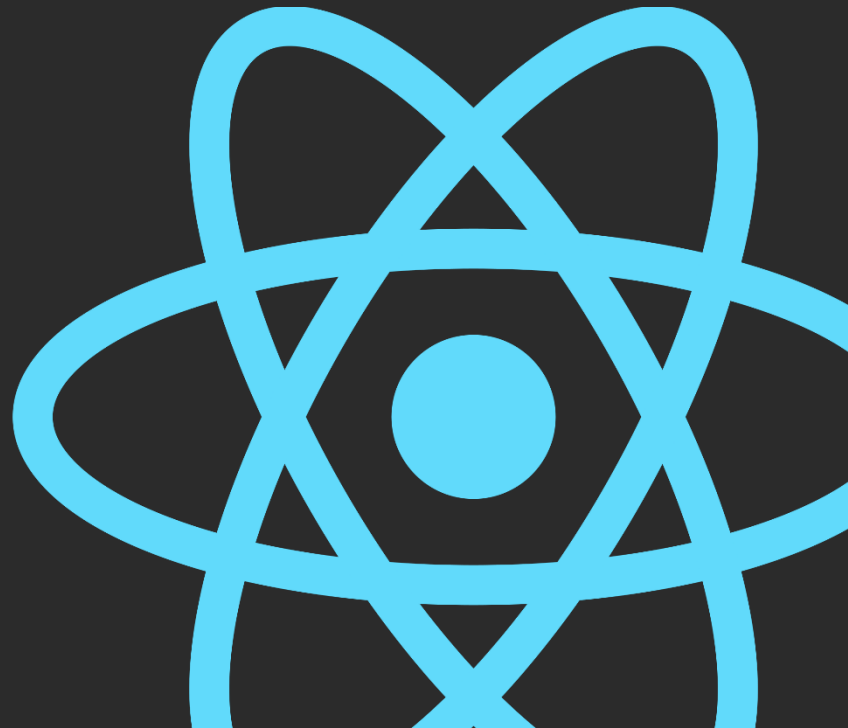
React.js

UI를 만들기 위한 JavaScript 라이브러리

#2. Component, Props, State



Contents.

01. Component에 대해 이해하고 활용할 수 있다.

02. Props에 대해 이해하고 활용할 수 있다.

03. State에 대해 이해하고 활용할 수 있다.

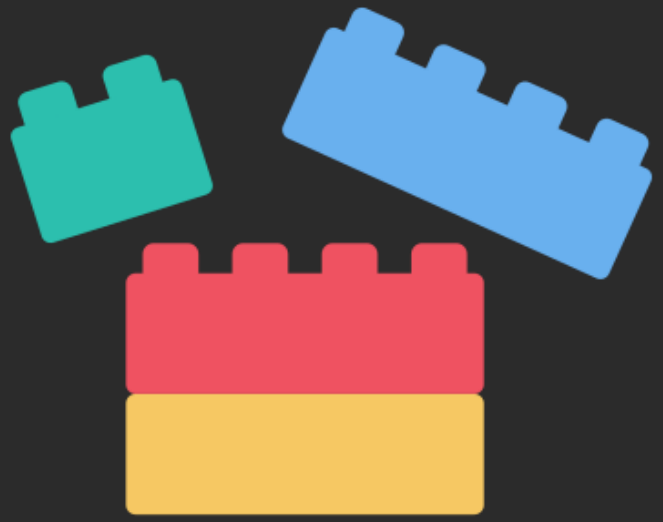
Component
(컴포넌트)

화면에서 UI 요소를 구분하는 최소 단위
(React에서 App을 이루는 가장 작은 단위)

Component 란?

=== Component
(컴포넌트)

Component 란?



컴포넌트 조립

하나의 웹 페이지 완성

Component 란?

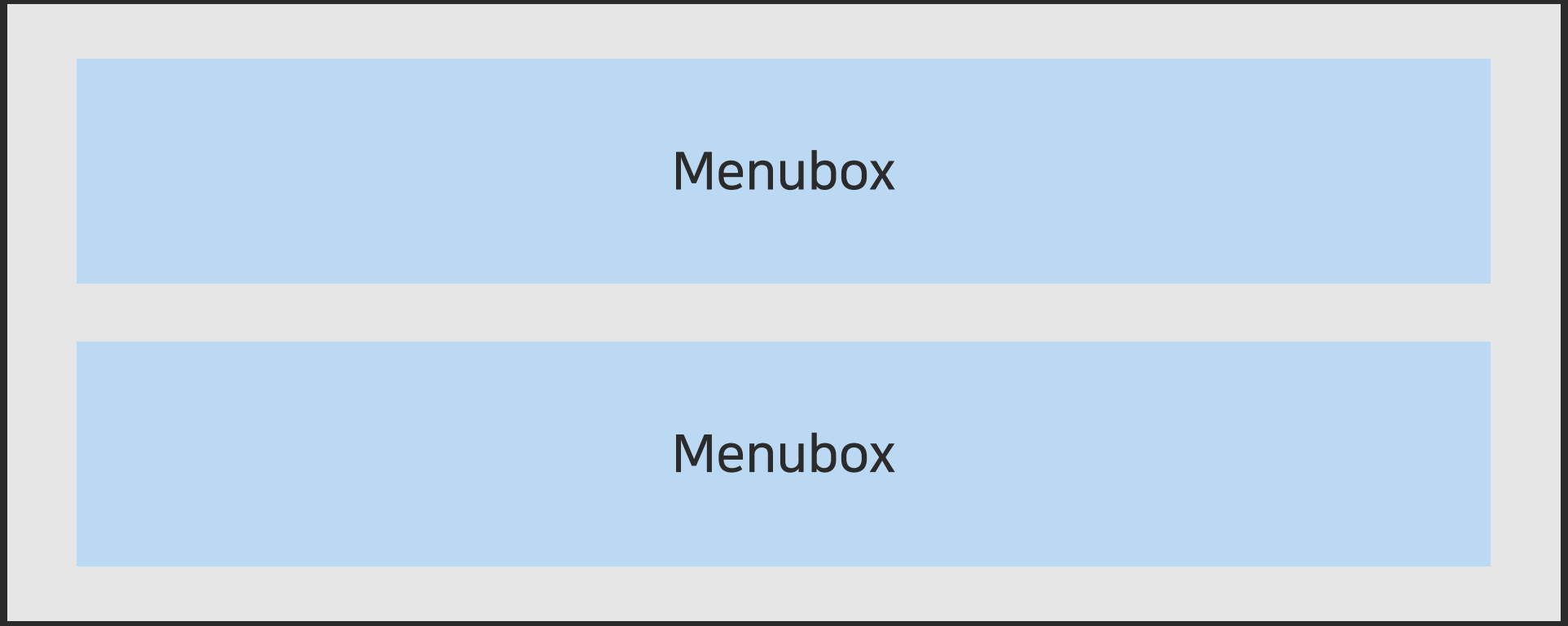
기존의 HTML 문서

Component 란?

