· 08/02

◉ React

→ React는 Client단의 Interaction이 많은 경우에 적합.

★ → React (framework (x)*
            (Library   (o)

IoC

→ platform, framework. Library

★

★ Framework ← 실행에 대한 제어
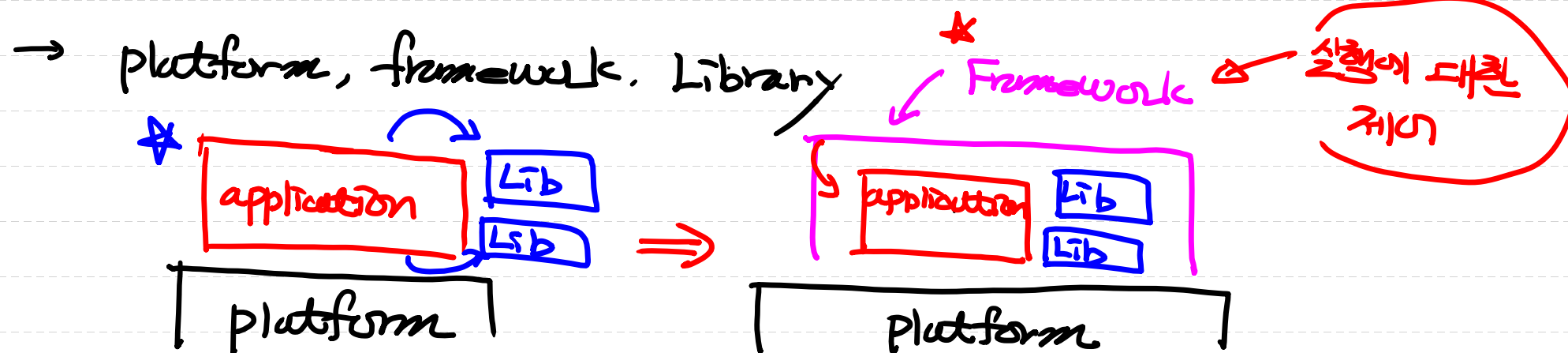
◉ 많은 Library를 이용해서 "UI rendering"
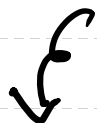
◎ React는 **component의 집합**

↳ React app을 구성하는 최소한의 단위

JS

검색창 → JS

Uso 정보 → JS

... ... → JS

과거 → React Hooks

① * class형 component ⟹ 복잡, 깊↑, 느려요

② 함수형 component ⟹ 간편, 깊↓, 빨라요

개선

함수형 component의
⇊
거의 대부분 사용
functional component

jQuery 는 쉽게 사용이 가능하지만

① maintenance (유지보수)

② 속도 문제
↳ 많은 양의 interaction 이 있는 경우

[DOM을 직접 제어하기 때문에
DOM이 변경을 따라서 repaint]

* Virtual DOM ←
를 이용해서 속도문제를 해결

↳ * React는 빠른가요?

* Node.js

Javascript → web browser 상에서 실행
chrome에는 V8 engine

* node.JS 는 * platform

* npm

(Node package Manager)

④ React App을 만들어 보아요

✗✓ Webpack module : Packaging module

✗✓ babel module : ES6를 기본으로 구현

저게 만든 아직도 많은 client들로 구형 browser 사용
ES6 지원을 으 하는 browser

그래서 ES6 → 구형 browser에서 동작출동있는
Javascript 코드로 변형

과거에는
먼저 설치하고
설치한후 React App 작성

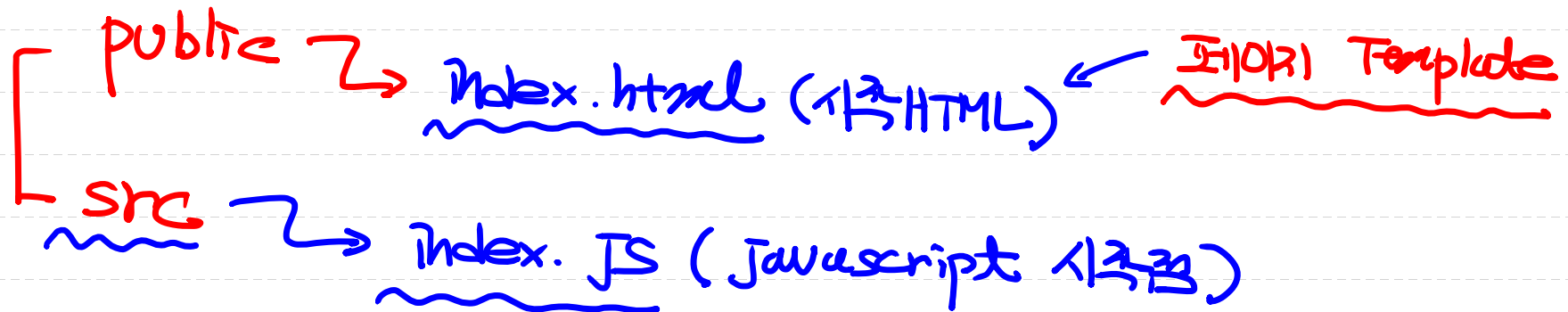Create-react-app 이용 → 편하게 구현 시작.

↳ package repository에 있어요

✗ npm install babel -g

npx → 보일러 플레이트

] → npx 이용 →

보일러 플레이트 되려게 나온것.

◉ 폴더 생성후 npx를 이용해 "React App 코드 생성"

```
┌─ public ⤵ → Index.html (시작HTML)  ← 페이지 Template
└─ src ⤵ → Index. JS ( Javascript 시작점)
```

"package.json" → 프로젝트에 대한 기본설정 확인

☆   ☆

오늘의 주 일

☐ 공부하기   delete

☐ 운동하기   delete

☑ ~~가구하기~~   delete

[할일 입력하기]   [등록]

◎ JSX (Javascript syntax extension) → Javascript 확장문법

↳ 이걸 이용해서 UI를 표현

✓ const str = "<H1> Hello world </H1>"; Javascript

✔ const str = <H1> Hello world </H1>; JSX (O)

Javascript ‾‾‾‾‾‾‾ HTML ‾‾‾‾‾ → Javascript

① JSX를 이용할 경우 여러 element 가
있다면 반드시 각각의 부모요소로 감싸줘야 해요.!!

~ 코드구현 ~