

23-10-19

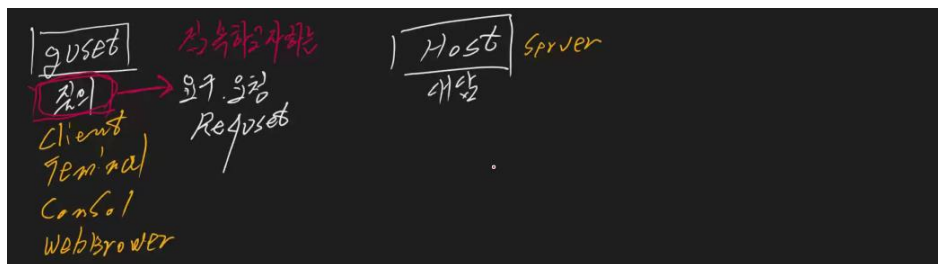
- guest : 질의 -> 요구 요청 request를 한다
 - Client
 - Teminal
 - Consol
 - webBrowser
- Host : 대답

홈페이지가서 로그인 하는 것은 => 우리는 client 이다.

가상머신의 설치한 mariadb와 리눅스os는 guest이다.(Host인 가상머신 입장에서)

Mariadb와 리눅스os 두개를 따지면

- Host : 리눅스os
- Guest : mariadb



<Tree 구조>

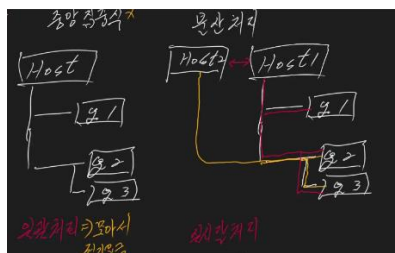
- 중앙집중식 : 일괄처리방식 -> 모아서 처리

Host가 다 관리하는 구조이다 -> 다 관리하여 누가 뭐 하는지 다 안다, 관리의 이점이 있다

Host가 멈추면 나머지도 다 멈추게된다

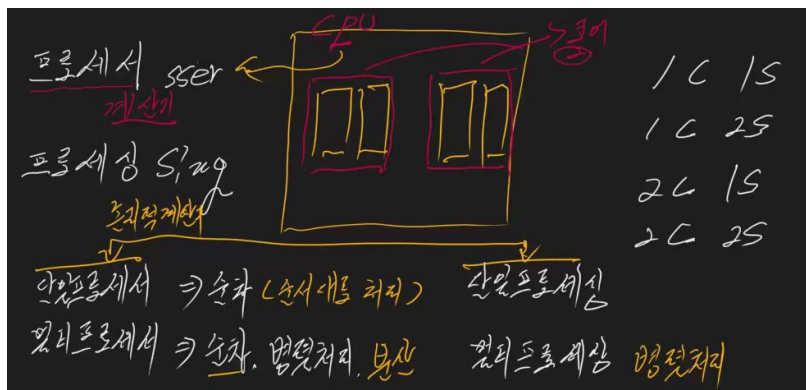
- 분산처리방식 : 실시간처리방식 -> 바로바로 처리

Host가 나눠서 관리 -> 응답과 처리속도가 빠르다, 한 곳에 몰아넣지만 한 곳이 삭제되어도 작동가능



<프로세서와 프로세싱>

- 프로세서 : 계산기(일하는 아이), 코어
 - 단일프로세서 : 코어1개 => 순차처리만 가능(순서대로 처리),
단일은 하나의 텅어리
 - 멀티프로세서 : 코어 여러 개 => 순차처리도 가능하지만 병렬 처리도 가능
일할 수 있는 계산기가 여러 개인 것(분산 처리)
- 프로세싱 : 논리적 계산기, 쓰레드
 - 단일프로세싱 : 하나의 일만 하는 것
 - 멀티프로세싱 : 여러 개 동시에 처리 병렬처리(병렬처리가 되면 순차처리도 가능)



- 분산처리 : 나눠서 처리한다
- 순차처리 : 순서대로 일처리 해야함, 이미 처리하기로 정해진대로 처리하는 것
(동기식 이라고도 함)
- 시분할 처리방식(TDM) : 시간을 쪼개서 사용