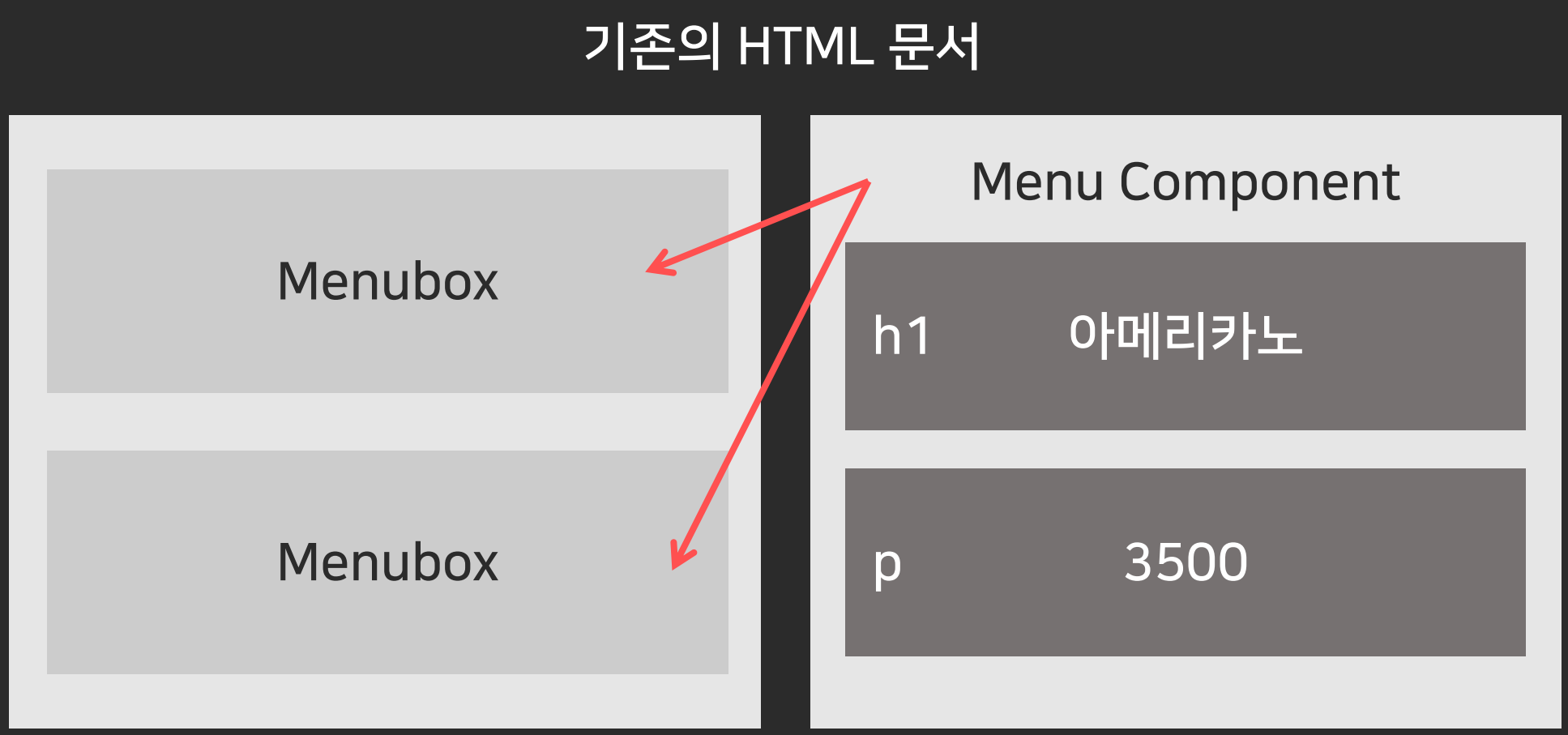
기존의 HTML 문서

Component 란?

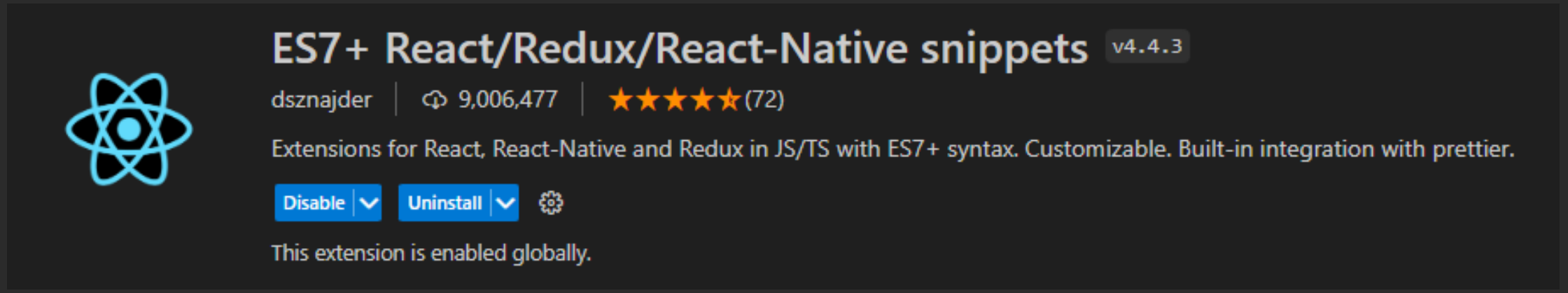
기존의 HTML 문서



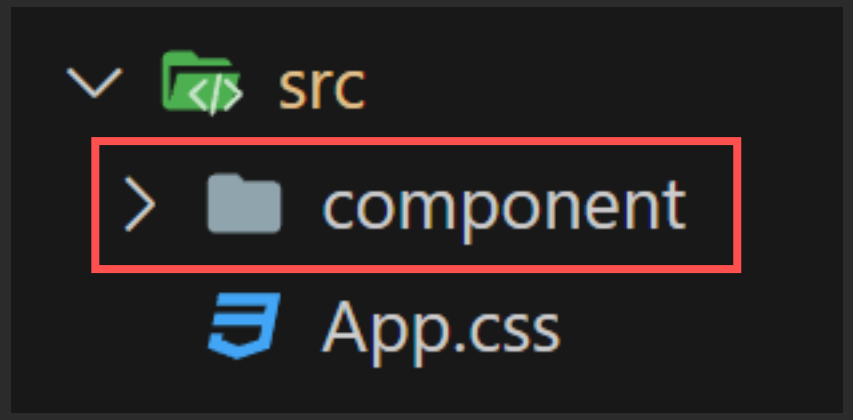
Component 란?



