

# 알아보자 CSR, SSR

CSR과 SSR 이해 및 비교



이강현

GDSC Soongsil  
lkh981107@gmail.com

```
filterByOrg = filterByOrg ? study.lead_organization === filterByOrg : true  
filterByStatus = filterByStatus ? study.status === filterByStatus : true  
matchStatus) {
```

```
function filterStudies({ studies, filterByOrg = false, filterByStatus = false }) {  
  return studies.filter(study => {  
    return filterByOrg === study.lead_organization && filterByStatus === study.status
```

# 주제 소개

**SSR**      📌      **Server Side Rendering**

**CSR**      📌      **Client Side Rendering**

## ★ 렌더링 준비를 어디서 하느냐?

페이지 내부 구성을 위한 로직이나 DB에서 데이터를 가져오는 data fetching 등의 과정



# 주제 소개



**SSR**



**완성된 HTML 파일** ★

**CSR**



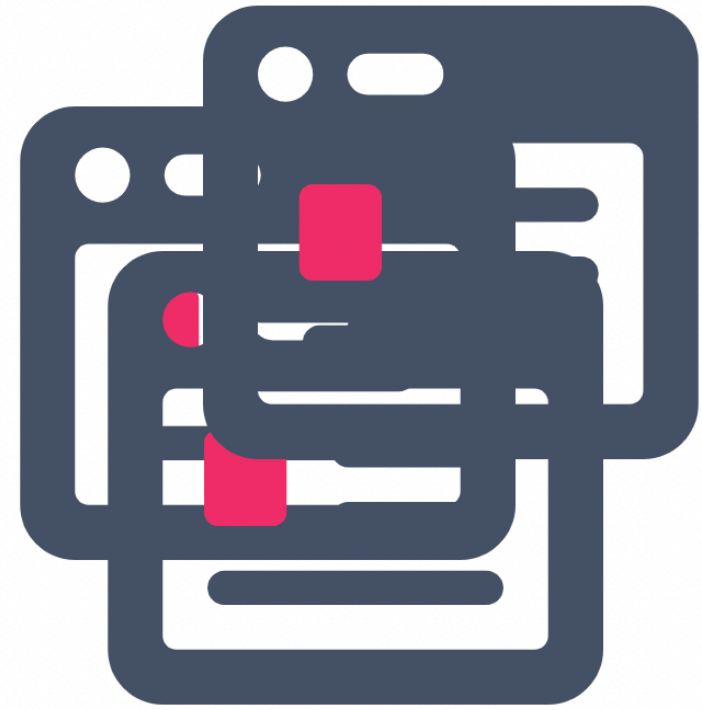
**페이지를 동적으로 구성  
할 수 있는 JS 파일** ★



# Web Application 역사



# SSR



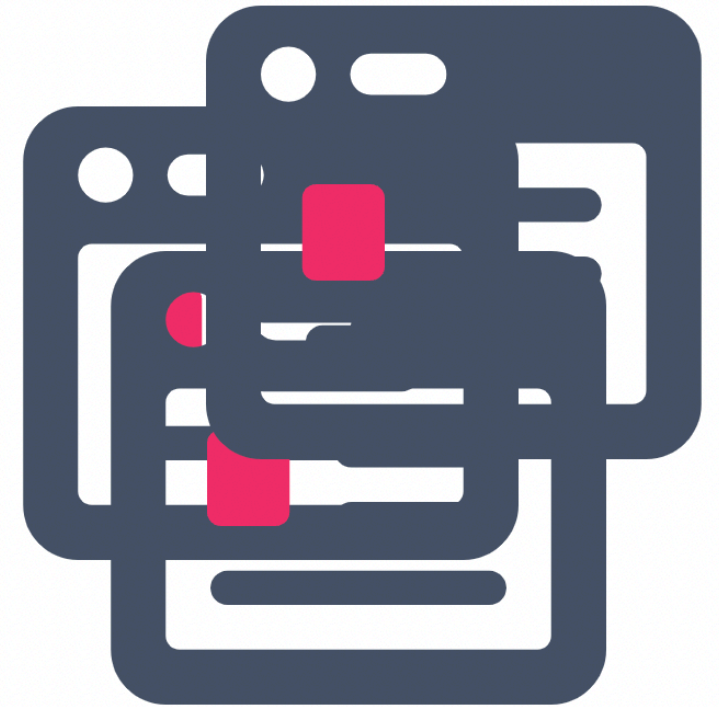
**MPA**  
**Multi Page Application**



# SSR



**Server Script**



**MPA**

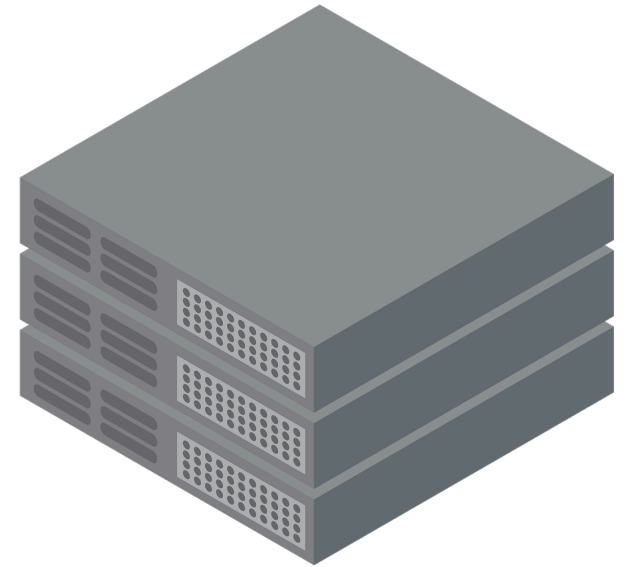
**Multi Page Application**

## AJAX

Asynchronous JavaScript And XML



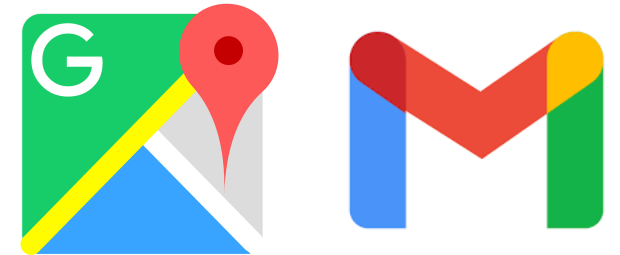
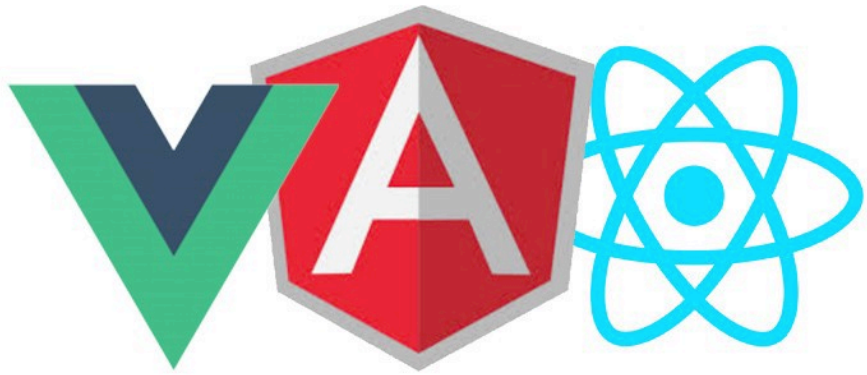
Client



Server



# CSR



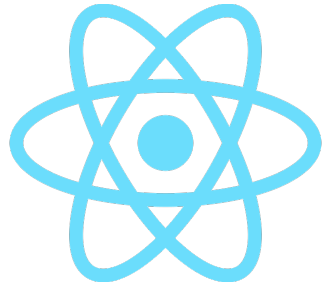
## SPA

Single Page Application





# Universal Rendering (CSR + SSR/SSG)



static sites < SSR < CSR < Universal Rendering ??

