컴포넌트 (Component)

리액트로 만들어진 앱을 이루는 **최소 단위**

컴포넌트

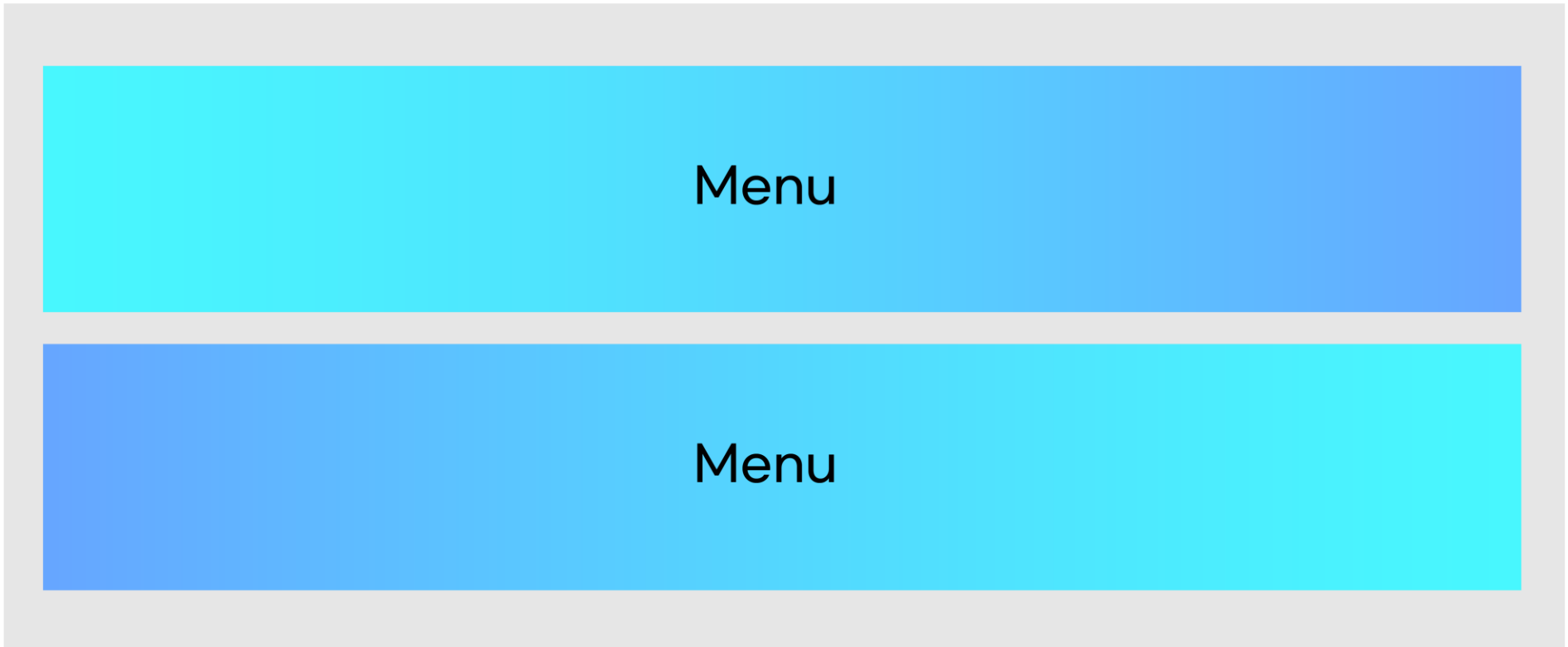
기존의 HTML 문서

컴포넌트

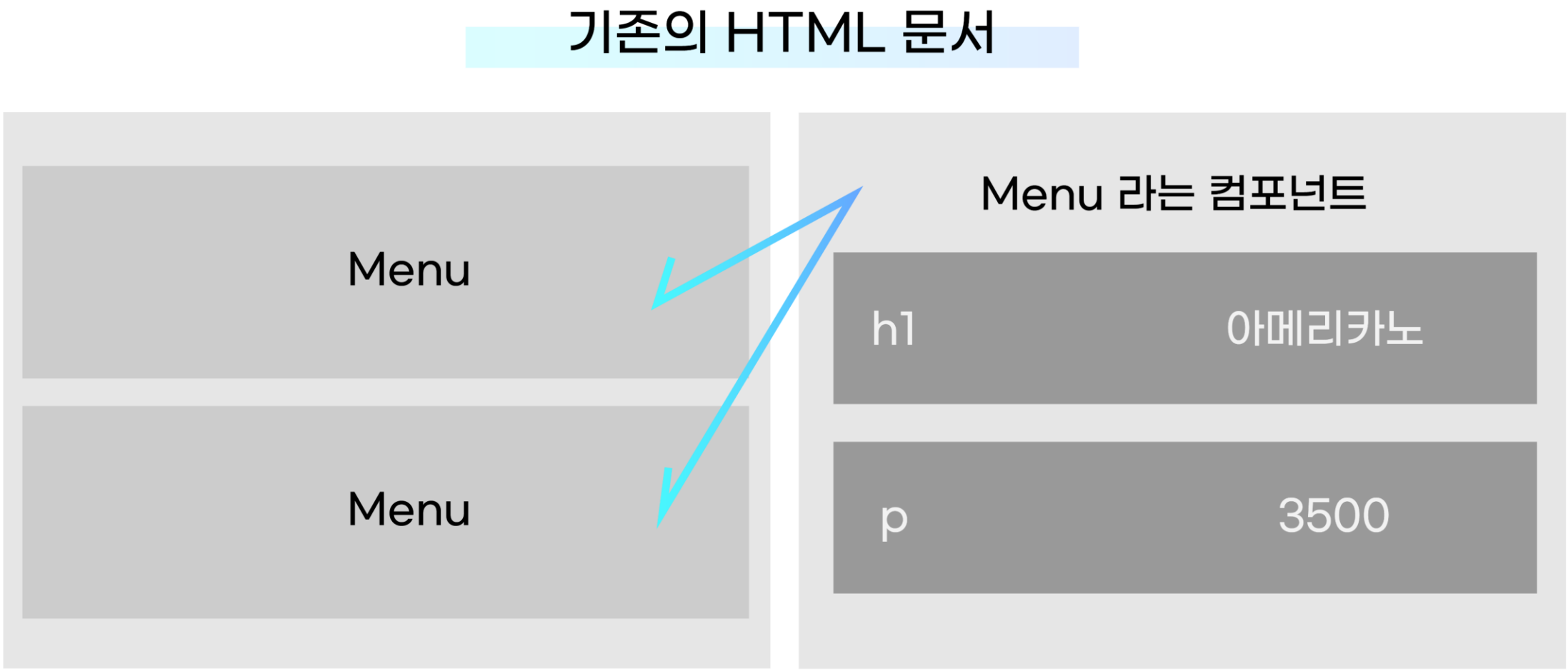
기존의 HTML 문서

컴포넌트

기존의 HTML 문서



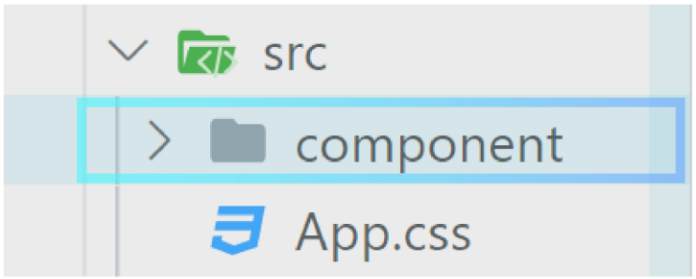
컴포넌트



컴포넌트

1. Extension 설치

2. component 폴더 생성 & 파일 생성



component → 파일이름.js 생성 (대문자)

이 때, 주의할 점! 컴포넌트의 이름은 반드시 대문자로 시작할 것!

3. 컴포넌트 파일 관리

rafce

```
import React from 'react'

const Menubox = () => {
    return (
        <div>Menubox</div>
    )
}

export default Menubox
```

컴포넌트 만드는 단축키 : rafce

함수형 컴포넌트 생성 완료!

