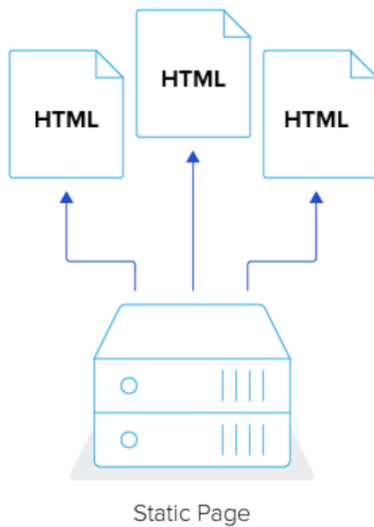


## 정적 웹(Static Web)

**편의점**(완성품/진열 상품 그대로 클라이언트가 이용)



언제 접속해도 같은 resource 를 보여주는 웹

서버는 프로그래머가 작성해 놓은 파일들(최종본)을 그대로 클라이언트의 브라우저에 전달.

(like 진열대에 놓인 상품) 브라우저(클라이언트)는 이 완성된 코드들을 읽음

Ex. 회사나 학교 소개 페이지, 댓글 기능이 없는 블로그 페이지 (접속시마다 내용이 변할 필요가 없는 사이트들)

But!!

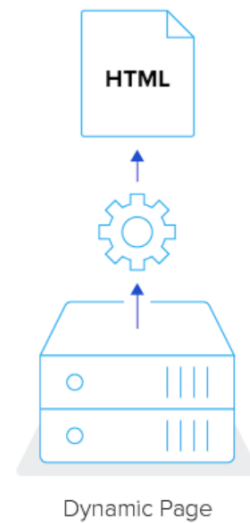
정적 웹이라고 해서 들어갈 때마다 완전 그대로인 화면은 아님.

시간이나 랜덤함수로 매번 다양한 화면을 보여주거나, 아예 직접 서버에 추가 정보를 요청해 받아온 내용을 보여주기도 함. [모두 Javascript로 구현 가능]

결국 정적 웹의 기준은 접속할 때마다 받게 되는 HTML, CSS, JavaScript 코드들, 동봉된 이미지, 동영상 등의 파일들이 같은가. 정확히는 서버에서 이것들을 매번 가공해서 제공하는 것이 아니라 프로그래머가 작성해서 갖다 놓은 제품들. 진열되어 있는 걸 그대로 가져가게 하나는 것.

## 동적 웹(Dynamic Web)

**식당**(주문 → 주방(서버)에서 그때그때 요리가 이루어짐)



수시로 바뀌는 데이터를 DB로부터 받아 그때그때 다른 화면을 보여주는 웹

웹페이지로부터 요청을 받으면 서버에서 추가적인 처리 후 클라이언트 브라우저에게 응답 보냄.

Ex. 사용자가 글을 올리는 게시판, sns 웹페이지

But!!

예전에 많이 이용된 동적 블로그 → 정보(작성자, 제목, 내용, 작성일, 댓글 등)가 DB에 따로 저장되고 접속시마다 이것들을 페이지로 가공해서 보여주는 방식

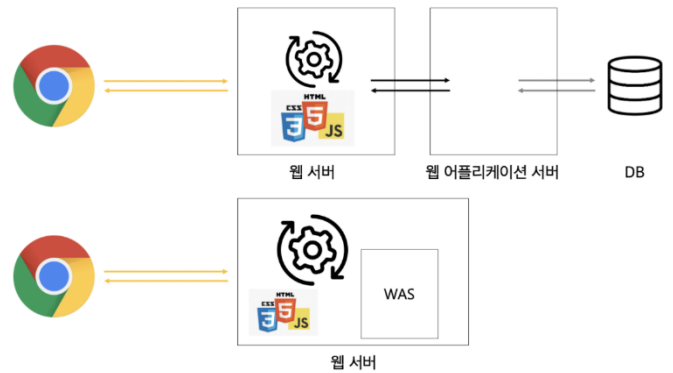
단점: 1모든 정보가 DB에 저장되다 보니 보관, 관리가 까다로울 수 있음. 2같은 글을 보여주느라 매번 서버가 일을 하도록 코딩하고 관리하는 것은 비효율적

요즘 많이 이용되는 정적 웹 생성 툴 (Jekyll, Hugo 등) → 글들을 하나하나 html 파일 단위로 잡지처럼 진열하여 운영. 보관도 글들을 복사해 따로 저장해두기만 하면 됨. (서버까지 갈 필요 없음)

## 요약 정리



## 정적 웹 구조



## 동적 웹의 다양한 구조

	정적	동적
생성 과정	개발자가 HTML, CSS, JavaScript를 사용자에게 보여질 모습, 최종본으로 코딩해서 서버에 올려놓음.	수시로 바뀌는 데이터를 DB로부터 받아 그때그때 다른 화면을 보여줌.
장점	<ul style="list-style-type: none"> <li>- 빠르다 (서버간 통신↓)</li> <li>- 비용이 적다(웹 서버만 구축)</li> </ul>	<ul style="list-style-type: none"> <li>- 서비스가 다양하다 (다양한 정보를 동적으로 생성)</li> <li>- 관리가 쉽다</li> </ul>
단점	<ul style="list-style-type: none"> <li>- 서비스가 한정적이다 (저장된 정보만)</li> <li>- 관리 불편 (추가/삭제/수정 모두 일일이)</li> </ul>	<ul style="list-style-type: none"> <li>- 느리다</li> <li>- 비용이 크다(웹 서버+어플리케이션 서버)</li> </ul>
!!	<p>사용자에게 보여줘야 할 데이터가 자주 바뀌지 않는 성질을 가진다</p> <p>→ 정적 웹</p>	<p>사용자에게 보여줘야 할 데이터가 상시로 바뀌는 성질을 가진다</p> <p>→ 동적 웹</p>