# 서버 컴포넌트와 클라이언트 컴포넌트

| | |
|---|---|
| 구현 어려움 정도 | 개념 |
| 구현 완료 | ☑ |
| 기간 | @December 16, 2024 |
| 정리 완료 | ☑ |

NextJS에는 2개의 컴포넌트가 있다.

> 1. Server Component
> 2. Client Component

React 18이후 도입된 개념으로 React와 Next.js의 렌더링 최적화에 중요한 역할을 합니다. 각 컴포넌트는 렌더링 방식과 실행되는 환경이 다릅니다.

## Server Component

- 서버 측에서 렌더링되며 HTML을 생성해 클라이언트로 전달합니다.
- 이 과정에서 Javascript 번들이 클라이언트에 전동되지 않기 때문에 렌더링 성능이 향상되고, 네트워크 트레픽이 줄어듭니다.

### 특징

- 서버에서만 실행
  - 블라우저 환경에서는 실행되지 않음
- Javascript 번들에 포함되지 않음
  - 클라이언트로 전달되는 HTML은 Javascript 없이 바로 렌더링됩니다.
- 데이터베이스 접근 가능
  - 서버에서 실행되기 때문에 데이터베이스, 파일 시스템 등 보안이 필요한 작업을 안전하게 수행할 수 있습니다.
- SEO(검색 엔진 최적화)에 좋음
  - 서버에서 HTML이 완성된 상태로 전송되므로 검색엔진 최적화에 유리합니다.

```
export default async function ServerComponent() {
  const res = await fetch("https://api.example.com/data");
  const data = await res.json();

  return (
    <div>
      <h1>서버 컴포넌트</h1>
      <pre>{JSON.stringify(data, null, 2)}</pre>
    </div>
  );
}
```

서버 컴포넌트는 Next.js에서 기본값입니다. 별다른 선언이 없어도 서버 컴포넌트로 동작합니다.

## Client Component

- 클라이언트 컴포넌트는 브라우저에서 실행되며 사용자와의 상호작용(이벤트, 상태관리 등)을 처리하는데 사용됩니다.

### 특징

- 브라우저에서 실행

    - 브라우저 환경에서 실행되므로 상호작용(클릭, 입력 등)이 가능합니다.
- Javascript 번들에 포함됨
    - 컴포넌트 코드가 클라이언트에 전달되고 실행됩니다.
- 상태와 이벤트 처리
    - useState, useEffect 등 React 훅을 사용해서 상태를 관리하거나 이벤트를 처리할 수 있습니다.
- 서버 자원 접근 불가
    - 보안상의 이유로 브라우저에서는 서버의 파일 시스템이나 데이터베이스에 접근할 수 없습니다.

```
"use client";

import { useState } from "react";

export default function ClientComponent() {
  const [count, setCount] = useState(0);

  return (
    <div>
      <h1>클라이언트 컴포넌트</h1>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}
```

클라이언트 컴포넌트 최상단에 `“use client”` 라는 지시어를 추가해야 합니다.