컴포넌트 내보내기

4. 내보낸 컴포넌트 불러오기

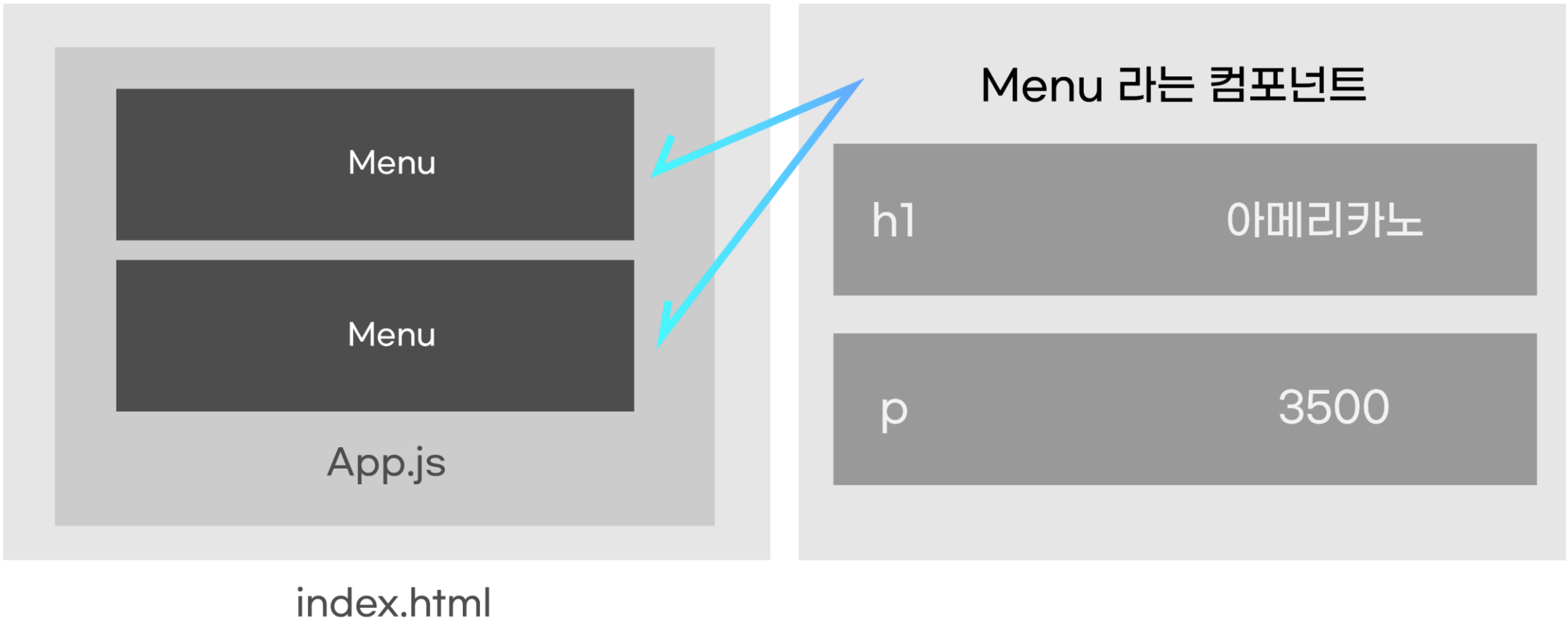
```jsx
import './App.css';
import MenuBox from './component/Menubox'

function App() {
  return (
    <>
    <MenuBox></MenuBox>
  </>


  );
}

export default App;
```

