

# 리액트 시작하기

프론트엔드 라이브러리

```
<div>  
  <h1>Counter</h1>  
  <h2 id="number">0</h2>  
  <button id="increase">+</button>  
</div>
```

```
var number = 0;
var elNumber = document.getElementById('number');
var btnIncrease = document.getElementById('increase');

btnIncrease.onclick = function() {
    number++;
    elNumber.innerText = number;
}
```

Angular, Ember, Backbone,  
Vue, React ...

지금 3대장

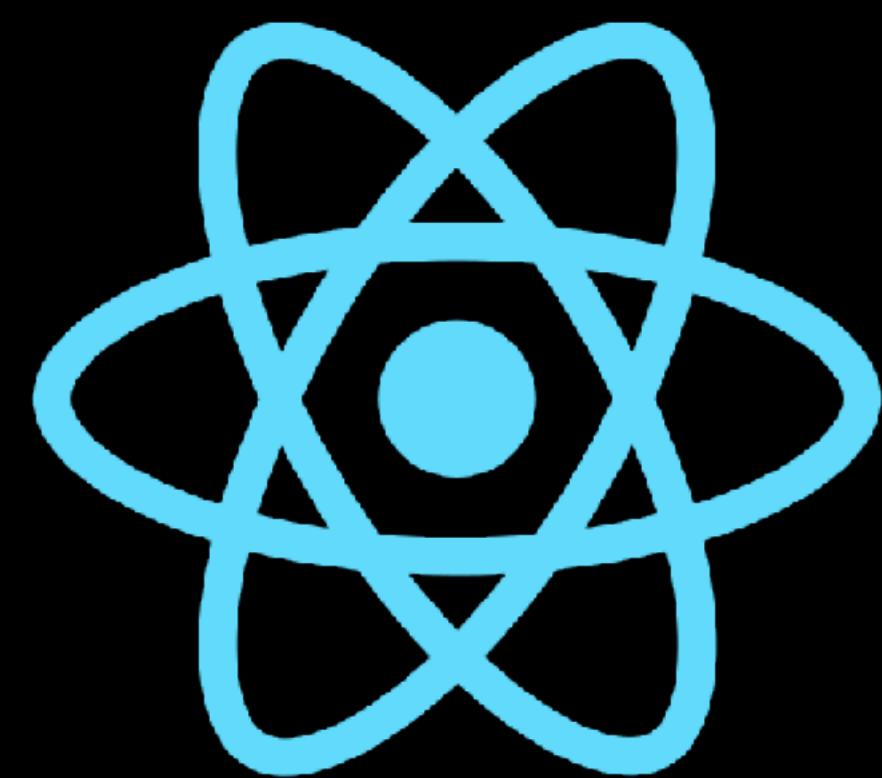
React, Angular, Vue

# Angular

React



Vue



***We built React to solve one problem:  
building large applications with data that  
changes over time.***

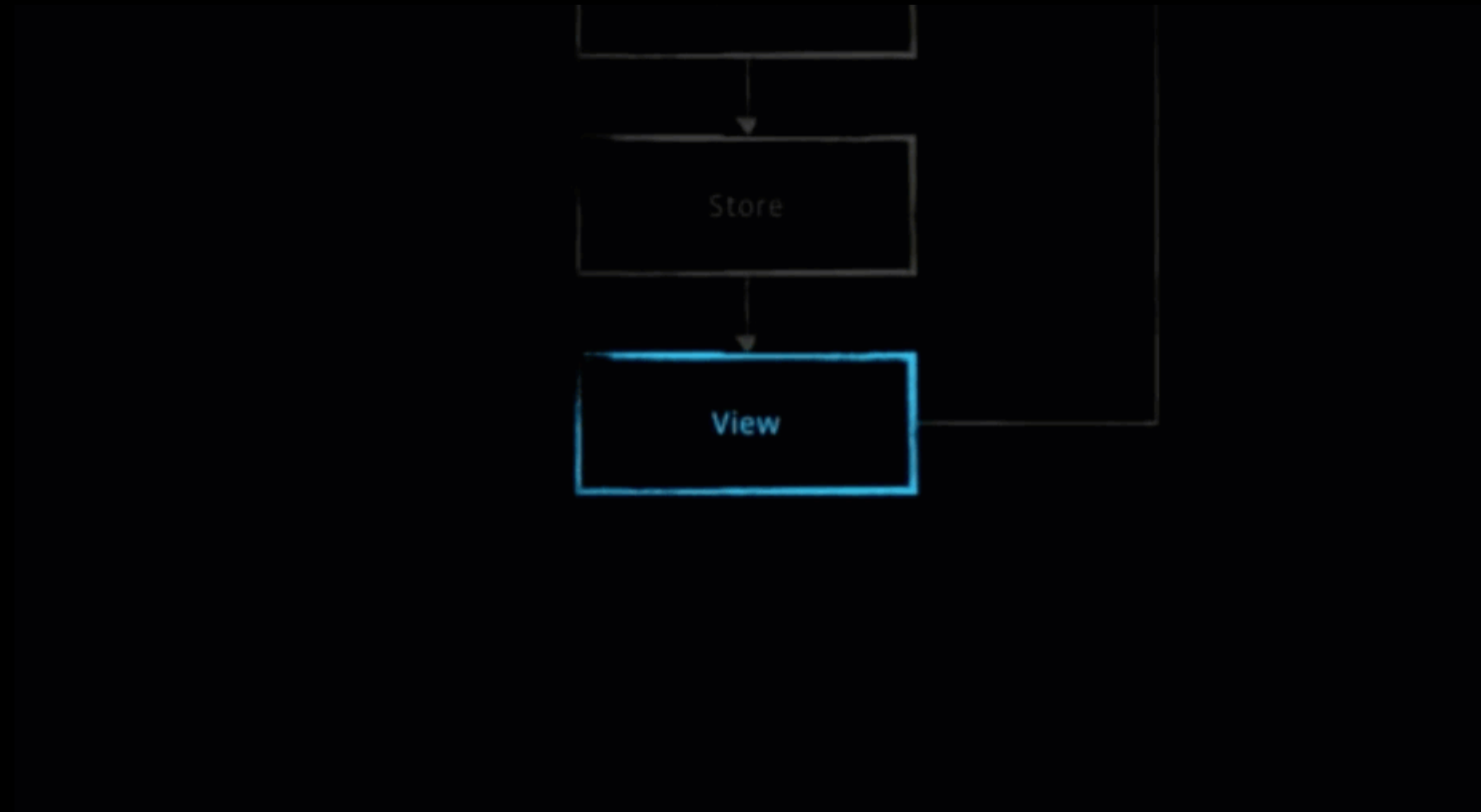
번역: 우리는 지속해서 데이터가 변화하는  
대규모 애플리케이션을 구축하기 위해 리액트를 만들었습니다.

MVC, MVVM, MVW 등을 사용하던  
기존 웹 프레임워크 / 라이브러리

공통점: Model

변화 (Mutation)

