

React

With Next.js 

1. JSX

연사 & 멘토 소개

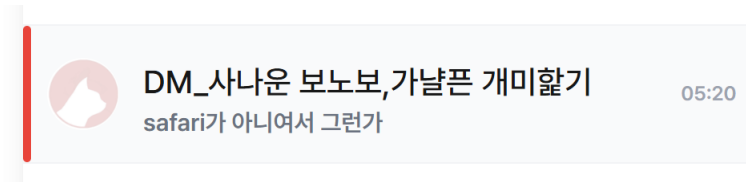
연사 : 박승범 (@killerwhale)

멘토 : 박승범 (@killerwhale)

멘토 : 박승범 (@killerwhale)

컴포넌트의 흐름을 제어해보자

조건에 따라 기능과 UI가 달라지는 컴포넌트



Selected = True



Selected = False

Post Example



게시판에 따라 달라지는 게시글 목록 UI

JSX

JSX(Javascript Syntax eXtension)

JavaScript 코드 내부에서 HTML과 비슷한 문법을 사용할 수 있도록 해주는 확장

Q) HTML 객체를 변수로 지정하려면 어떻게 해야할까? -> 일반 JS에서는 불가능

```
const element = <h1>Hello, world!</h1>;
```

```
<MyButton color="blue" shadowSize={2}>  
  Click Me  
</MyButton>
```

JSX



```
React.createElement(  
  MyButton,  
  {color: 'blue', shadowSize: 2},  
  'Click Me'  
)
```

React Object

JSX = React Object의 Syntax Sugar 이다.

컴포넌트 다시보기

```
"use client"; // 버튼은 클라이언트 컴포넌트여야 함

import { useState } from "react";

export default function ToggleButton() {
  const [active, setActive] = useState(false);

  return (
    <button
      onClick={() => setActive(!active)}
      className={`px-4 py-2 rounded-lg text-white transition
        ${active ? "bg-green-500" : "bg-gray-500"}
        hover:opacity-80`}
    >
      {active ? "활성화됨" : "비활성화됨"}
    </button>
  );
}
```

JSX !

JSX 문법

컴포넌트의 return 값은 반드시 하나의 루트노드만 존재하는 트리 형식이다.

```
return (  
  <h1>Title</h1>  
  <p>Description</p>  
);
```

O

```
return (  
  <div>  
    <h1>Title</h1>  
    <p>Description</p>  
  </div>  
);
```

X

Fragment : 익명의 태그

```
<>  
  <button  
    onClick={() => setIsPostReportVisible(true)}  
    className="p-2 flex flex-row gap-2 cursor-pointer border  
  >  
    <Image src="/Alert.svg" alt="" width={15} height={15} />  
    신고  
  </button>  
  <div  
    className="p-2 flex flex-row gap-2 cursor-pointer border  
    onClick={handleShare} // 공유 핸들러 연결  
  >  
    <Image src="/Share.svg" alt="" width={15} height={15} />  
    공유  
  </div>  
  { /* 답아두기 버튼 */ }  
  <div  
    className={`p-2 flex flex-row gap-2 cursor-pointer border  
    `}  
    onClick={handleScrap}  
  >  
    <BookmarkIcon isFilled={!!post.my_scrap} />  
    답아두기  
  </div>  
</>
```

모종의 이유로 <div> 사용이 어렵다면....

JSX 문법 : Expression in JSX

JSX 내부에 표현식을 넣어 흐름에 따라 다른 컴포넌트를 렌더링 할 수 있다.

표현식 (Js 혹은 Ts 코드)를 넣을 때는, 필요한 부분에 중괄호 {} 를 넣고 그 안에 작성한다.

Api로 받아온 comment 값에 따라 달라지는 텍스트와 프로필 이미지

```
<div className='flex flex-row gap-[4px] items-center font-medium'>
  <img src={comment.created_by.profile.picture} alt="example" width={20} />
  {comment.created_by.profile.nickname}
  <div className='flex text-[#B5B5B5] text-xs ml-3'>
    {formatDate(comment.created_at)}
  </div>
</div>
```

표현식에서 함수 사용 가능!

```
export const formatDate = (iso: string): string => {
  const date = new Date(iso);
  const yyyy = date.getFullYear();
  const mm = String(date.getMonth() + 1).padStart(2, '0');
  const dd = String(date.getDate()).padStart(2, '0');
  const hh = String(date.getHours()).padStart(2, '0');
  const min = String(date.getMinutes()).padStart(2, '0');
  return `${yyyy}.${mm}.${dd} ${hh}:${min}`;
};
```