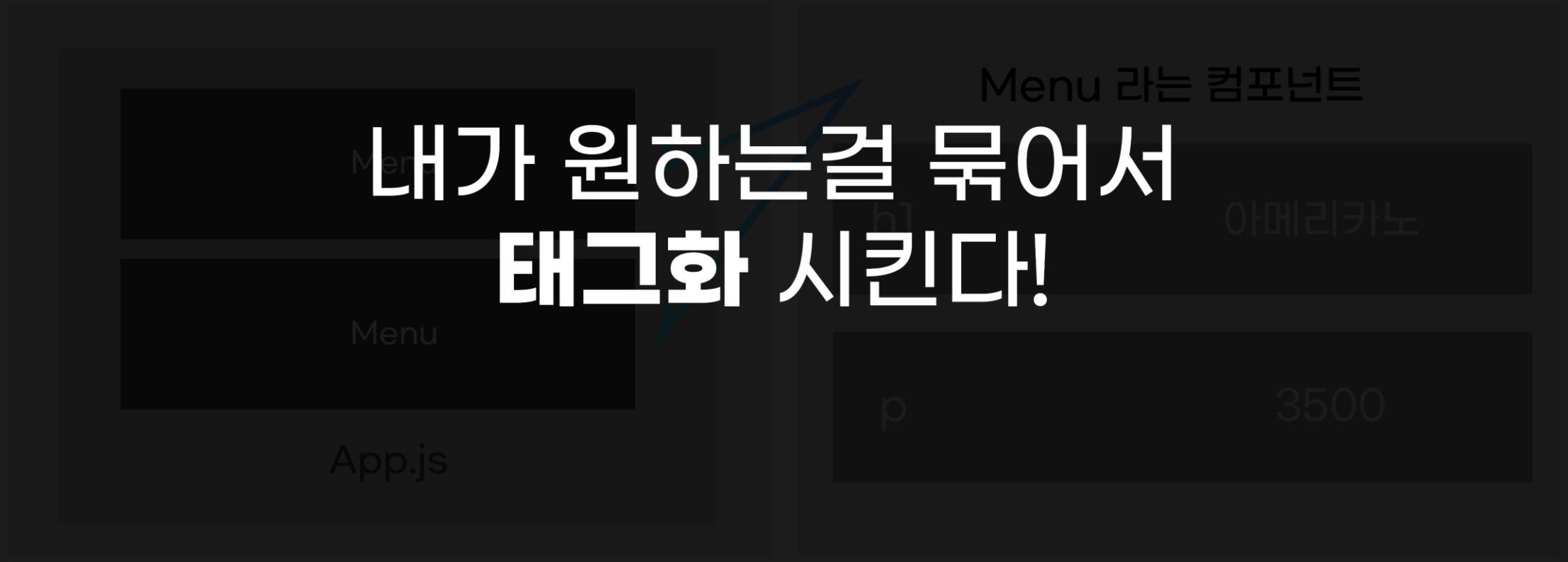
App.js

컴포넌트 불러오기

사용하기

컴포넌트





내가 원하는걸 묶어서
태그화 시킨다!

컴포넌트 사용 시 주의사항

컴포넌트의 이름은 반드시 **대문자로 시작**해야한다

리액트가 컴포넌트와 일반 HTML 태그를 구분할 수 있어야한다.
그렇기 때문에 리액트에서는 일반 HTML 태그는 소문자로,
컴포넌트는 반드시 대문자로 시작해야한다!

props

기존의 HTML 문서

props

기존의 HTML 문서

props

props (프로퍼티)

상위 컴포넌트가 하위 컴포넌트에 **값을 전달**할 때 사용한다

상위 컴포넌트에서 값을 입력하고, 하위 컴포넌트에서는 값을 읽어온다

props 상위 컴포넌트

props (프로퍼티) 사용방법

```
function App() {
  return (
    <>
    <MenuBox menu="카페라떼"></MenuBox>
    <MenuBox></MenuBox>
  </>
  );
}
```

```
function App() {
  let coffee = "카페라떼"
  return (
    <>
    <MenuBox menu={coffee}></MenuBox>
    <MenuBox></MenuBox>
  </>
  );
}
```

문자열을 전달할 때는 **큰따옴표를 (“”)**
문자열 외의 값을 전달할 때는 **중괄호 ({})**를 사용한다

props 하위 컴포넌트

props (프로퍼티) 사용방법

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

컴포넌트 & props 예제

교육운영부

선영표

전략기획팀

강예진

홍보팀

임보미

기획팀

최영화

컴포넌트와 props를 이용해서
팀원들을 소개하는 페이지를 만들어보자!

State

컴포넌트 내부에서 관리되는 변경이 가능한 데이터

state

State

컴포넌트 내부에서 관리되는 변경이 가능한 데이터

변수와의 차이점은?