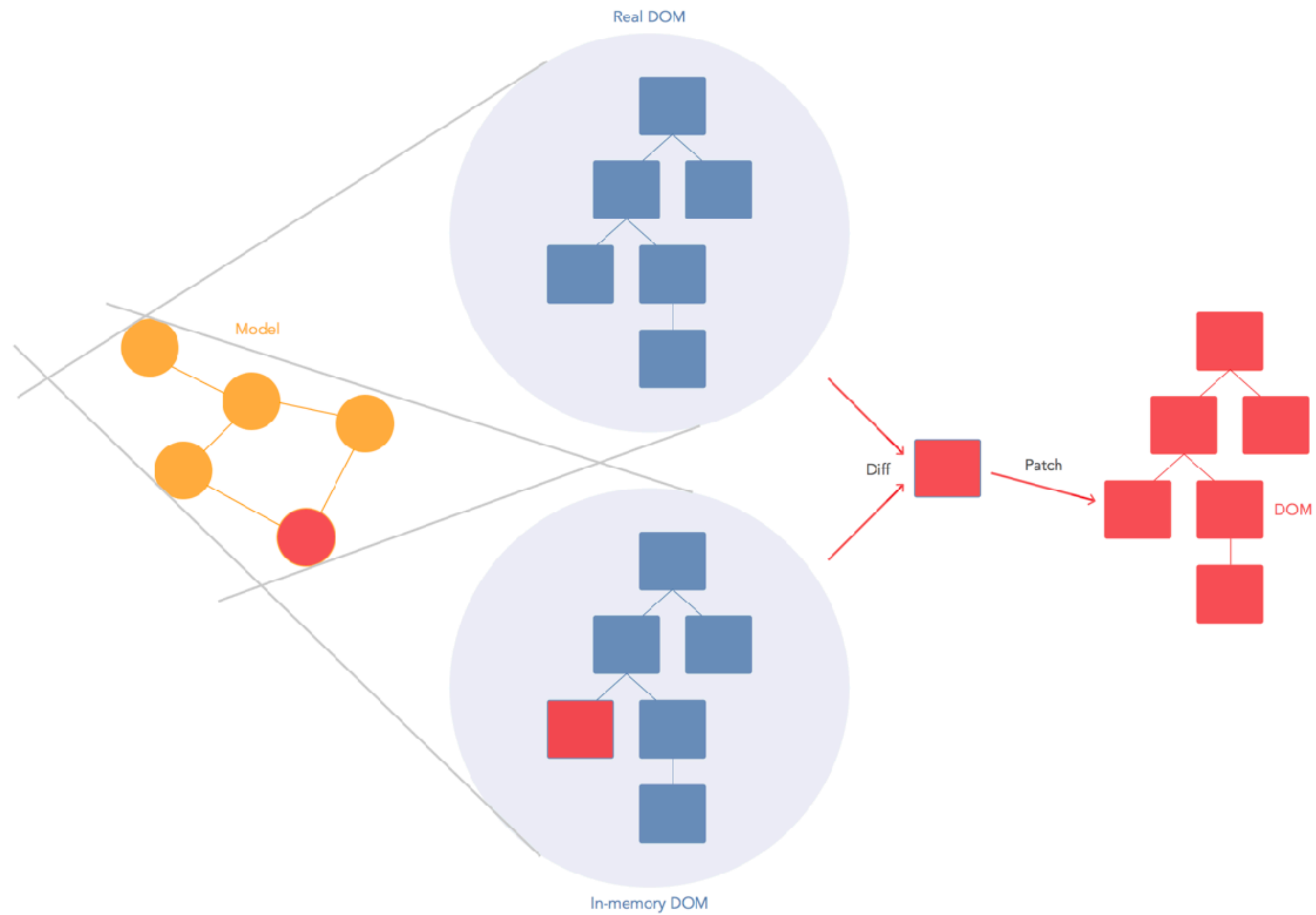




그냥 Mutation 을 하지 말자. 그 대신에, 데이터가 바뀌면  
그냥 뷰를 날려버리고 새로 만들어버리면 어떨까?



# Virtual DOM



# React and Virtual DOM

# 리액트에서만 Virtual DOM 쓰나?

Vue, Marko, Maquette, Mithril...

리액트를 특별하게 만드는 점은?

어마어마한 생태계

사용하는곳이 많다

Airbnb, BBC, Cloudflare, Codecademy, Coursera,  
Dailymotion, eBay, Twitch, Walmart, Yahoo,

**FACEBOOK**

한번 사용하면 좋아하게 된다!