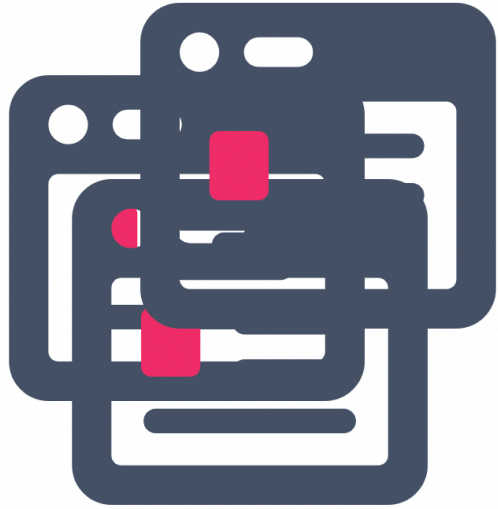
절대 L L



# SSR, CSR, MPA, SPA 관계



**SSR**



**MPA**

Multi Page Application

**SSR + CSR**



**CSR**



**SPA**

Single Page Application

**SSR  $\neq$  MPA**  
**CSR  $\neq$  SPA**



# CSR, SSR + SSG

## 분석 및 비교



# only SSR

## 동작방식

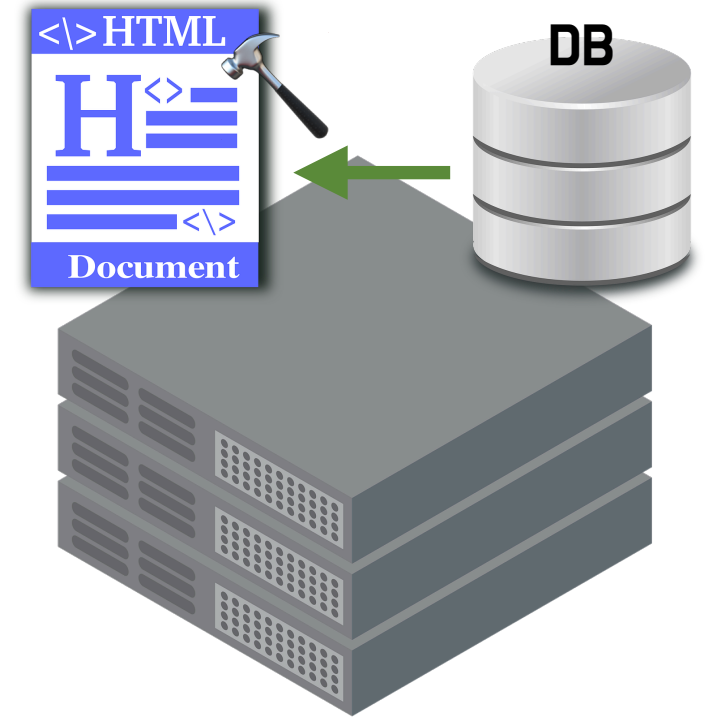


Client

www.test.com



1. 완성된 HTML  
2. JS



Server



# only SSR

# 장,단점

## 장점

- 첫번째 페이지의 빠른 로딩
- SEO에 유리함  
(Search Engine Optimization)