State와 변수의 차이점

state는 일반 변수와 다르게
값이 변하면 화면이 렌더링된다

이 때, **setState() 라는 함수**를 이용한다

state

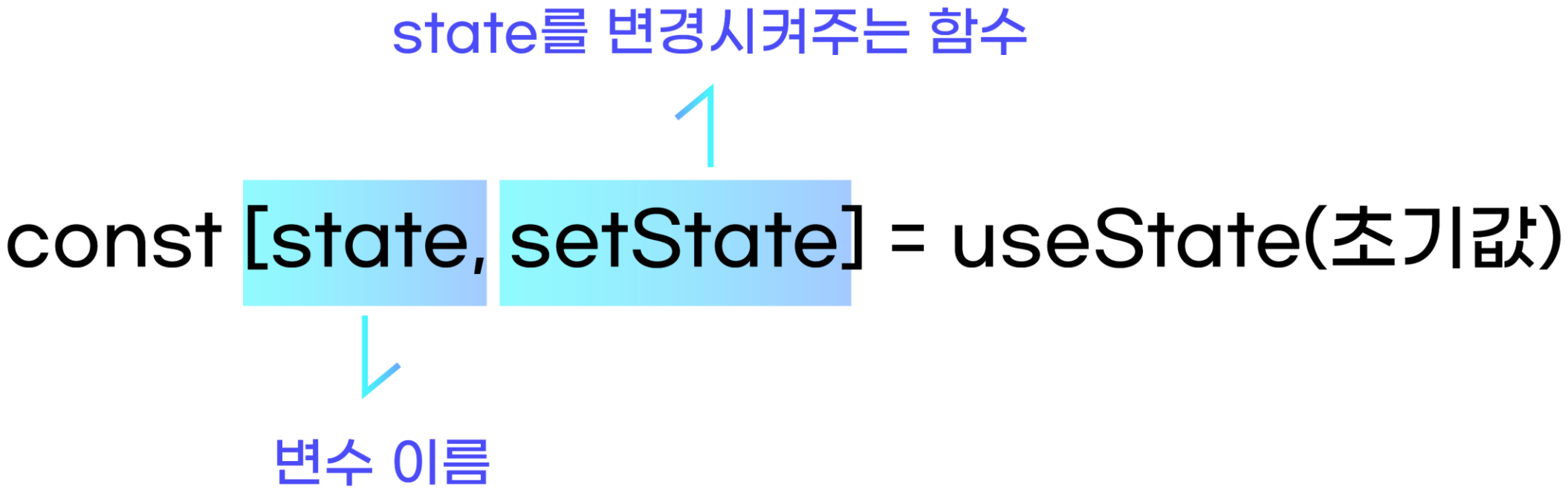
const [state, setState] = useState(초기값)

state

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

변수 이름

state



state


state를 변경시켜주는 함수
const [state, setState] = useState(초기값)
변수 이름
return 하면서 초기값 설정

state 예제

좋아요 +1 / 좋아요 -1 버튼을 만들어보자!

(단, 좋아요가 0 밑으로는 내려갈 수 없다.)