Most Loved, Dreaded, and Wanted Frameworks, Libraries and Other Technologies



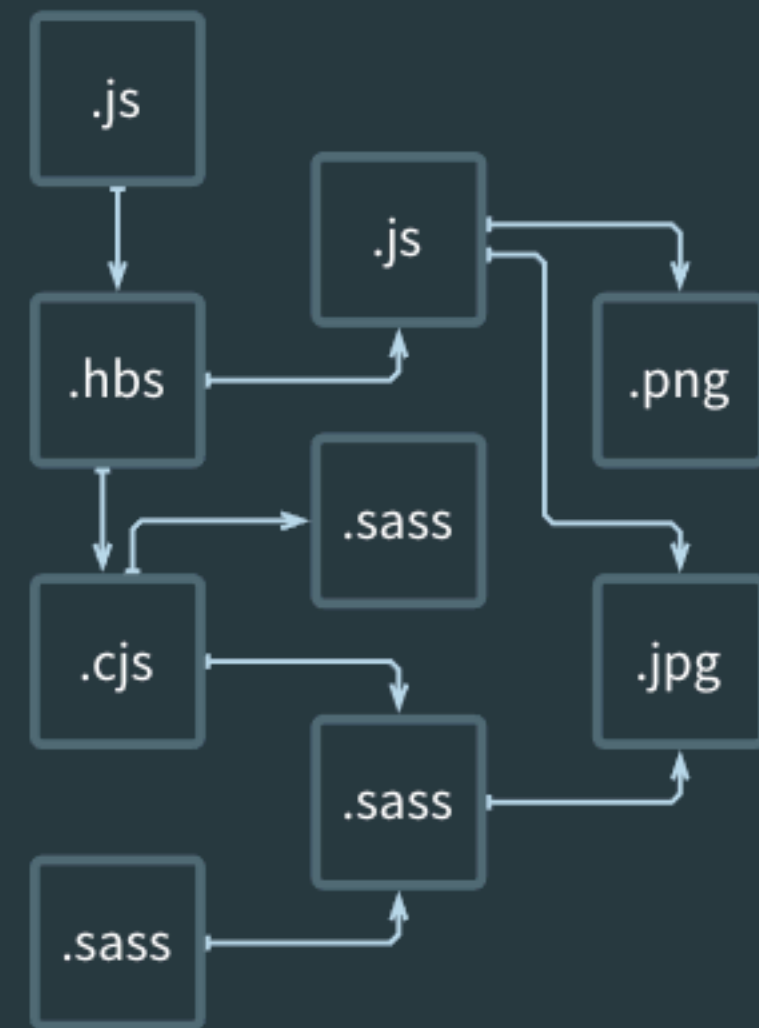
% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

React is the most loved among developers, whereas Cordova is the most dreaded. However, Node.js is the most wanted.

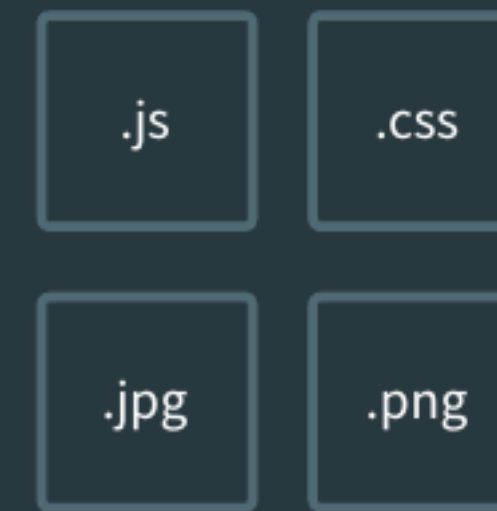
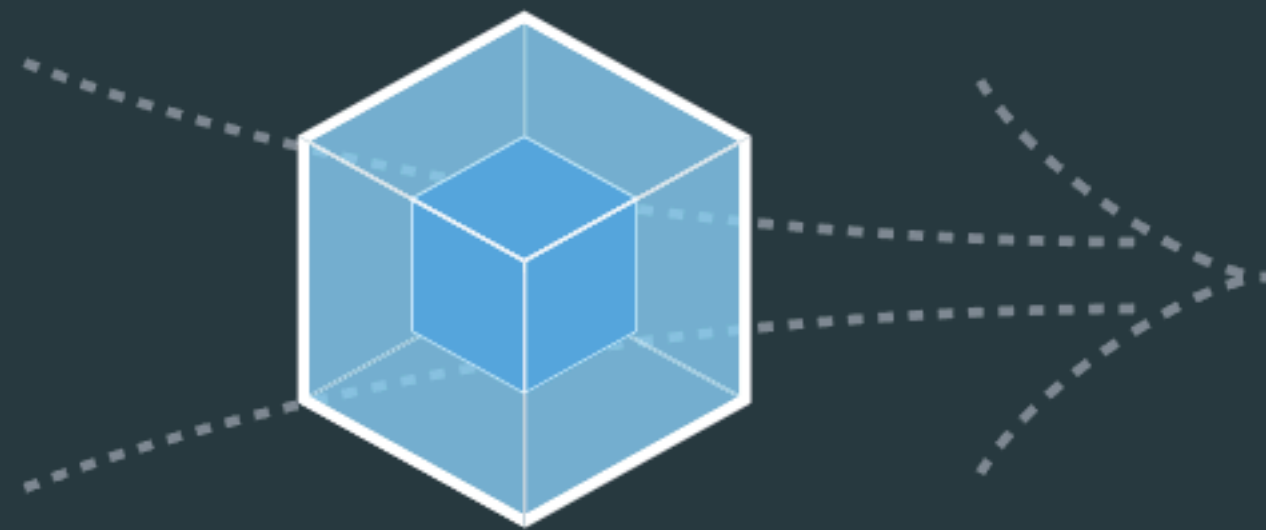
리액트 프로젝트 시작하기

# Webpack 과 Babel

bundle your scripts



MODULES WITH DEPENDENCIES



STATIC ASSETS

# Webpack

# Babel is a JavaScript compiler.

Use next generation JavaScript, today.

