**React + react-slick + Bootstrap** 을 사용해서 만든  
📸 **이미지 슬라이더(캐러셀)** 예제입니다.

아래에서 **구조 → 주요 코드 → 작동 원리** 순으로 아주 쉽게 설명드릴게요 👇

**🧩 1️⃣ 전체 구조 요약**

이 코드는 다음과 같은 역할을 합니다:

| **구성** | **역할** |
| --- | --- |
| useState | 현재 몇 번째 이미지가 보이는지 상태 저장 |
| react-slick (<Slider />) | 슬라이드(캐러셀) 기능을 구현하는 라이브러리 |
| bootstrap | 레이아웃과 여백 등 기본 스타일 적용 |
| slick-carousel | react-slick 전용 CSS 스타일 적용 |
| images 배열 | 슬라이드에 표시할 이미지 목록 데이터 |

즉,  
➡ 이미지를 자동으로 넘겨주는 슬라이더를  
➡ react-slick이 담당하고,  
➡ useState로 현재 슬라이드 번호를 추적하는 구조입니다.

**🧠 2️⃣ 코드 분석 (중요 부분 중심으로)**

**✅ import 부분**

import { useState } from 'react' import Slider from 'react-slick' import 'bootstrap/dist/css/bootstrap.min.css' import 'slick-carousel/slick/slick.css' import 'slick-carousel/slick/slick-theme.css' import './App.css'

* react-slick → 슬라이더 컴포넌트 라이브러리
* slick-carousel → 슬라이더의 기본 디자인 CSS
* bootstrap → 그리드, 여백 등의 레이아웃 스타일
* App.css → 사용자가 추가 정의한 스타일

**✅ 이미지 배열**

const images = [ { id: 1, src: 'https://via.placeholder.com/800x400/FF6B6B/FFFFFF?text=Image+1', alt: 'Image 1' }, { id: 2, src: '...', alt: 'Image 2' }, ... ]

* 슬라이드에 표시할 이미지 데이터를 객체 형태로 저장합니다.
* src는 이미지 주소, alt는 이미지 이름 (캡션으로도 사용).

**✅ 슬라이더 설정**

const settings = { dots: true, *// 하단 점 네비게이션 표시* infinite: true, *// 무한 반복* speed: 500, *// 슬라이드 전환 속도 (ms)* slidesToShow: 3, *// 한 화면에 3개씩 표시* slidesToScroll: 1, *// 한 번 넘길 때 1장 이동* autoplay: true, *// 자동 넘김 켜기* autoplaySpeed: 3000, *// 자동 넘김 간격 (3초)* beforeChange: (current, next) => setCurrentSlide(next), *// 상태 업데이트* arrows: true, *// 좌우 화살표 표시* fade: false, *// 페이드 효과 대신 슬라이드 효과* responsive: [ { breakpoint: 992, *// 화면 992px 이하일 때* settings: { slidesToShow: 2 } *// 2개씩 표시* }, { breakpoint: 768, *// 768px 이하일 때* settings: { slidesToShow: 1 } *// 1개씩 표시* } ] }

즉 👉  
📱 화면 크기에 따라 자동으로 슬라이드 개수를 줄이고  
💫 자동 재생 기능도 있는 반응형 슬라이더입니다.

**✅ JSX 렌더링 부분**

<Slider {...settings}> {images.map((image) => ( <div key={image.id} className="slide-item"> <div className="px-2"> <img src={image.src} alt={image.alt} className="img-fluid w-100" style={{ height: '300px', objectFit: 'cover' }} /> <div className="slide-caption mt-2"> <h6>{image.alt}</h6> </div> </div> </div> ))} </Slider>

* Slider 컴포넌트 안에 images.map()을 돌려서 각 이미지 생성
* Bootstrap 클래스(img-fluid, w-100)로 반응형 처리
* objectFit: 'cover'로 이미지를 비율 유지한 채 꽉 차게 표시

**✅ 현재 슬라이드 표시 부분**

<p className="text-muted"> 현재 이미지: {currentSlide + 1} / {images.length} </p>

* currentSlide 상태를 통해 현재 몇 번째 이미지인지 표시
* 예: 현재 이미지: 2 / 6

**🎯 3️⃣ 동작 순서 정리**

1️⃣ 페이지가 로드되면  
images 배열의 첫 번째 이미지부터 표시됩니다.

2️⃣ react-slick이 설정(settings)에 따라 자동으로 슬라이드 전환

3️⃣ beforeChange 이벤트에서  
다음 슬라이드 인덱스를 setCurrentSlide(next)로 업데이트

4️⃣ 하단에 현재 이미지: x / y 가 자동으로 바뀝니다.

**🪄 4️⃣ 추가 팁**

* 슬라이더 크기를 바꾸려면  
  img의 height 또는 objectFit을 조정하면 됩니다.
* autoplay: false로 바꾸면 수동 슬라이드로 전환됩니다.
* fade: true로 바꾸면 “부드럽게 사라졌다 나타나는” 페이드 효과 적용됩니다.