순차처리 : A가 10초, B가 15초, C가 25초 => total 50초

분산처리 : A가 10초, B가 15초, C가 25초

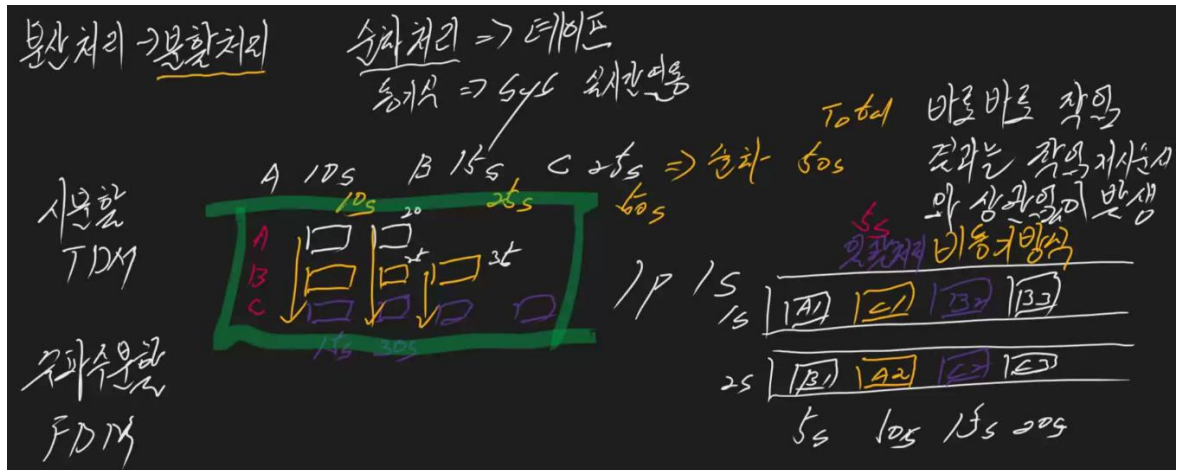
A 5초 -> B 5초 -> C 5초 -> A 5초 -> B 5초 -> C 5초 -> B 5초 -> C 5초 -> C 5초

결국 A가 순차처리로 20초에 끝난 것을 분산처리를 하게 되면 20초에 끝남

⇒ 이것을 뒤집어 놓고 생각해 보면 (C가 맨 위로 온다고 생각 A가 맨 아래로 간다고 생각)

C 5초 -> B 5초 -> A 5초 -> C 5초 -> B 5초 -> A 5초 -> C 5초 -> B 5초 -> C 5초

50초에 끝날걸 30초에 끝난다



- 비동기화 방식 : 바로 작업할 수 있지만 결과는 작업개시 순서와 상관없이 발생
(ex. 이메일, 자바스크립트 Call Back 함수)
- 동기화 방식 : 요청과 결과가 동시에 일어남 (ex. 회원가입, 전화거는 것)

주파수분할방식(FDM) : 주파수 폭 지정해 준 것

다른 것과 섞이지 않게 보내주는 역할

작업의 순서와 상관없이 공통으로 사용하는 것이 있을 것

➔ 공중통신망 : 내 신호와 다른 신호 섞이지 않게 해주는 것

유동아이피 주는 것과 똑같다

➔ 공중통신망 PSN ⇒ 공중통신망 데이터
↳ 패킷 packet 통신시 사용되는 데이터 크기
64KB ⇒ 이내

<전자통신>

최초의 통신망 -> 봉화, 모스부호, 팽과리, 소리

통신 -> 떨어져 있는 사람과 소통(통신)하는 것, 사용하는 도구는 전파, 빛, 전자냐에 따라 다르다

- 매쉬형태(그물망)

A와 B와 연결을 하면 직선으로 연결하면 됨

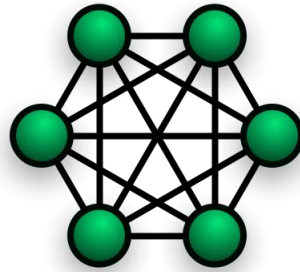
A와 B와 C가 연결을 하면 동그라미 형태로 모두 각각 연결해주면됨

이것이 점점 늘어나면 늘어날수록 통신회선이 늘어난다 (통신확보측면엔 좋다->안전하다)

이 형태가 매쉬형태 즉, 그물망이다

특징

1. 통신선 설치비용 ↑
2. 관리하는 비용 ↑
3. 새로운 통신망 생기면 설치 비용



- 공용망

A B C D E 를 연결하는 공용 망이 존재

간편하고 간결한 상태

그러나 공용망 항상 거쳐가기 때문에 혼선되는 상태가 된다(보안에 문제가 생긴다)