1. Extension 설치

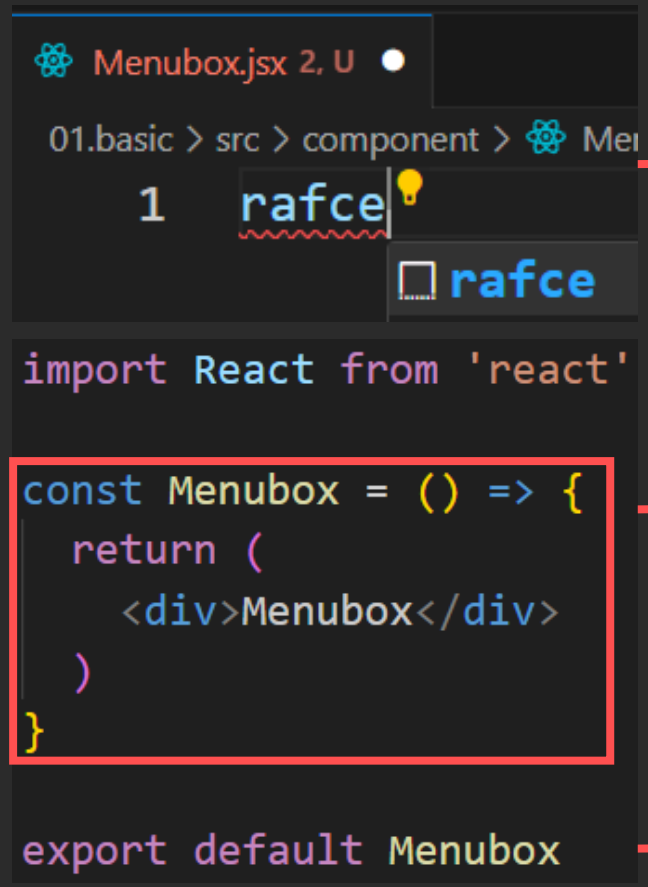


2. component 폴더 생성 & 파일 생성



component -> 파일이름.js 생성(대문자)
이 때, 주의할 점! 컴포넌트의 이름은 반드시 **대문자로 시작**할 것!

3. Component 파일 관리



컴포넌트 만드는 명령어 : rafce

함수형 컴포넌트 생성 완료!

컴포넌트 내보내기

4. 내보낸 컴포넌트 불러오기

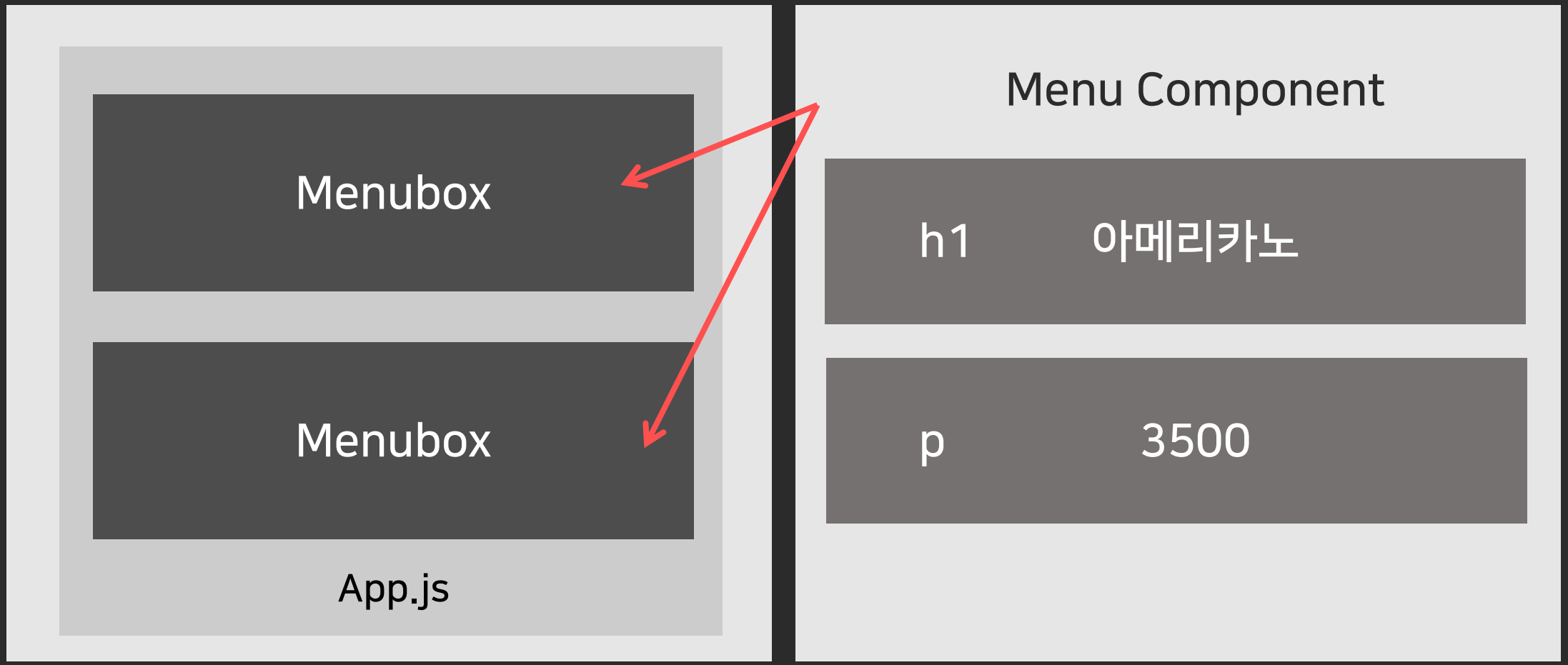
```
import './App.css';
import Menubox from './component/Menubox';

function App() {
  return (
    <div>
      <Menubox></Menubox>
    </div>
  );
}

export default App;
```

컴포넌트 불러오기

사용하기

App.js

Component 란?


Menubox
Menubox
App.js
Menu Component
h1 아메리카노
p 3500
index.html

Menu Component

Menubox

Menubox

App.js

index.html

내가 원하는 걸 묶어서
태그화(컴포넌트) 시킨다!