## 단점

- 깜빡임
- 서버 과부하 가능성 높음



# only CSR

# 동작방식



Client

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="/logo192.png" />
13    <link rel="manifest" href="/manifest.json" />
14    <title>React App</title>
15    <script defer src="/static/js/bundle.js"></script></head>
16    <body>
17      <div id="root"></div>
18    </body>
19  </html>
20
```



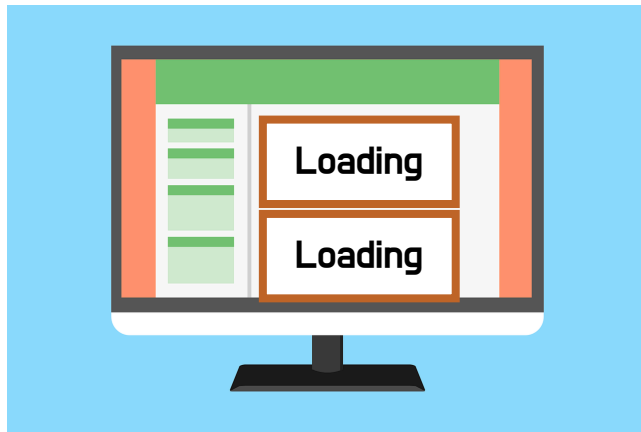
Server



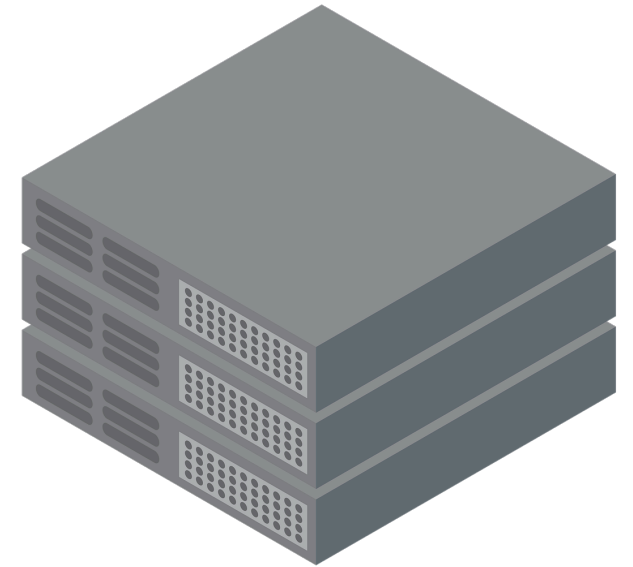
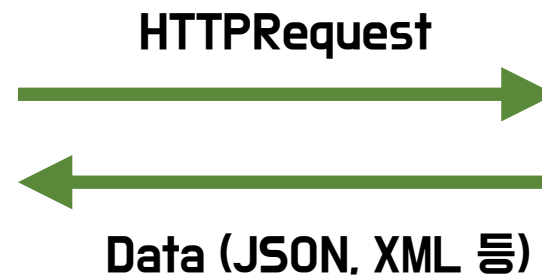


# only CSR

# 동작방식



**Client**



**Server**



# only CSR

# 장,단점

## 장점

- 부드러운 화면 전환
- 서버 과부하 가능성 낮음

## 단점

- 첫 페이지 로딩이 느림
- SEO에 불리함



# CSR,SSR 비교

|                | CSR | SSR |
|----------------|-----|-----|
| 첫번째 페이지 로딩 속도  | 느림  | 빠름  |
| 페이지 전환에 따른 깜빡임 | 없음  | 있음  |
| SEO            | 나쁨  | 좋음  |
| 서버 과부하 가능성     | 낮음  | 높음  |



# SSG

# 동작방식

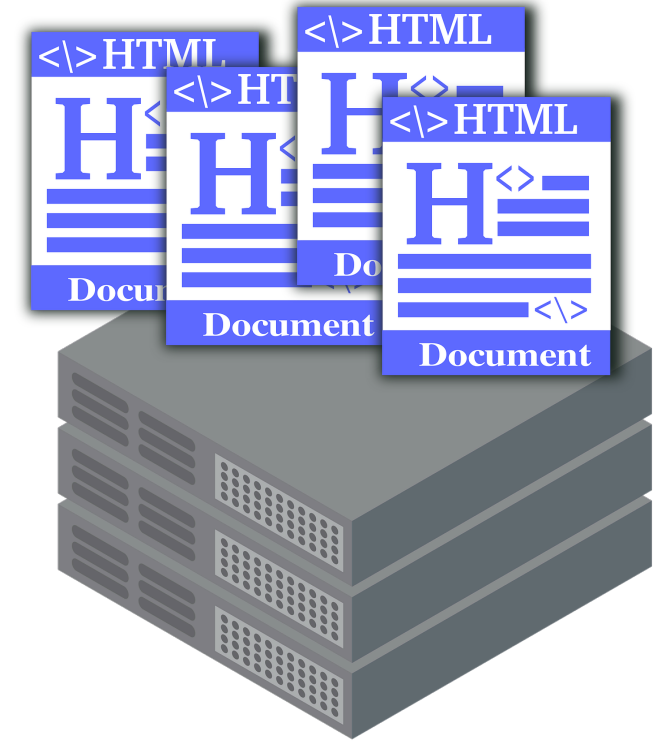


Client

www.test.com



1. 완성된 HTML
2. JS



Server



# SSG vs SSR

## SSR

- 데이터가 항상 최신 상태 유지
- 자주 변경되는 사용자별 데이터

## SSG

- 정적 사이트
- 바뀔 일이 거의 없는 페이지



# CSR + SSR/SSG (Universal Rendering)



# CSR + SSR(SSG)

ex) 첫 페이지: SSR, 그 이후 페이지들은 CSR