-> 공용혼선에 도청하기가 쉬웠다

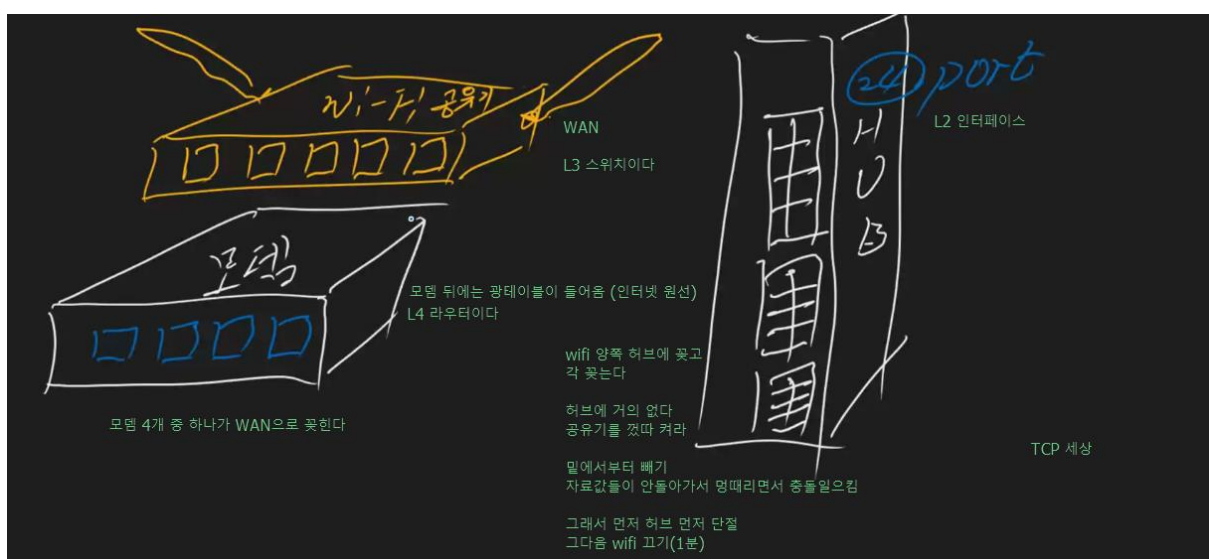
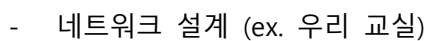
여기에 전전자교환기가 들어온다

-> 모두 전송되지만 모든 정보를 보여주지 않아서 보안에 유리하다



네트워크 프로토콜이 통신하는 구조를 7개의 계층으로 분리하여 각 계층간 상호 작동하는 방식을 정해놓은 것이다. -> 통신에 기준이 된다

A. 3~7단계 포함된 걸 그래픽으로 활용할 수 있게 하는 것



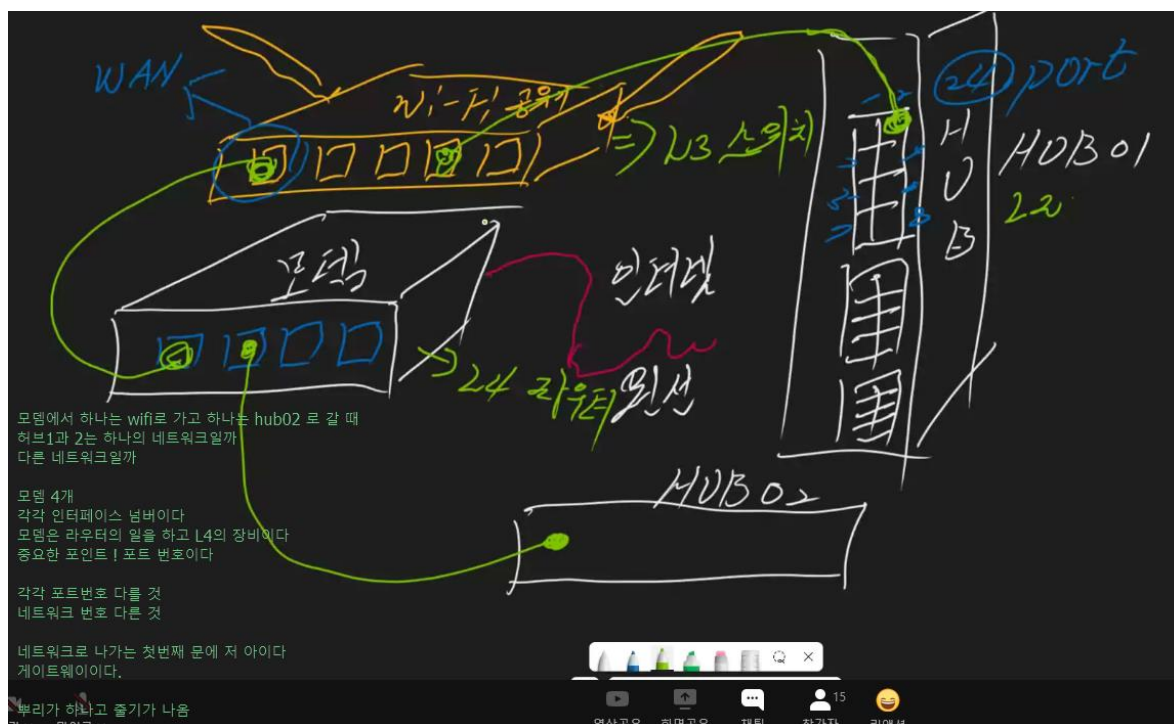
우리 교실에 인터넷이 꺼진적이 있다 그래서 우리교실의 연결상태를 살펴보니