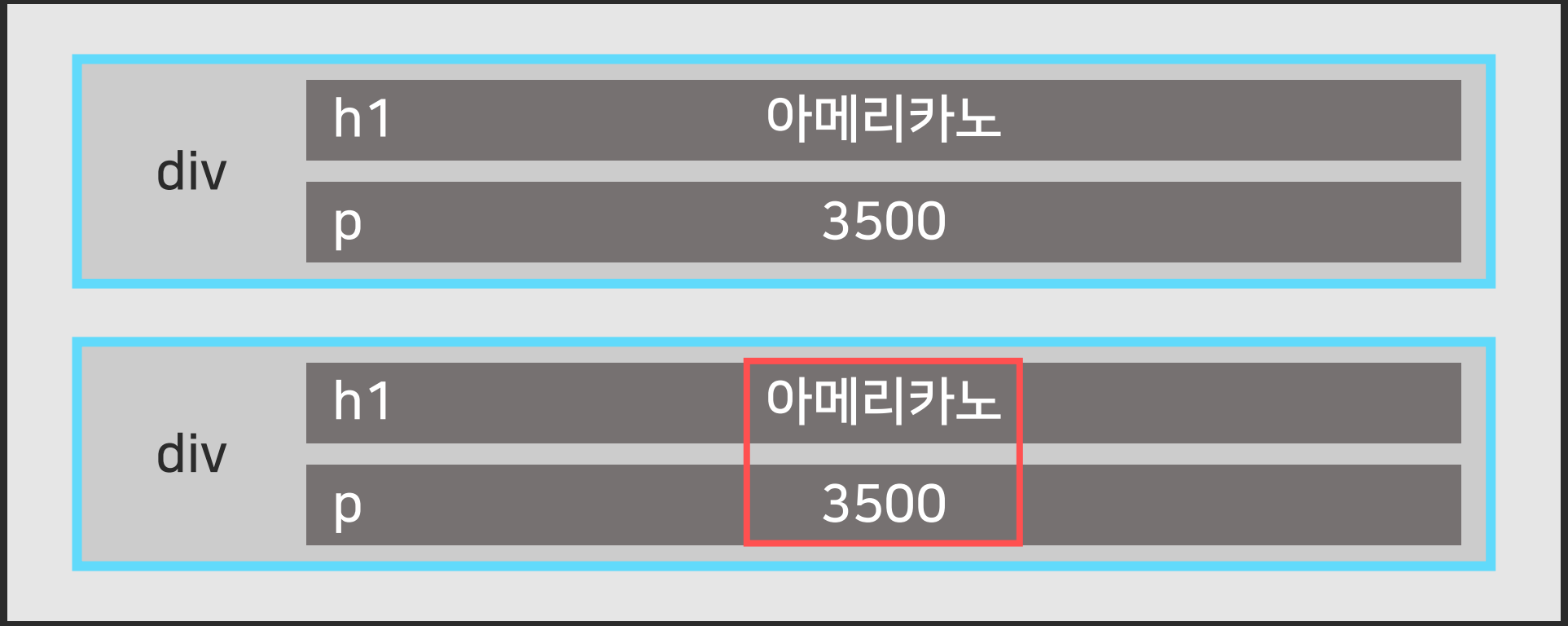
Component 사용 시 주의사항

Component의 이름은 반드시 **대문자**로 시작해야 한다

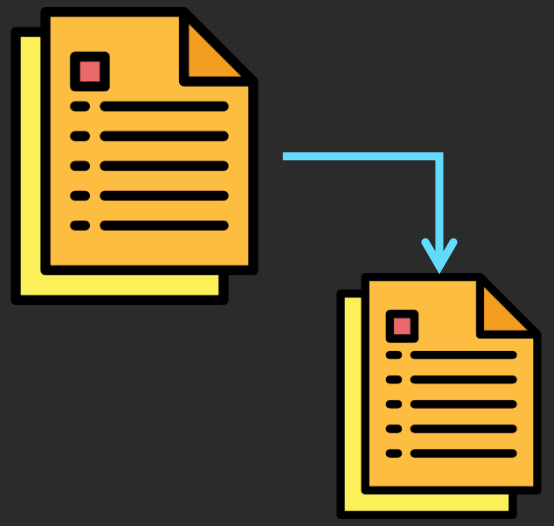
React가 Component와 일반 HETML 태그를 구분할 수 있어야 한다.
따라서, React에서는 일반 HTML 태그는 소문자로,
Component는 반드시 대문자로 시작해야 한다!

기존의 HTML 문서

기존의 HTML 문서



props 란?



props
(프로퍼티)

부모(상위) 컴포넌트가 자식(하위) 컴포넌트에 값을 전달할 때 사용한다
부모 컴포넌트에서 값을 입력하고, 자식 컴포넌트에서는 값을 읽어온다

props 사용 방법

```jsx
function App() {
  return (
    <div>
      <Menubox menuName="아메리카노"></Menubox>
      <Menubox></Menubox>
    </div>
  );
}
```

```jsx
function App() {

  let coffee = "아메리카노"

  return (
    <div>
      <Menubox menuName={coffee}></Menubox>
      <Menubox></Menubox>
    </div>
  );
}
```

문자열을 전달할 때는 작은따옴표를('') or 큰따옴표를("")
문자열 외의 값을 전달할 때는 중괄호({})를사용한다.

props 하위 컴포넌트

```jsx
function App() {

  let coffee = "아메리카노"

  return (
    <div>
      <Menubox menuName={coffee}></Menubox>
      <Menubox></Menubox>
    </div>
  );
}
```

상위 컴포넌트
(App.js)

menuName속성의 값 전달

props = {"menuName":"아메리카노"}

```jsx
const Menubox = (props) => {
  return (
    <div className='menuBox'>
      <p>{props.menuName}</p>
      <span>3500</span>
    </div>
  )
}
```

하위 컴포넌트
(Menubox.jsx)

컴포넌트 & props 예제

컴포넌트와 props를 이용해서 야구팀 대표선수를 소개하는 페이지를 만들어보자!

state

컴포넌트 내부에서 관리되는 변경이 가능한 데이터

숫자를 카운팅하는 웹 페이지 구현 - JS

```
let count = 0;

const handleClick = () => {
    count += 1;
}

const renderer = () => {
    const countText = document.getElementById("count-text");

    countText.innerText = `현재 count ? ${count}`;
}
```

state

컴포넌트 내부에서 관리되는 변경이 가능한 데이터

변수와의 차이점은?