좋아요2

❤️‍🔥 💔

map 함수

map() 함수

배열의 내장 함수
반복되는 컴포넌트를 렌더링 하기 위해 사용됨

map() 함수

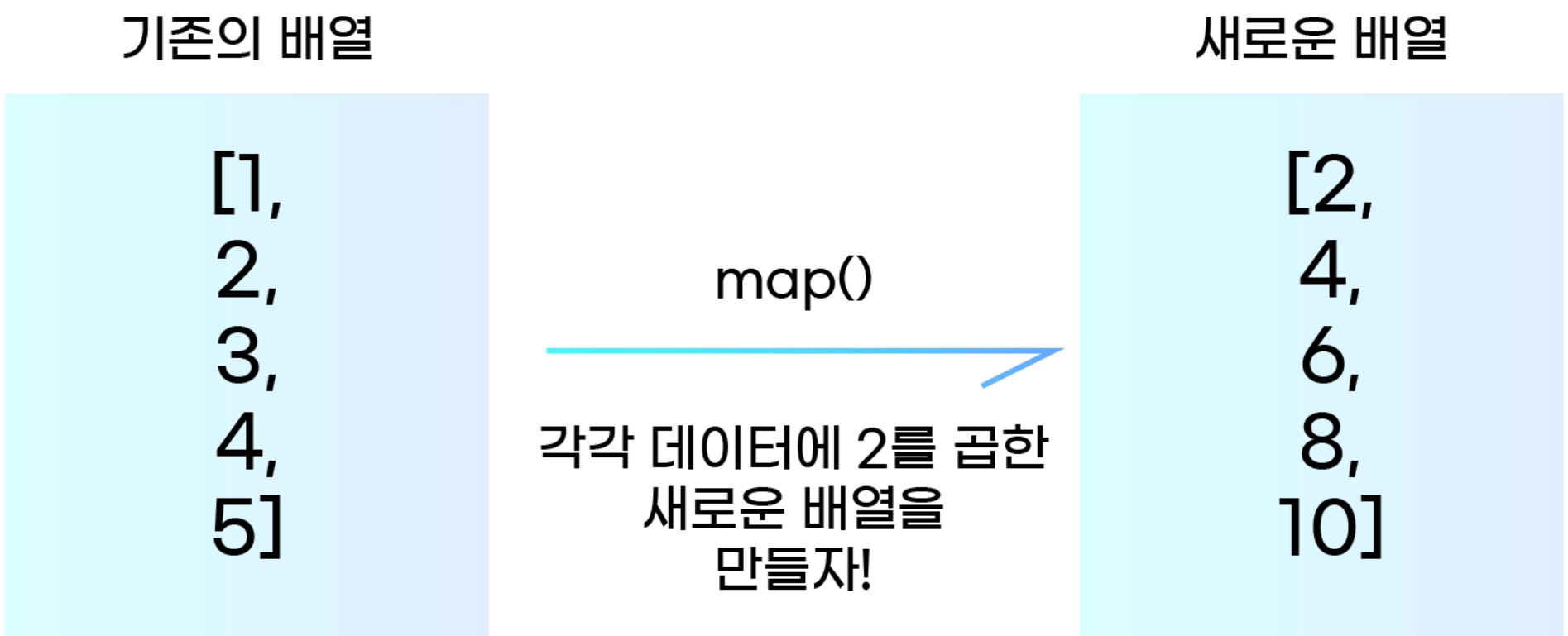
배열 안에서
내가 **원하는 규칙**에 따라
새로운 배열을 생성한다!

배열의 내장 함수
반복되는 컴포넌트를 렌더링 하기 위해 사용됨

map 함수

배열이름.map(콜백함수)

map 함수



map 함수

기존의 배열

[1,
2,
3,
4,
5]

map()

각각 데이터에 2를 곱한
새로운 배열을
만들자!

새로운 배열

[2,
4,
6,
8,
10]

```
let list = [1,2,3,4,5]
let resList = list.map(num=>num*2)
console.log(resList)
```

새로운 배열을
단순한 데이터 배열이 아닌
컴포넌트 배열로 만들면 어떨까?

map 함수

기존의 배열

[1,
2,
3,
4,
5]

map()

각각 데이터에 2를 곱한 새로운 컴포넌트 배열을 만들자!

새로운 배열

[
2,
4,
6,
8,
10]

map 함수

기존의 배열

[아이유,
수지,
나연,
사나,
원영]

map()

각각 데이터를
태그로 묶은
컴포넌트 배열을 만들자!

새로운 배열

[아이유 ,
수지 ,
나연 ,
사나 ,
원영]

map 함수 예제

오늘의 점심메뉴!

- 행복한 담벼락
- 그냥 집밥
- 시루시루
- 김밥집

return 문 내에 ** 태그를 사용하지 않고**
위와 같은 결과를 구현하시오.

map 함수 실습

아이유	수지	나연
		

위와 같은 결과를 구현하시오

조건 1 : **PhotoCard.js** 라는 컴포넌트를 생성할 것
조건 2 : member 라는 배열 안에 **name,url 이 포함되어있는 객체**들을 생성할 것
조건 3: 배열에 값을 추가하면 **바로 새로운 카드가 생성되도록** 할 것