Put in next-gen JavaScript

```
var obj = {  
  shorthand,  
  method() {  
    return "😄";  
  }  
};
```

Get browser-compatible JavaScript out

```
var obj = {  
  shorthand: shorthand,  
  method: function method() {  
    return "😄";  
  }  
};
```

[Check out the REPL to experiment more!](#)

# Babel

**JSX**





HTML 이랑 비슷하지만,  
지켜야 할 규칙이 몇가지 있다

# Props 와 State

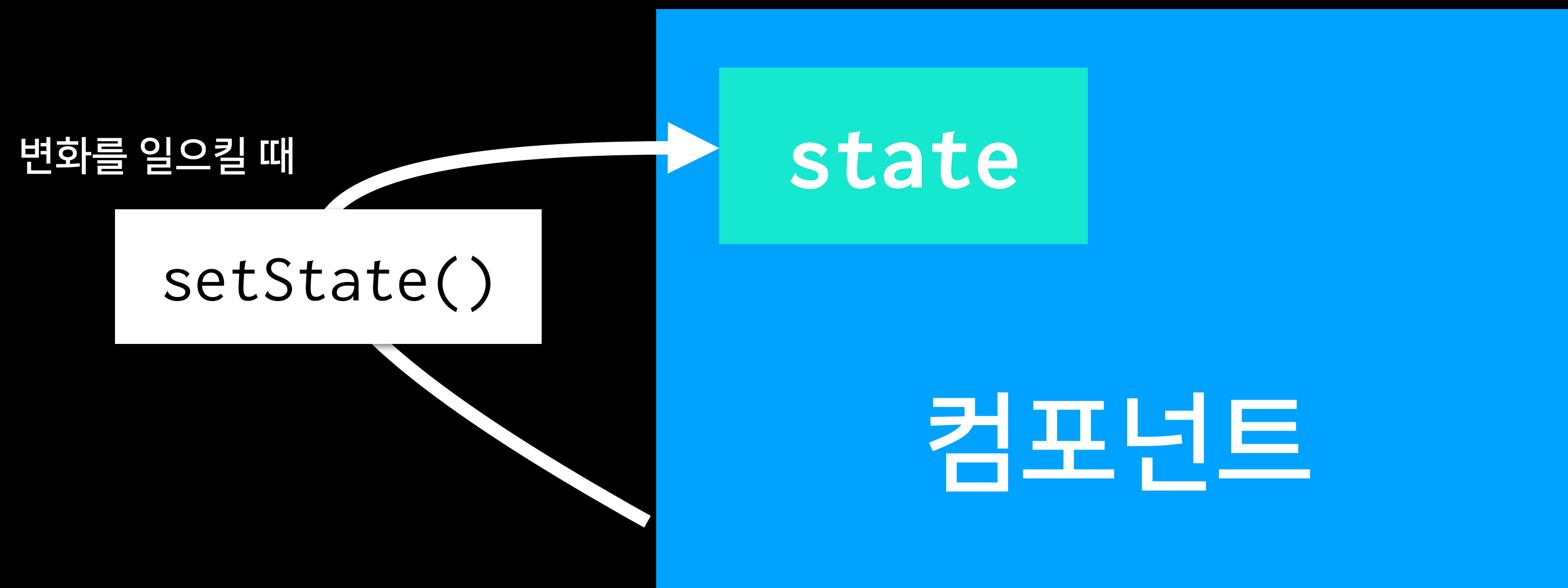
부모



props

자식

```
<Child value="value" />
```



state는 내부에서 변경 할 수 있다  
변경 할 때는 언제나 setState 라는 함수를 사용한다

# LifeCycle API

1. 나타날 때
2. 업데이트 될 때
3. 사라질 때