JSX 문법 : 조건, 분기 처리하기

JSX의 문법은 '표현식' 만 허용한다. Functional 이라고 생각하면 된다.
따라서 조건문을 쓰려면 3항 연산자를 활용한다.

```
{pending} && (  
  <div className="m-2 flex items-center gap-2 self-start">  
    {pending.type === 'IMAGE' ? (  
      <Image src={pending.url} alt={pending.name || 'image'} width={pending.width} height={pending.height} />  
    ) : (  
      <div className="px-3 py-2 rounded-md border text-sm flex items-center gap-2">  
        <svg viewBox="0 0 24 24" className="w-4 h-4" fill="none" stroke="red" />  
        <span className="truncate max-w-[16rem]">{pending.name}</span>  
      </div>  
    )}  
    <button type="button" className="text-xs text-red-500 hover:text-red-600">삭제</button>  
  </div>  
)  
{(!pending)} && (  
  <TextareaAutosize  
    className="w-full px-2 py-1.5 focus:outline-none resize-none bg-white border border-gray-200">  
    placeholder={pending ? '' : '메시지를 입력하세요...'}  
    value={input}  
    onChange={(e: React.ChangeEvent<HTMLTextAreaElement>) => setInput(e.target.value)}  
    onKeyDown={handleKeyDown}  
    disabled={!pending || isUploading}  
    maxRows={8}  
    rows={1}  
  />  
)  
}
```

Functional이 궁금하다면?

CS220 (Programming Principle)
CS320 (Programming language)

이미지 타입과 텍스트 타입에서
다른 컴포넌트를 렌더링 하는 방법

JSX 문법 : 조건, 분기 처리하기

And 연산자 (&&)를 활용해서 특정 조건을 만족할때만 컴포넌트를 렌더링 할 수 있다.

```
{pending && (  
  <div className="m-2 flex items-center gap-2 self-start">  
    {pending.type === 'IMAGE' ? (  
      <Image src={pending.url} alt={pending.name || 'image'} width  
    ) : (  
      <div className="px-3 py-2 rounded-md border text-sm flex it  
        <svg viewBox="0 0 24 24" className="w-4 h-4" fill="none  
        <span className="truncate max-w-[16rem]">{pending.name  
      </div>  
    )}  
    <button type="button" className="text-xs text-red-500 hover:u  
  </div>  
)}  
{(!pending) && (  
  <TextareaAutosize  
    className="w-full px-2 py-1.5 focus:outline-none resize-none bg  
    placeholder={pending ? '' : '메시지를 입력하세요...'}  
    value={input}  
    onChange={(e: React.ChangeEvent<HTMLTextAreaElement>) => setInp  
    onKeyDown={handleKeyDown}  
    disabled={!pending || isUploading}  
    maxRows={8}  
    rows={1}  
  />  
)}  
)}
```

JS의 && 연산자

A && B

➔ A가 True면 A반환, False면 B 반환

➔ (조건) & (컴포넌트) 형식으로 쓰면 if문을 구현할 수 있다.

true && "Hello" → "Hello"

false && "Hello" → false

0 && "Hello" → 0

null && "Hello" → null

JSX 문법 : 반복문 처리하기

map함수를 통해 for문의 작동 방식과 비슷하게 컴포넌트를 렌더링 할 수 있다.

```
const numbers = [1, 2, 3, 4, 5];
const listItems = numbers.map((number) =>
  <li>{number}</li>
);
```

```
function NumberList(props) {
  const numbers = props.numbers;
  const listItems = numbers.map((number) =>
    <li>{number}</li>
  );
  return (
    <ul>{listItems}</ul>
  );
}

const numbers = [1, 2, 3, 4, 5];
```

리스트 컴포넌트 구현하기

```
import React from "react";

export default function App() {
  const items = ["사과", "바나나", "포도", "딸기"];

  return (
    <div className="flex gap-4 p-4">
      {items.map((item, index) => (
        <div
          key={index}
          className="p-4 bg-gray-200 rounded-lg shadow"
        >
          {item}
        </div>
      ))}
    </div>
  );
}
```

Flex 컨테이너 안에서 사용하는
경우도 많아요!