위와 같이 HUB와 wi-fi와 모뎀으로 구성이 되어 있었다 그리고 모든 컴퓨터가 하나 하나의 HUB에 연결되어 있는 상태가 아니라 연결선이 길어지기 때문에 나눠서 HUB로 각각(2대) 연결되어 있는 상태였다.

Q. 모뎀에서 하나는 wi-fi로 가고 하나는 Hub02로 갈 때 Hub01과 Hub02는 하나의 네트워크일까? 다른 네트워크일까? (팀 질문이었다 같이 의논해보기 시간)

A. 다른 네트워크이다 모뎀에 각 연결되어 있는 곳은 각각 포트 넘버이다

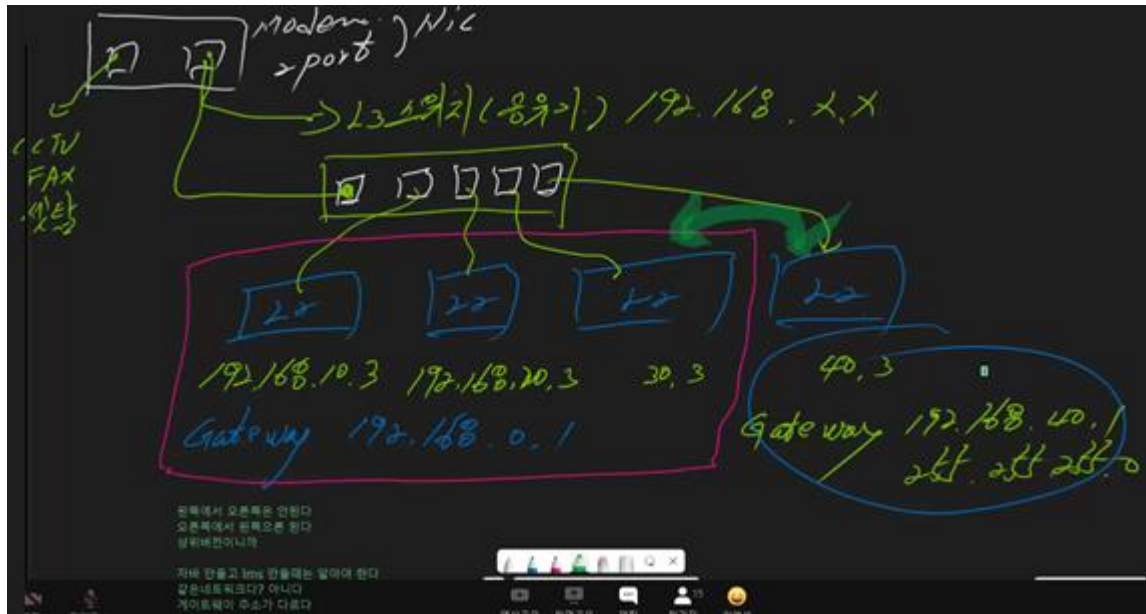
위에는 모뎀이 4개로, 4개로 나뉜다.



모뎀 2port만 허용 -> Nic도 가지고 있다

오른쪽이 스위치(공유기) 192.168.x.x

왼쪽이 FAX 셋업



게이트웨이를 위와 같이 주면 이 안에서만 작동됨

IP 0.0.0.0 ~ 255.255.255.255
8bit 32bit 32bit 32bit
IPv4 주소 32bit
IPv6 주소 128bit
HTTP port 80 인터넷이 사용되면 port 다.
HTTPS port 8080 보안이 있는 인터넷 주소
SSL 보안 목적 적용
↳ 서버링사이드 방식 => 서버엔딩 => 서버링

이전버전 사용하는 이유 : 많은 사람이 사용하고 있다.