state와 변수의 차이점

state는 일반 변수와 다르게
값이 변하면 화면이 자동으로 렌더링된다

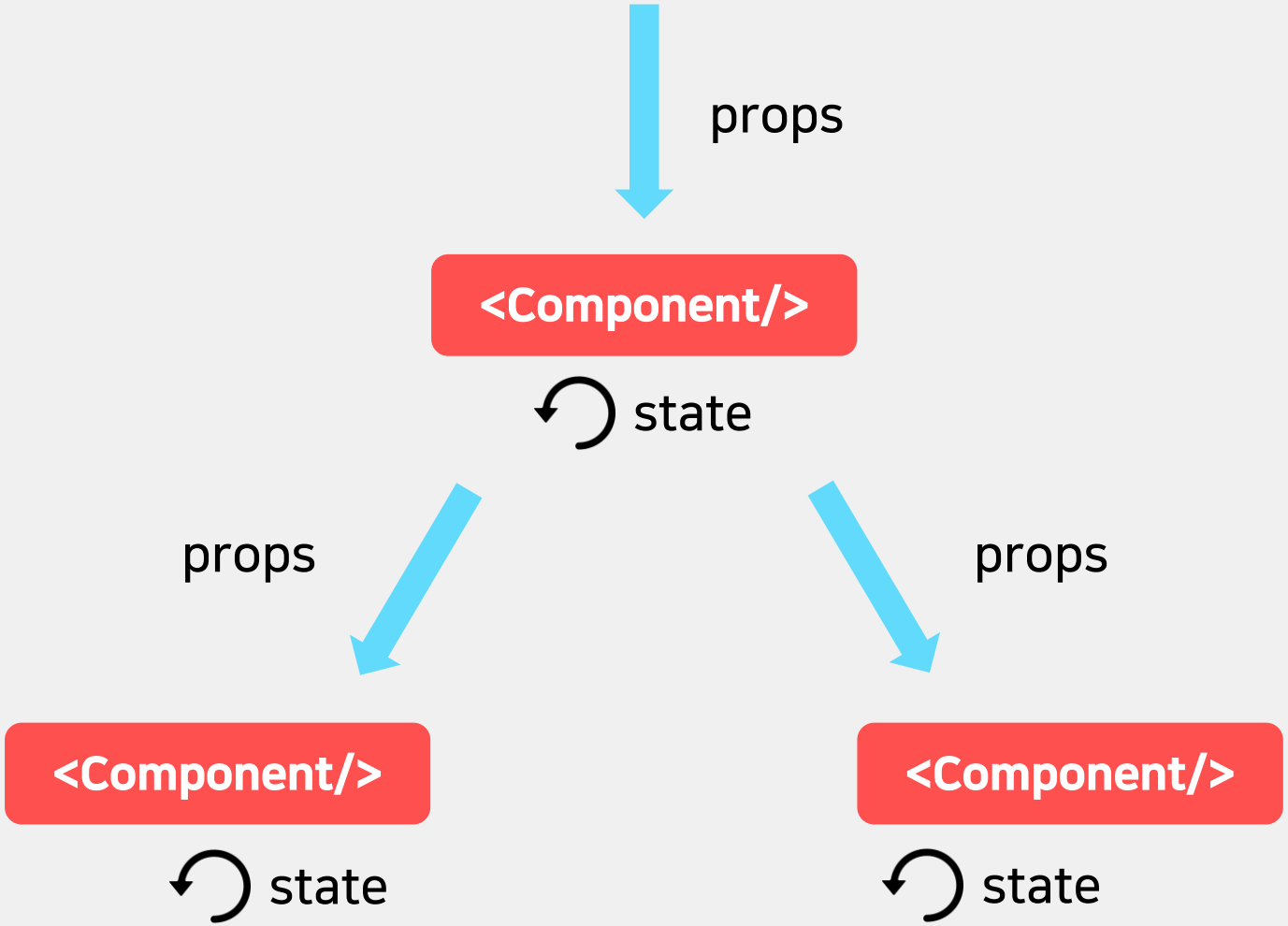
이 때, **setState()라는 함수**를 이용한다

state와 props의 차이점

props 를 전달받은 자식 컴포넌트에서는
수정이 불가하고 읽는 용도로만 사용할 수 있다 (Read Only)

별도로 변경해야 할 때, **setState()라는 함수**를 이용한다

React props & state



useState()

컴포넌트 내부에서 관리되는 변경이 가능한 state를
생성/변경을 처리하는 React Hooks 중 하나로
함수형 컴포넌트에서만 사용

state 란?

```
const [state, setState] = useState(초기값)
```

```
const [state, setState] = useState(초기값)
```

변수 이름

state 란?

state를 변경시켜주는 함수

```
const [state, setState] = useState(초기값)
```

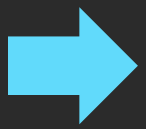
변수 이름

React state를 통해 화면이 보여지는 과정

React =>

Component
(컴포넌트)

생김새 동작



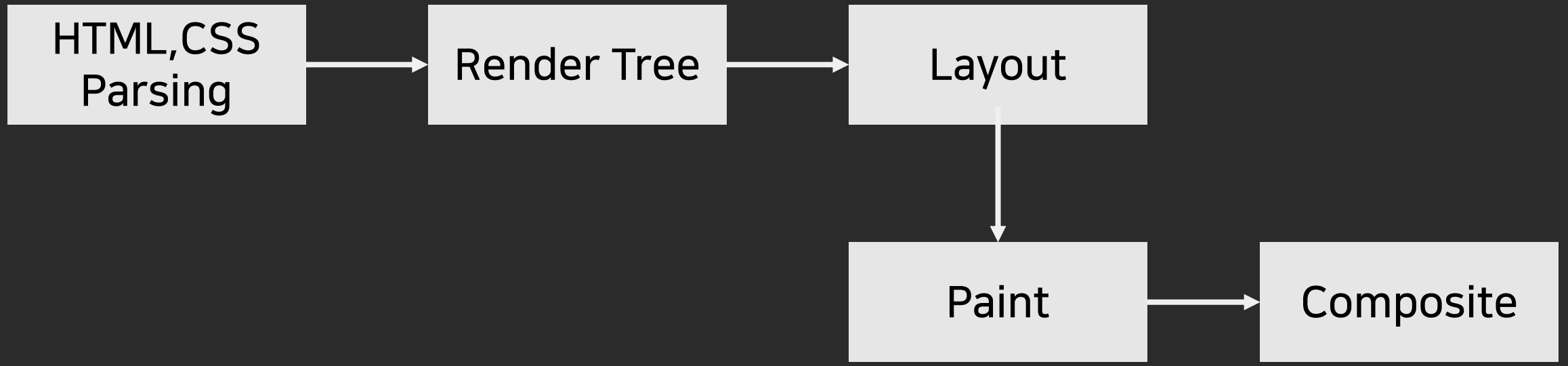
Rendering
(렌더링)

React는 Rendering을 통해 맨 처음 화면을 초기화하고

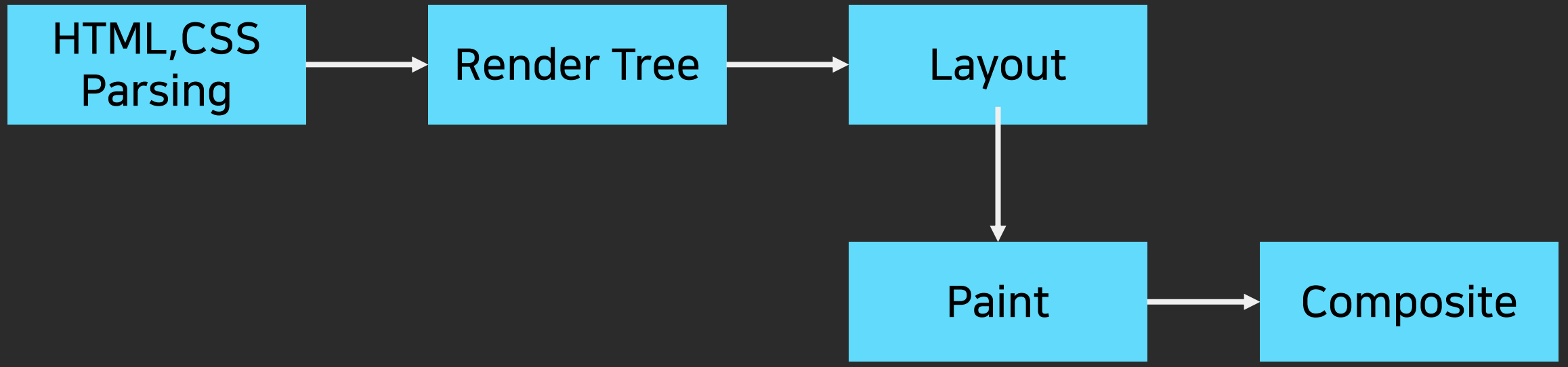
데이터가 변경이 일어났을 때 Re-rendering을 통해

화면을 그려준다!

