

# Nextjs 14 컴포넌트에 대해서...

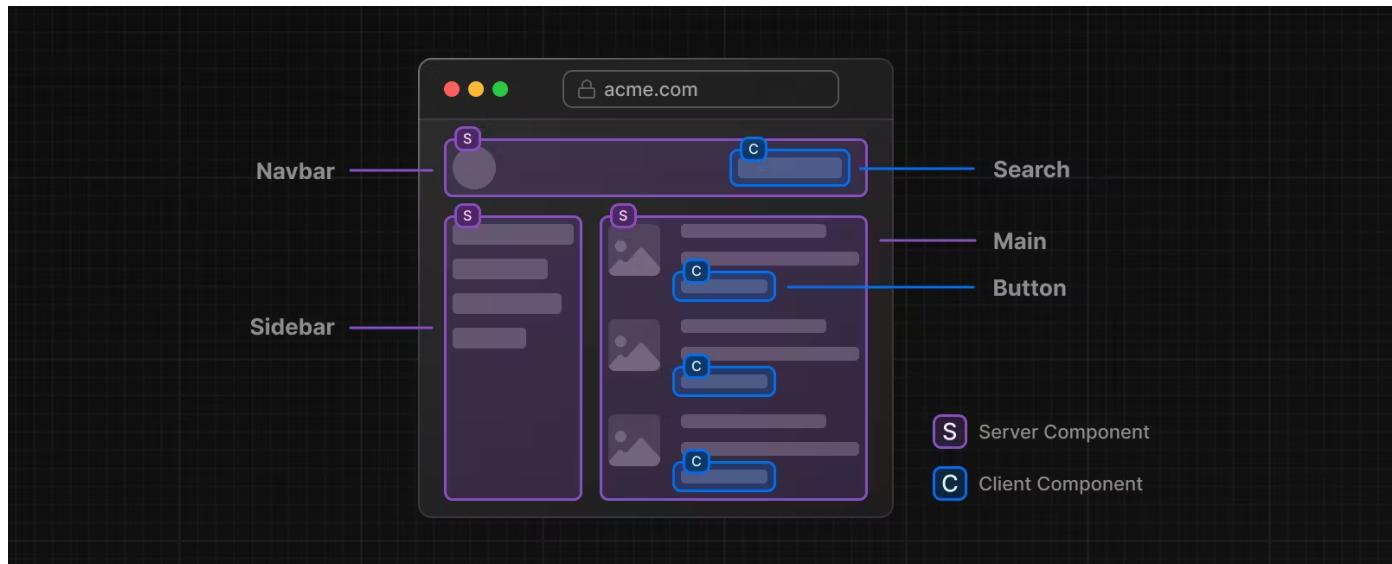
## table of contents

- Server components
- Client components
- 그럼 언제 써야 하나?

## Server components

Next.js app router는 Server component를 적용할 수 있게 구현이 되어있다

Server components를 통해 전체 어플리케이션을 클라이언트에서 렌더링 하는 것이 아닌 의도에 따라 어디에서 컴포넌트를 렌더링할지 정할 수 있다.



예를 들어 위 그림에서 Navbar, Sidebar, Main 컴포넌트는 interactive 한 부분이 없는 정적인 컴포넌트이기 때문에 서버에서 Server component로 렌더링 할 수 있다. 나머지 interactive한 UI는 Client component로 렌더링하게 된다. 이를 Next.js에선 Server-first 접근법이라 한다.

App router에서는 기본적으로 모든 컴포넌트가 Server component 디폴트이다. Client component로 사용하고 싶다면 `use client` 선언문을 통해 변경할 수 있다.