## 동작방식

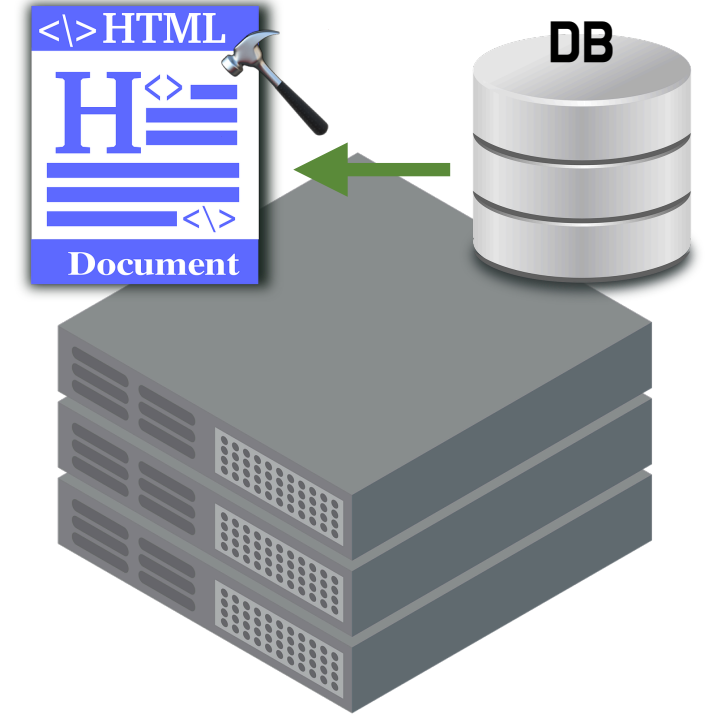


Client

www.test.com



1. 완성된 HTML
2. JS



Server



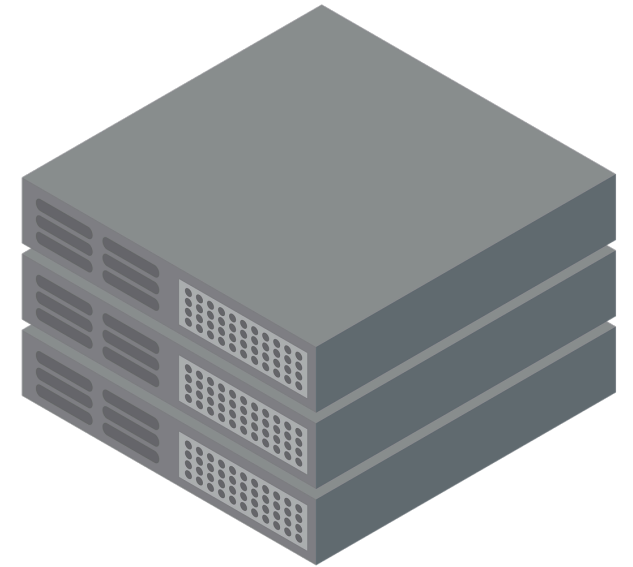
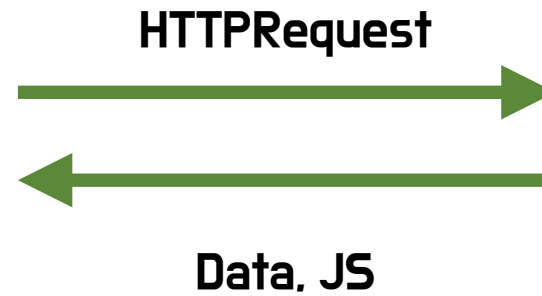
# CSR + SSR(SSG)

ex) 첫 페이지: SSR, 그 이후 페이지들은 CSR

## 동작방식



Client



Server





## 장점

- 첫 페이지 빠른 로딩
- 첫 페이지에 대한 SEO 유리
- 부드러운 화면 전환

## 단점

- 매우 높은 코드 복잡도
- 프레임워크 사용 시 블랙박스 영역 생김



# 감사합니다..!!



# 참고자료

- Rendering on the Web) <https://developers.google.com/web/updates/2019/02/rendering-on-the-web>
- The Benefits of Server Side Rendering Over Client Side Rendering) <https://medium.com/walmartglobaltech/the-benefits-of-server-side-rendering-over-client-side-rendering-5d07ff2cefe8>
- 어서 와, SSR은 처음이지? - 도입 편) <https://d2.naver.com/helloworld/7804182>
- [정직하게 배워보는 Next.js] (번외) 웹의 발전 과정으로 보는 CSR 그리고 SSG와 SSR) <https://wonit.tistory.com/361?category=829651>
- AJAX-나무위키 <https://namu.wiki/w/AJAX>
- 서버사이드 렌더링 (개발자라면 상식으로 알고 있어야 하는 개념 정리 ★)) <https://www.youtube.com/watch?v=iZ9csAfU50s>
- [10분 테코톡] 🎨 신세한탄의 CSR&SSR) <https://www.youtube.com/watch?v=YuqB8D6eCKE&t=645s>

