

# React 핵심 Hook과 Next.js 활용법

## 1. `useState` - React의 상태 관리 Hook

### 📌 개념

- 컴포넌트의 상태값을 정의하고 업데이트할 수 있도록 해주는 Hook
- 상태값이 변경되면 컴포넌트는 **자동으로 리렌더링**



React에서 컴포넌트가 리렌더링되는 **3가지** 경우

1. props가 변경될 때
2. 부모 컴포넌트가 리렌더링될 때
3. **state**가 변경될 때

### 활용 예시

```
import { useState } from 'react';

const Counter = () => {
  const [count, setCount] = useState(0); // 초기값 0

  return (
    <button onClick={() => setCount(count + 1)}>
      Count: {count}
    </button>
  );
};
```

## 2. **useEffect** - React의 사이드 이펙트 처리 Hook

### 개념

- 컴포넌트가 마운트되거나 업데이트될 때,

부수 효과(Side Effect)를 처리

#### ▼ 부수 효과(Side Effect)란?

: 화면에 영향을 주는 외부 작업

ex) 데이터 fetch, DOM 조작, 타이머 설정 등

### 기본 사용법

```
import { useEffect, useState } from 'react';

const Timer = () => {
  const [seconds, setSeconds] = useState(0);

  useEffect(() => {
    const interval = setInterval(() => setSeconds(s => s + 1), 1000);

    return () => clearInterval(interval);
    // 메모리 누수 방지를 위한 타이머 정리
  }, []);
  // [] 의존성 배열 : 빈 배열이면 컴포넌트 마운트 시 한 번만 실행
};
```

- 의존성 배열 안에 값이 있다면, 해당 값이 **변할 때마다 다시 실행**

### 활용 예시

- 클라이언트 측 데이터 fetching 시 사용

ex) 페이지 진입 후 API 호출

```
useEffect(() => {
  fetch('/api/data') // 서버에 데이터 요청 보냄
    .then(res => res.json()) // 응답을 JSON으로 바꿈
    .then(setData); // 가져온 데이터를 상태에 저장
}, []);
```

**SSR** (getServerSideProps) , **SSG** (getStaticProps) 와는 용도와 실행 시점이 다름

#### 1 `useEffect`

: 페이지가 열리자마자, 브라우저에서 API 요청해서 데이터 가져오기

#### 2 `getServerSideProps`, `getStaticProps`

: SEO(검색 엔진 최적화) 필요하거나, 페이지를 서버에서 미리 만들고 싶을 때

### 3. `useRouter` - Next.js 라우팅 제어 Hook

#### 개념

- Next.js 전용 Hook
- 현재 라우터 상태를 읽거나, 페이지 이동 등의 라우팅 기능 제공

#### 기본 사용법

```
import { useRouter } from 'next/router';

const NavigateButton = () => {
  const router = useRouter();

  return (
    <button onClick={() => router.push('/about')}>
      Go to About
    </button>
  );
};
```

#### 주요 기능

- `router.push(path)` : 페이지 이동
- `router.query` : URL 쿼리스트링 조회
- `router.pathname` : 현재 경로 확인
- `router.replace` : 뒤로 가기를 눌러도 방금 페이지로 못 돌아감
- `router.back()` : 이전 페이지로 돌아가기

## 버튼 클릭 시 페이지 이동 ( `router.push()` 사용)

```
// pages/index.js

import { useRouter } from 'next/router';

export default function HomePage() {
  const router = useRouter();

  const goToAbout = () => {
    router.push('/about'); // '/about' 페이지로 이동
  };

  return (
    <div>
      <h1>홈페이지</h1>
      <button onClick={goToAbout}>About 페이지로 이동</button>
    </div>
  );
}
```