기존 방식

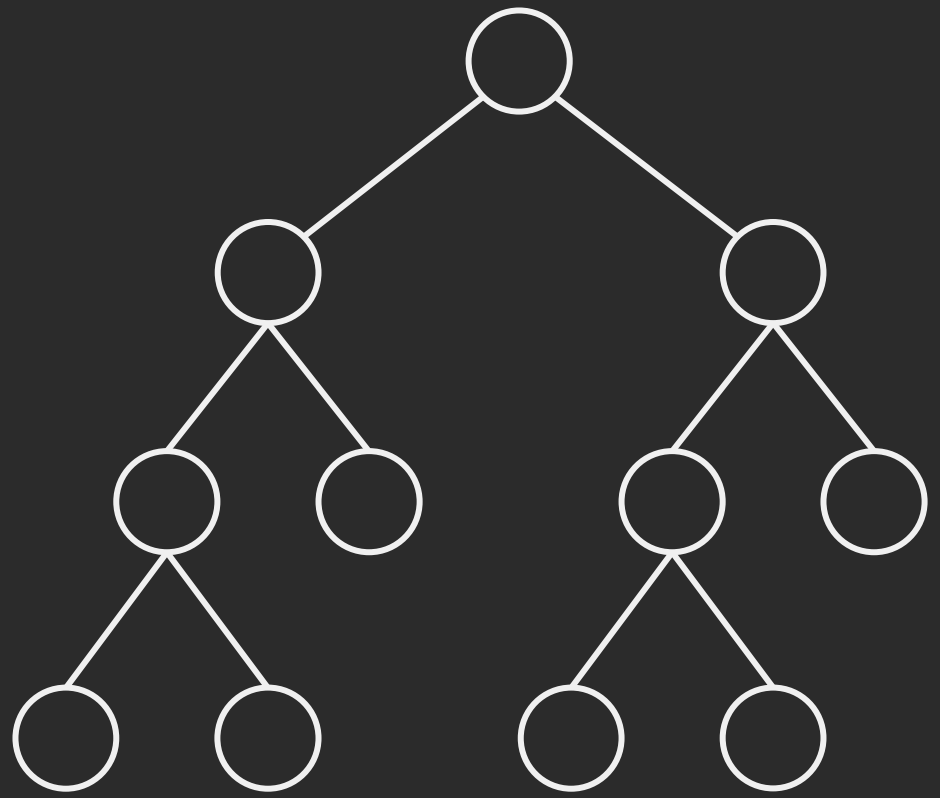


DOM 변화 발생



React state를 통해 화면이 보여지는 과정

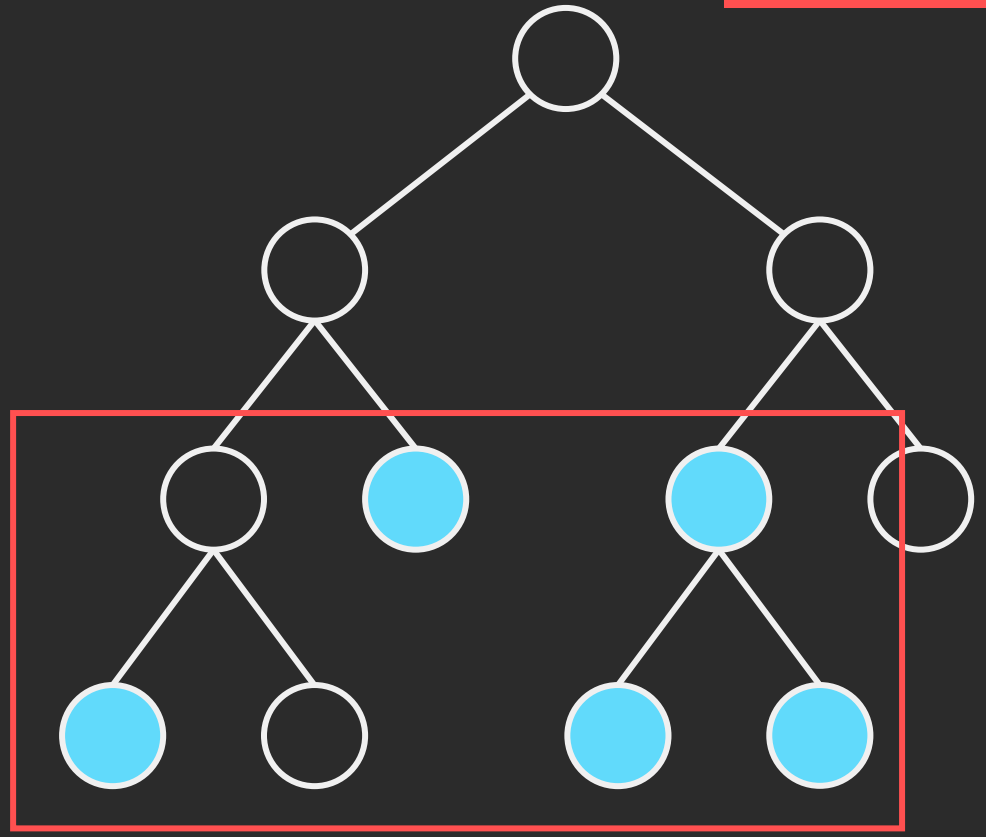
[이전 DOM 트리]



비교

[새로운 DOM 트리]