**✅ 정리 요약**

| **항목** | **설명** |
| --- | --- |
| useState | 현재 슬라이드 상태 저장 |
| react-slick | 슬라이드 기능 구현 핵심 |
| slick-carousel | react-slick 전용 CSS |
| autoplay, dots, responsive | 자동재생, 점, 반응형 설정 |
| beforeChange | 다음 슬라이드로 이동할 때 상태 갱신 |
| Bootstrap | 레이아웃 및 정렬 스타일링 |

**React + Swiper.js**를 이용해 만든 **반응형 이미지 슬라이더(캐러셀)** 컴포넌트입니다.  
앞서 보던 react-slick 슬라이더와 비슷하지만, Swiper.js는 모듈 방식으로 좀 더 유연하고 최신 환경에 최적화되어 있습니다.

아래에서 구조와 핵심 포인트를 단계별로 설명드릴게요 👇

**🧩 1️⃣ 전체 구조 요약**

| **구성** | **역할** |
| --- | --- |
| Swiper, SwiperSlide | 슬라이더와 각 슬라이드 요소를 생성 |
| Navigation, Pagination, Autoplay | 화살표, 점, 자동재생 기능 모듈 |
| images 배열 | 슬라이드에 표시할 이미지 데이터 저장 |
| breakpoints | 반응형(모바일, 태블릿, 데스크탑) 설정 |
| CSS | Bootstrap + Swiper 전용 + 커스텀 스타일 |

즉, **자동 재생 + 반응형 + 네비게이션/페이지네이션** 기능을 가진 이미지 캐러셀을 구현한 구조입니다.

**🧠 2️⃣ 코드 분석**

**① import**

import { Swiper, SwiperSlide } from "swiper/react"; import { Navigation, Pagination, Autoplay } from "swiper/modules"; import "swiper/css"; import "swiper/css/navigation"; import "swiper/css/pagination"; import "./SwiperSlide.css";

* Swiper 라이브러리와 모듈을 import
* CSS도 각각 import → Swiper 기본 스타일과 화살표, 점 스타일 적용
* ./SwiperSlide.css → 커스텀 스타일 적용 가능

**② 이미지 데이터 배열**

const images = [ { id: 1, src: "./images/c1.jpg", alt: "Swiper Image 1" }, ... ]

* 슬라이드에 표시할 이미지 목록
* id → key로 사용
* src → 이미지 경로
* alt → 이미지 설명, 캡션에도 사용

**③ Swiper 설정**

<Swiper modules={[Navigation, Pagination, Autoplay]} spaceBetween={20} slidesPerView={3} slidesPerGroup={1} navigation pagination={{ clickable: true }} autoplay={{ delay: 3000, disableOnInteraction: false }} breakpoints={{ 320: { slidesPerView: 1, spaceBetween: 10 }, 768: { slidesPerView: 2, spaceBetween: 15 }, 992: { slidesPerView: 3, spaceBetween: 20 }, }} >

* modules → 사용할 기능 모듈 지정
  + Navigation → 좌우 화살표
  + Pagination → 하단 점
  + Autoplay → 자동 슬라이드
* spaceBetween → 슬라이드 간 간격(px)
* slidesPerView → 한 화면에 몇 장 표시할지
* slidesPerGroup → 한 번 이동 시 슬라이드 개수
* navigation → 화살표 표시
* pagination={{ clickable: true }} → 점 클릭 가능
* autoplay → 자동 재생 속도와 인터랙션 옵션
* breakpoints → 화면 크기별 반응형 설정

**④ 슬라이드 JSX**

{images.map((image) => ( <SwiperSlide key={image.id}> <div className="swiper-slide-item"> <img src={image.src} alt={image.alt} className="img-fluid w-100" style={{ height: "300px", objectFit: "cover" }} /> <div className="swiper-caption mt-2"> <h6>{image.alt}</h6> </div> </div> </SwiperSlide> ))}

* SwiperSlide 하나가 **슬라이드 한 장**
* 내부에 이미지 + 캡션
* objectFit: "cover" → 이미지 비율 유지하면서 영역 채우기
* Bootstrap 클래스(img-fluid w-100, mt-2)로 반응형 + 여백 처리

**⑤ 반응형**

breakpoints={{ 320: { slidesPerView: 1, spaceBetween: 10 }, 768: { slidesPerView: 2, spaceBetween: 15 }, 992: { slidesPerView: 3, spaceBetween: 20 }, }}

* 모바일(320px) → 1장씩
* 태블릿(768px) → 2장
* 데스크탑(992px 이상) → 3장
* spaceBetween도 화면별로 다르게 지정 가능

**🧩 3️⃣ 특징 요약**

1. **반응형 최적화**: breakpoints 설정으로 모바일/태블릿/데스크탑 자동 조정
2. **자동 재생 + 반복**: Autoplay 모듈 사용, disableOnInteraction: false로 유저가 클릭해도 자동 재생 계속
3. **네비게이션/페이지네이션**: 좌우 화살표 + 클릭 가능한 점 표시
4. **이미지 스타일링 유연**: objectFit, height, Bootstrap 사용으로 이미지 비율 안정적 표시
5. **확장 가능**: CSS를 바꿔서 화살표, 점, 캡션 디자인 자유롭게 커스터마이징 가능