm run eject : 리액트 프로젝트를 꺼내서 create-react-app의 기능을 제공받지 않을 때 쓰는 명령어. 현업에서는 잘 사용하지 않는다. 실제로 사용하면 리액트 앱에 들어있던 라이브러리들이 바깥으로 나오게 된다. 리액트를 실행할 때 편리한 react app의 도움을 받을 수가 없게 된다.

- npm start

\* react-scripts start

\* Starting the development server...

- npm run build

\* react-scripts build

\* Creating an optimized production build...

- npm test

\* react-scripts test

\* Jest 를 통해 test code 를 실행합니다.

- npm run eject

\* react-scripts eject

- npm install serve -g : serve 라는 패키지를 전역으로 설치한다.

- serve -s build : serve 명령어를 -s 옵션으로 build 폴더를 지정하여 실행한다. => -s 옵션은 어떤 라우팅으로 요청해도 index.html 을 응답하도록 한다.

2. ESLint : The pluggable linting uility for JavaScript and JSX.

- 코딩 스타일을 내부적으로 규정할 때 lint라는 툴을 사용해서 규칙을 설정하고, 잘못된 부분은 알려주는 역할을 한다.

- 모든 자바스크립트로 이루어진 프로젝트는 ESLint를 이용해서 코딩스타일을 미리 맞춰볼 수 있다.

3. Prettier : An opinionated(주관을 가지고) code formatter -> 코딩 스타일을 일관적으로, 좀 더 예쁘게 사용하게 하는 것.

4. husky : Git hooks made easy

5. lint-staged : Run linters on git staged files(실제로 커밋이 되서 커밋하려고 하는 파일들.)

- ESLint 와 husky를 연동해서 사용함.

- 예시 : .js 파일이 올라와있으면 eslint에 fix하고, git add 하겠다.

"lint-staged": {

"\*\*/\*.js": [

"eslint --fix",

"prettier --write",

"git add"

]

}

6. React Developer Tools : 개발모드