Virtual DOM



이전 DOM트리와 비교하여 최종 실제 DOM에 적용

업데이터될 DOM 노드

Virtual DOM

실제 DOM이 아닌 추상화된 DOM 트리를 구성한 구조
가벼운 사본 형태

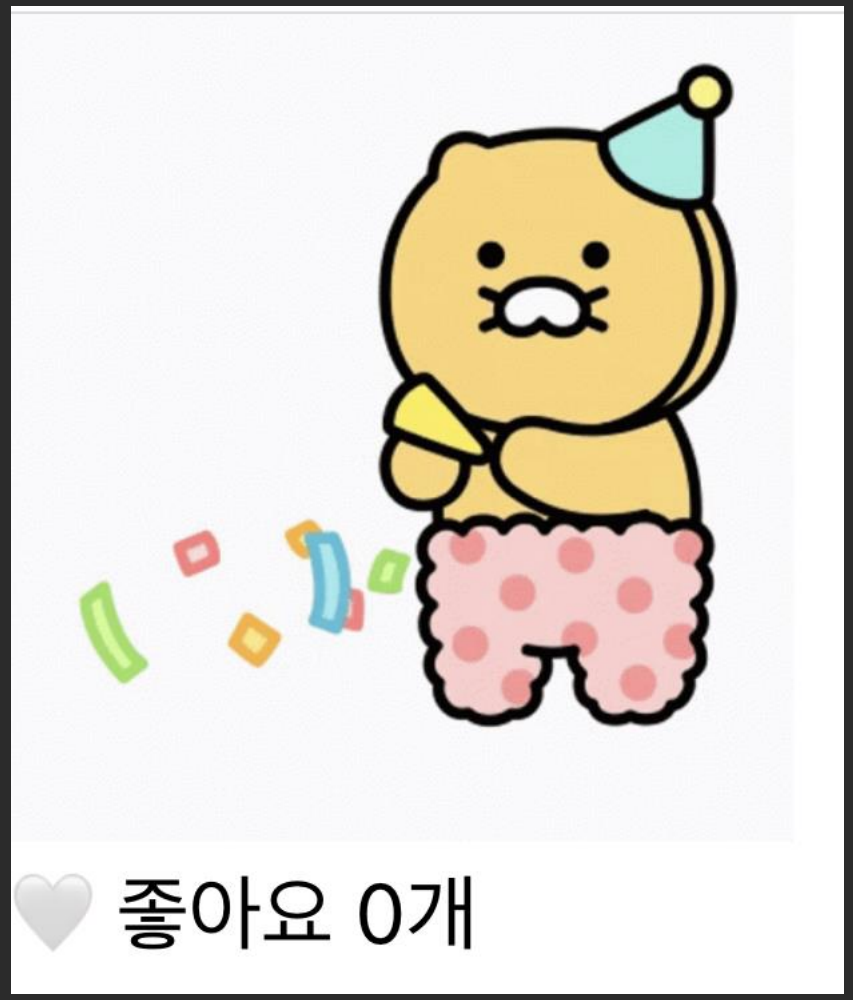
업데이트 3가지 절차

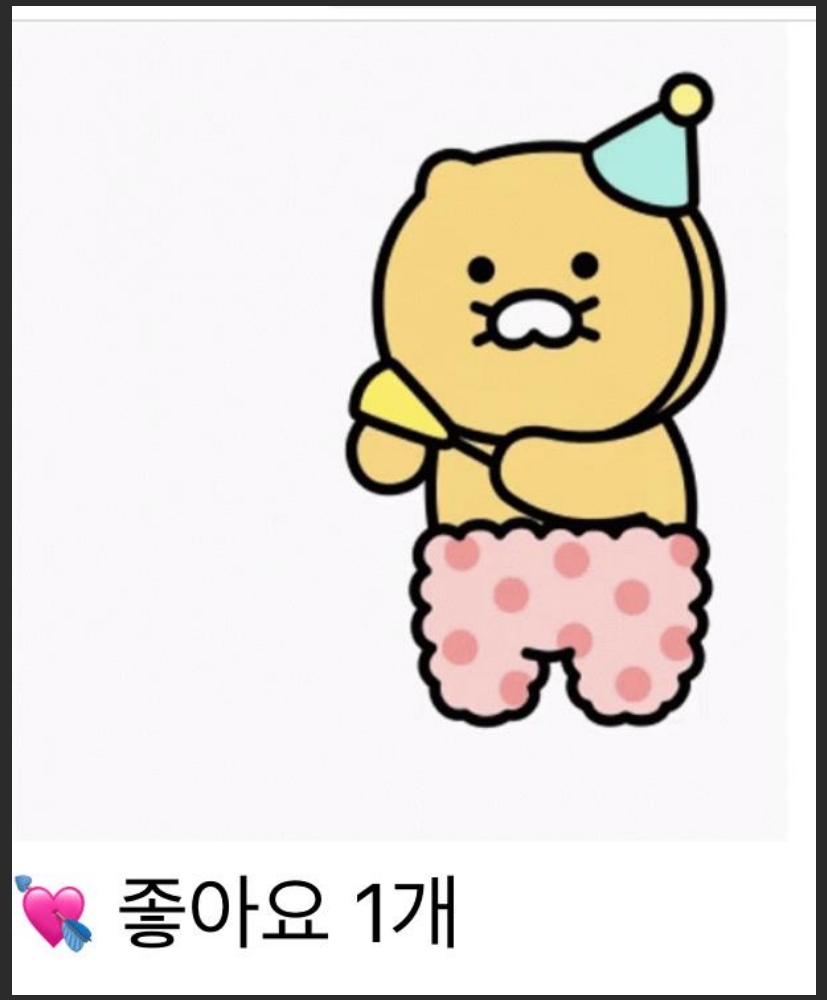
1. 데이터를 업데이트하면 전체 UI를 Virtual DOM에 리렌더링 한다.

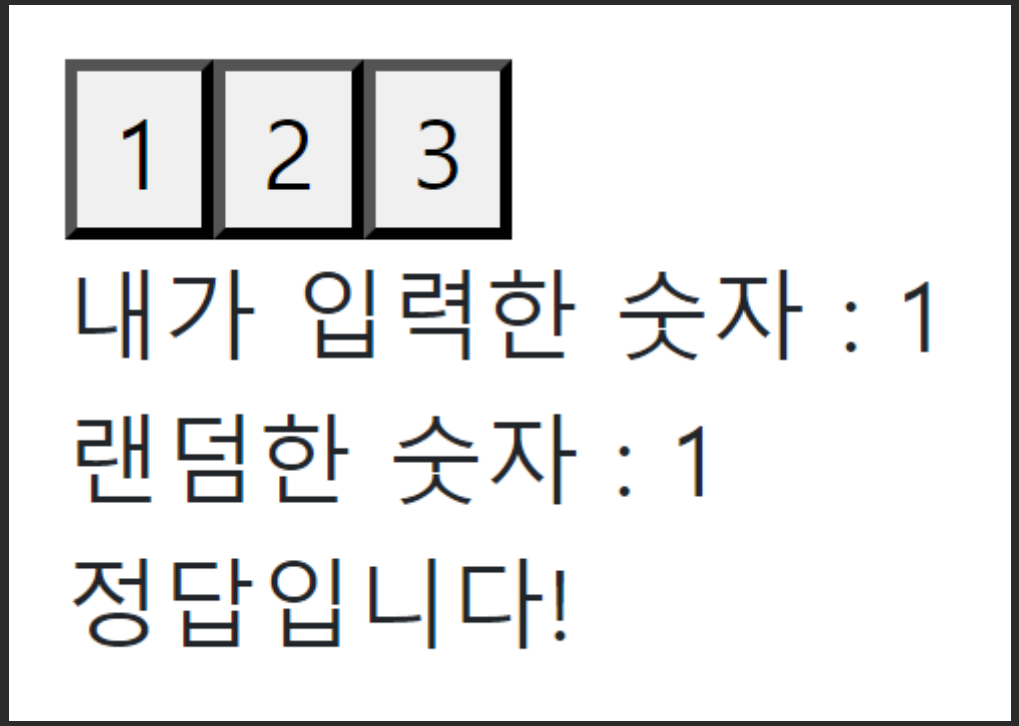
2. 이전 Virtual DOM에 있던 내용과 현재 내용을 비교한다.

