

# Next.js 심화 6

## 1 React 18과 함께 살펴보기

## Error Boundary 관련

`errorInfo.componentStack` 오류 수정

## React 18

(<https://reactjs.org/blog/2022/03/29/react-v18.html>)

Automatic Batching / Transitions / New Suspense

New Client & Server Rendering APIs / Strict Mode

useId / useTransition / useDeferredValue

useSyncExternalStore / useInsertionEffect...

## 동시성

기존에는 상태가 변경되면 update와 렌더링까지 react가 알아서 처리

React 18부터 update 와 렌더링 과정에서 중지가 가능

## React 18 with Next.js

(<https://nextjs.org/docs/advanced-features/react-18/overview>)

## Streaming SSR with Suspense

(<https://nextjs.org/docs/advanced-features/react-18/streaming>)

Stream 을 활용해서 SSR의 결과물을 업데이트 해준다.

(<https://nextjs.org/docs/api-reference/next/streaming>)

client-side에서 server component를 업데이트 할 수 있음

## 주의 사항

SSR을 활용하면,  
third party script 로드 시점을 보장할 수 없어진다.

next/head => next/script

\_document.js 로 옮기고 beforeInteractive strategy 적용

## Server Components

(<https://nextjs.org/docs/advanced-features/react-18/server-components>)

.server.js .client.js

(<https://github.com/vercel/next-react-server-components>)

yarn create next-app server-component –example

“<https://github.com/vercel/next-react-server-components>”



Edge runtime

(<https://nextjs.org/docs/advanced-features/react-18/switchable-runtime>)

runtime: node.js / edge

ssr streaming을 위해서는 web streams를 서포트해야함...

## 정리

### React 18과 함께 살펴보기

