



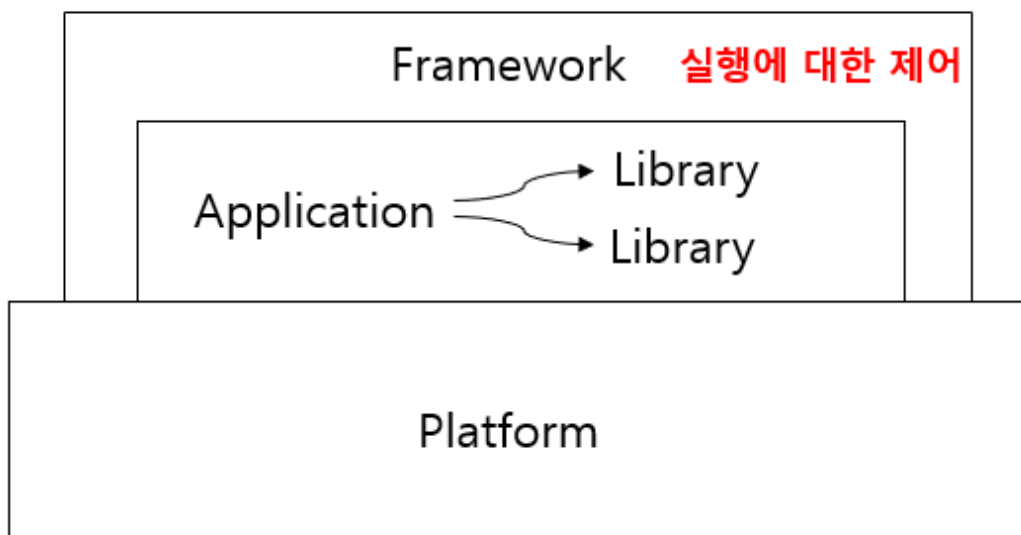
# DAY 1



React는 Client와의 상호작용(interaction)이 많은 경우에 적합하다.  
React는 **Libaray**이다. framework가 아니다.

## platform

다른 프로그램을 실행시켜줄 수 있는 프로그램을 뜻한다.



## React는 Component의 집합

React app을 구성하는 최소한의 단위

### 과거의 Component

1. **Class**형 Component
  - a. Code가 많고 복잡하다.
  - b. 속도가 느리다
  - c. 기능이 많다.

## 2. 함수형 Component

- Class형 컴포넌트에 비해 Code가 간결하다
- 속도가 빠르다.
- 기능상에 제약이 있다.

## 현재의 Component

### React Hook 출시



React 에서 기존에 사용하던 Class를 이용한 코드를 작성할 필요 없이, state와, 여러 React 기능을 사용할 수 있도록 만든 라이브러리

- React Hook이 나와서 함수형 컴포넌트의 기능이 대폭 개선이 되었다.
- Code도 간단, 속도도 빠르고, 기능도 많아졌다.
- 그래서 이제는 함수형 Component를 사용한다.

## JQuery

JQuery는 쉽게 사용이 가능하지만 단점이 있다.

1. maintenance(유지보수)
2. 속도 문제



JQuery는 DOM을 직접 제어하기 때문에 DOM이 변경될 때 마다 Repainting 작업을 실행한다. 즉, 화면의 대한 interaction이 발생할 때 마다 Repainting 작업을 실행하기 때문에 많은양의 interaction이 발생하면 속도가 느려진다.

## Virtual DOM

현재의 JQuery는 Virtual DOM을 이용해서 속도 문제를 해결할 수 있는 방안을 만들어줬다.



Virtual DOM (VDOM)은 UI의 이상적인 또는 “가상”적인 표현을 메모리에 저장하고 ReactDOM과 같은 라이브러리에 의해 “실제” DOM과 동기화하는 프로그래밍 개념입니다

## Node.js

JavaScript는 web browser 상에서 실행되는 것이다. 그럼 JS 코드를 실행시키는 주체가 있어야 한다.

- Chrome에는 V8 engine이 있다. → node.js 는 V8 Engine을 가지고 있는 하나의 Platform 이다.
  - node를 설치하면 다른 모듈들이 같이 깔리는데 그 중에 npm(node package manager) 이라는 것이 있다.
  - React 는 node 상에서 실행되는 것이 아니라 그 안에 있는 npm을 이용하는 것이다.
- 

## React

### **Webpack Module**

- packaging module

### **Babel Module**

- ES6를 기본으로 구현
- 하지만 아직도 많은 Client들은 구형 browser를 사용한다.
- ES6자원을 안쓰는 browser
- 그래서 ES6 → 구형 browser에서 동작할 수 있는 JavaScript 코드로 변형

과거에는 먼저 설치하고 설정한 후 React App을 작성했다.

### **create-react-app**

- 이것을 이용하면 자동으로 webpack,babel이 깔리고 훨씬 편하게 작업할 수 있다.
- npm을 이용해서 다운받아야한다.
- package repository에 있다.

```
npm install babel -g
```

node의 공용공간에 저장해 놓고 다른 프로젝트 들이 애를 공통으로 가져다가 사용할 수 있다.

하지만 문제가 있다 . 이렇게 사용하면 서버쪽에서 버전업의 여부를 확인할 수 없다.

한번 설치해놓고 계속 땡겨서 사용하기 때문에.

#### npx

- 보일러 플레이트(약간씩 변경이 되어서 자주 사용/업데이트되는 것)라는 것이 있는데, npx를 이용하면 항상 최신의 것을 편하게 사용할 수 있다.
- 보일러 플레이트 같은 것을 global로 설치하게 되면 최신버전을 사용할 수 없으므로 이런 경우에는 npx 를 사용하는 것이 좋은 것이다.
- npm의 단점을 보완해주기 위해서 나온 것이다.

```
C:\kakaocloud\React\Code\my-todo-app>npx create-react-app .
```

cmd에서 위에 명령어를 통해서 바탕화면에 만들어 놓은 my-todo-app 폴더에 npx를 이용해서 create-react-app을 실행시킨다.

#### npm run start

React 시작하기

## JSX(JavaScript Syntax Extension)

JavaScript 확장문법을 이용해서 UI를 표현한다.

### Ex Code

```
const str = "<h1>hello world!</h1>";  
  
const str = <h1>hello world!</h1>;
```



**JSX를 이용할 경우 여러 element가 있다면 반드시 하나의 부모요소로 감싸줘야 한다.**

## Javascript 문법

### .map()

```
// Array.prototype.map()  
const array1 = [1, 4, 9, 16];  
  
// map은 각각의 요소에 대해 callback function을 호출한다.  
const map1 = array1.map(function (x) {
```

```
    return x * 2;
  });

console.log(map1);
```

## Arrow Function

```
// Array.prototype.map()
const array1 = [1, 4, 9, 16];

// map은 각각의 요소에 대해 callback function을 호출한다.
const map1 = array1.map(function (x) {
  return x * 2;
});

const map2 = array1.map((x) => x * 2);
console.log(map2);
```

## Filter

```
// Array.prototype.filter()
// filter는 주어진 함수의 테스트를 통과하는
// 모든 요소를 모아 새로운 배열로 변환시켜준다.
const words = ['세종대왕', '이항', '강감찬', '샤인머스켓', '소리없는 아우성'];

// 글자 길이가 3자를 초과하는 데이터만 추려서 새로운 배열을 생성한다.
const result = words.filter((word) => word.length > 3);
console.log(result);
```

## Todo List 만들기

 Todo List

What is your main focus for today?

☐ Training

delete

☐ Eat

delete

☐ Study

delete

새로운 할 일을 입력하세요!

입력