3. 바뀐 부붐안 실제 DOM에 적용한다.

state 실습

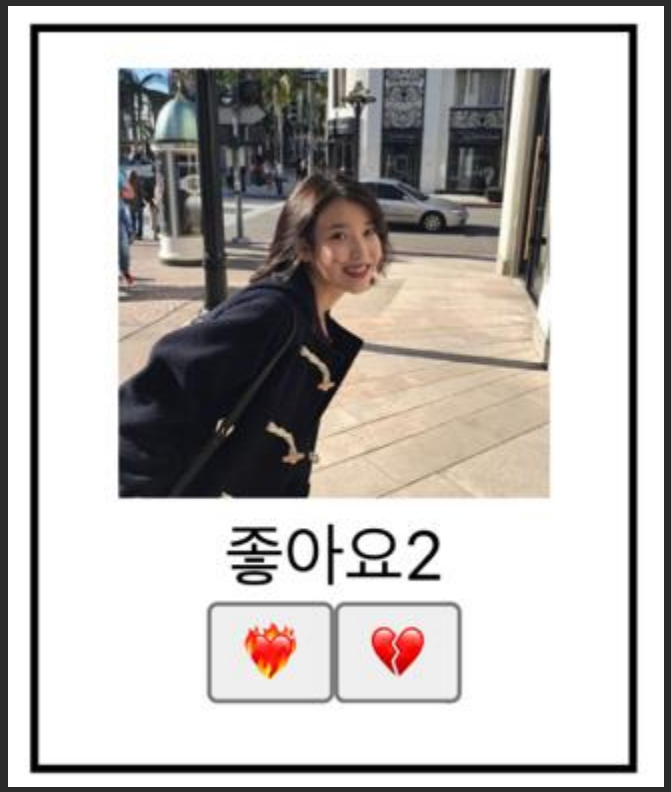




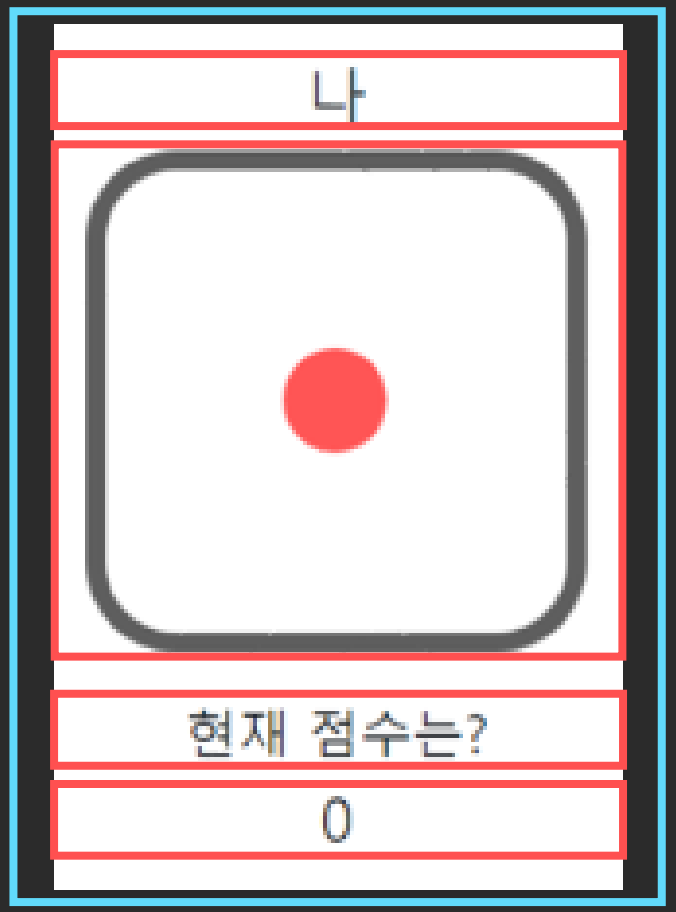

1
2
3
내가 입력한 숫자 : 1
랜덤한 숫자 : 1
정답입니다!

좋아요 +1 / 좋아요 -1 버튼을 만들어보자!

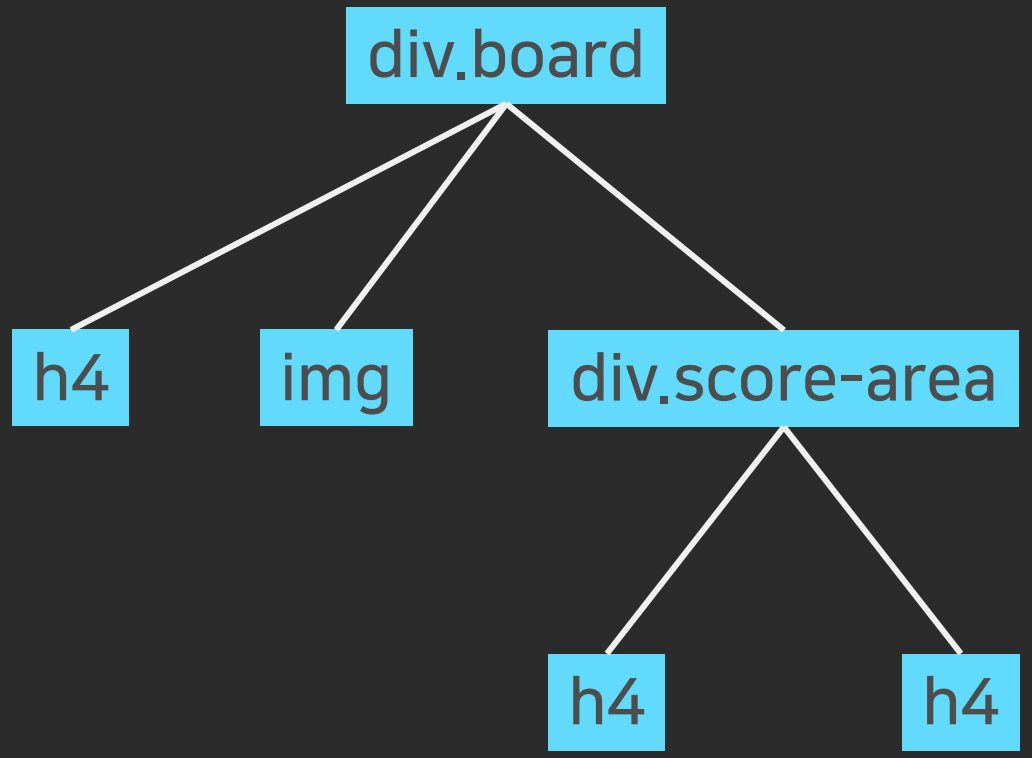
(단, 좋아요가 0 밑으로는 내려갈 수 없다)



Board 컴포넌트 생성



Board Component



Board Component 구조

[다음 수업 내용]

Map Filter

다음 이 시간에...