## Client components

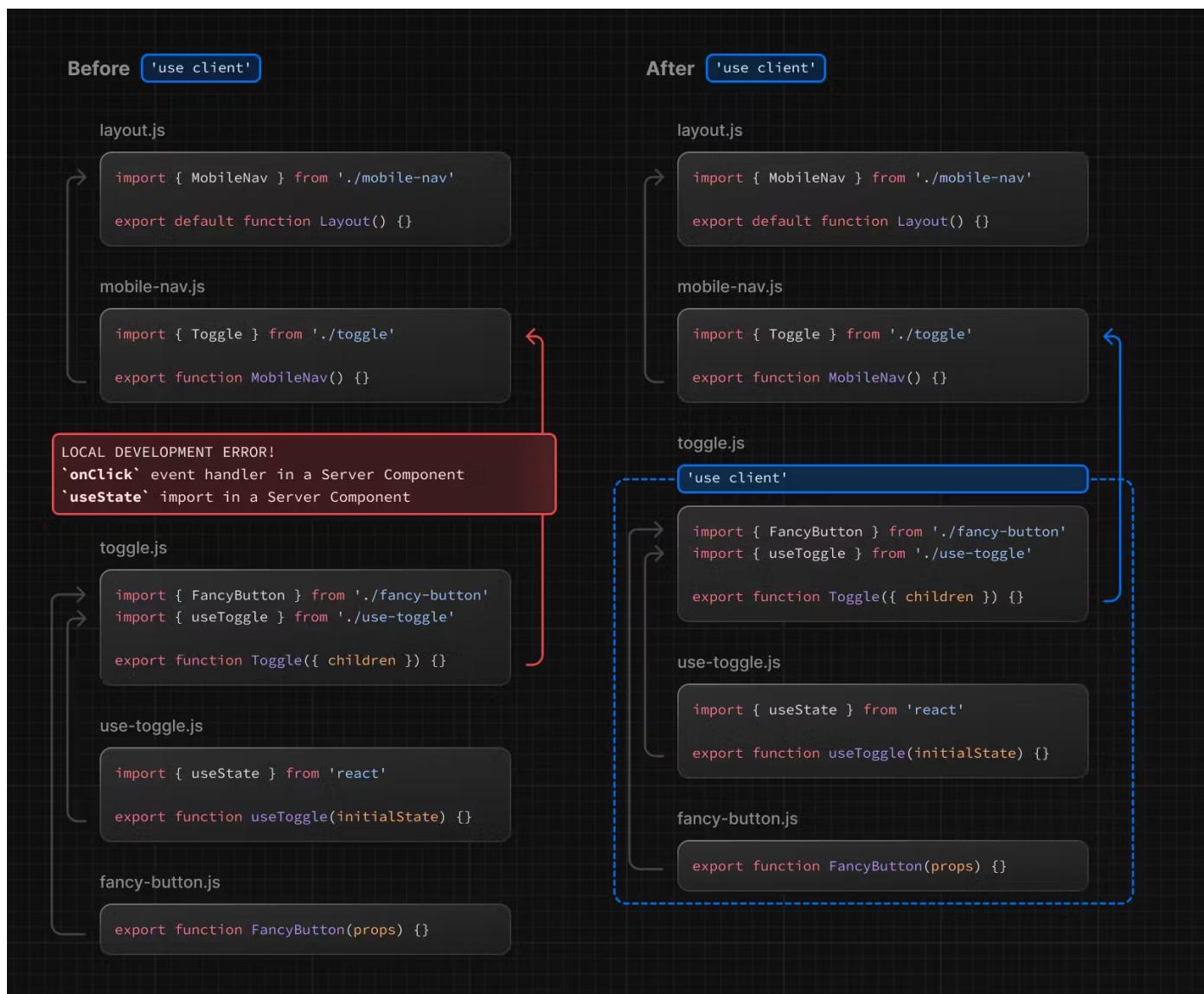
Client component는 어플리케이션에 client 측 interactivity를 추가할 수 있게한다. Next.js에선 이 컴포넌트들은 먼저 서버에서 pre-render 된 후 클라이언트에서 hydrate(JS가 실행되면서 interactive하게 ...) 된다.

```
'use client'; // 요런식으로 선언해주면 여기서부터 client 컴포넌트임

import { useState } from 'react';

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

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```



## 그럼 언제 써야 하나?

Next.js에서는 기본적으로 먼저 Server component를 사용하고 필요할때 Client component를 사용하라고 가이드 하고 있다.

- Server component
  - 데이터 fetching
  - 백엔드 자원에 접근
  - 민감한 정보를 서버에 유지
  - large dependencies 를 서버에서 유지 / 클라이언트 js 번들 사이즈 감소
- Client component
  - interactivity, event listener (onClick등) 가 필요할 때
  - state 및 라이프사이클이 필요할 때
  - browser-only API 사용
  - custom hook depend on state

[Next.js 14](#)