Props

전달받은 Props를 사용하는 방법

```
export default function MembersPanel({
  isOpen,
  onClose,
  members,
  roomType,
  myId,
  onLeaveRoom,
  onBlockUser,
  onBlockAndLeave,
  onDeleteRoom,
  onInviteClick, // 추가
}: MembersPanelProps) {
  const [menu, setMenu] = useState<{ visible: boolean; x: number; y: number }>({ visible: false, x: 0, y: 0 });

  const myRole = members.find(m => m.user.id === myId)?.role;

  interface MembersPanelProps {
    isOpen: boolean;
    onClose: () => void;
    members: Member[];
    roomType?: string;
    myId?: number;
    onLeaveRoom: () => void;
    onBlockUser: () => void;
    onBlockAndLeave: () => void;
    onDeleteRoom: () => void;
    onInviteClick: () => void; // 추가
  }
```

Ts에서는 Props를 전달할 때 위와 같이 Interface로 타입을 지정해 주어야 한다.

```
<div className="font-semibold">멤버 : {members.length}명</div>
```

Props로 받은 값은 컴포넌트 내부에서 사용 가능, but 변경은 불가능

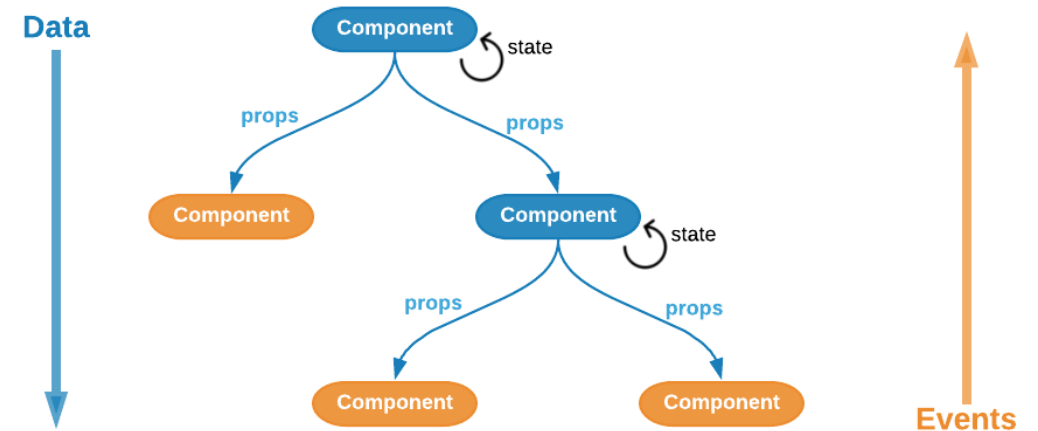
Props

자식 컴포넌트에 Props를 전달하여 자식 컴포넌트 내부에서 사용할 수 있도록 할 수 있다.

```
<MembersPanel
  isOpen={isOpen}
  onClose={() => setIsPanelOpen(false)}
  members={members}
  roomType={room?.room_type}
  myId={myId ?? undefined}
  onLeaveRoom={handleLeaveRoom}
  onBlockUser={handleBlockUser}
  onBlockAndLeave={handleBlockAndLeave}
  onDeleteRoom={handleDeleteRoom}
  onInviteClick={() => setInviteDialogOpen(true)}
/>
```

자식 컴포넌트에 props를 전달하는 방법

컴포넌트들은 트리 형식으로 구성되고,
자식들에게 Props를 순차적으로 전파한다.



Homework

HW1_JSX 완성하기 : <https://github.com/sparcs-kaist/seminar-NextJS>

제출 기한 :

Homework

Props에 전달된 Option에 따른 UI 구성하기

Options

☒ Show Rank ☒ Show Board ☒ Show Writer ☒ Show Hit ☒ Show Time Ago

1

아라 팀원 모집합니다.
자유게시판 · 박승범 · 조회 100 · 1시간 전

10

2

5

2

와이파이가 느려요
학교에게 전합니다 · 김경택 · 조회 200 · 2시간 전

20

1

10

3

수강 신청 안내
포탈 공지 · 학적팀 · 조회 300 · 3일 전

30

0

15

4

엄청나게 긴 안전에 대한 엄청나게 긴 회의를 통해 의결된 엄청...
총학 · 사나운 보노보 · 조회 150 · 1일 전

16

17

7

5

Ara 이용 약관 변경 안내
공지사항 · 신나는 포도 · 조회 500 · 7월 25일

45

1

20

Options

☐ Show Rank ☒ Show Board ☐ Show Writer ☐ Show Hit ☒ Show Time Ago

아라 팀원 모집합니다.
자유게시판 · 1시간 전

10

2

5

와이파이가 느려요
학교에게 전합니다 · 2시간 전

20

1

10

수강 신청 안내
포탈 공지 · 3일 전

30

0

15

엄청나게 긴 안전에 대한 엄청나게 긴 회의를 통해 의결된 엄청나게...
총학 · 1일 전

16

17

7

Ara 이용 약관 변경 안내
공지사항 · 7월 25일

45

1

20

The End

감사합니다