0~255 => 0~7

2^8 => 2^3

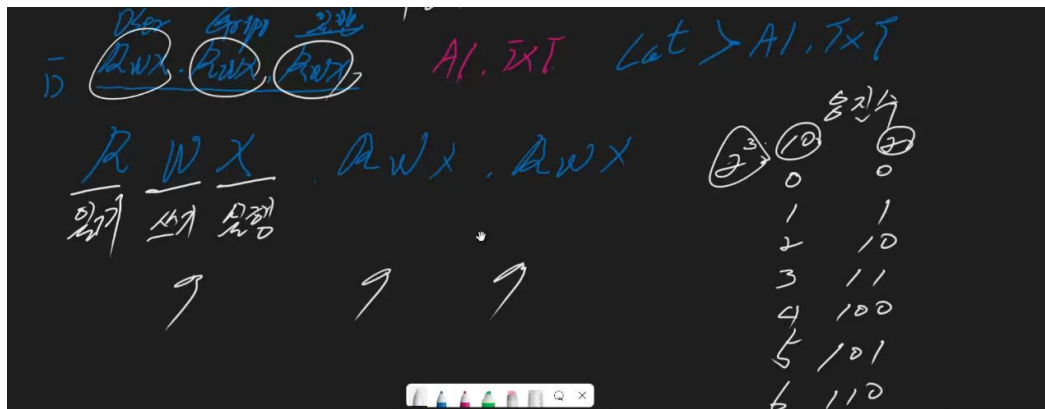
1111 1111 => 111(2^3 2^2 2^1) : 리눅스에서 파일관리, 디렉토리 관리

코드로 사용하기 때문에 그렇다

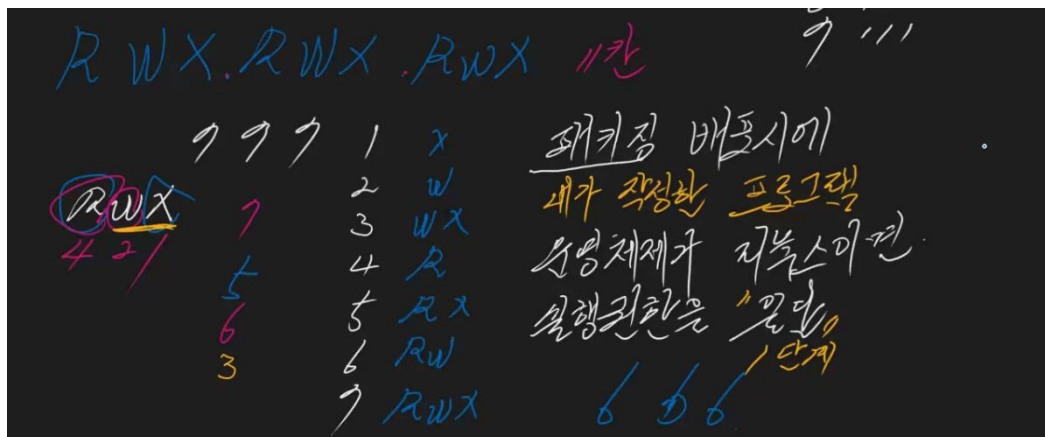
(8 4 2 1) CODE => 문자를 만들기 위한 코드이다
: 가중치코드

User.Grop.일반 -> 총 11칸

파일 디렉토리 권한 여기에서 한다.



➔ 내가 만든 웹 페이지를 웹에 배포하는 것




```

→ linux-deep ls -l
total 8
-rw-r--r-- 1 username staff 29 4 12 17:44 helloworld.js
drwxr-xr-x 2 username staff 64 4 12 17:43 linux

```

여기서 `username` 은 사용자 이름이다.

출력의 표현 중 첫 시작인 `-` 와 `d` 는 not directory, directory 를 나타낸다.

폴더이면 `d`, 파일이면 `-` 로 표시된다.

그리고 이어지는 `rwX` 는 각 알파벳이 read, write, execute인 읽기, 쓰기, 실행에 대한 각각의 권한이 있는지를 나타내는 표현이다. 여기서 `rwX` 가 3번씩 반복되는 이유는 사용자, 그룹 그리고 나머지에 대한 권한을 표시하기 때문이다. 그러므로 위에서 helloworld.js의 파일에 나타나는 권한은 소유자는 읽기와 쓰기가 가능하고, 그룹과 나머지는 읽기만 가능하다는 뜻이다.

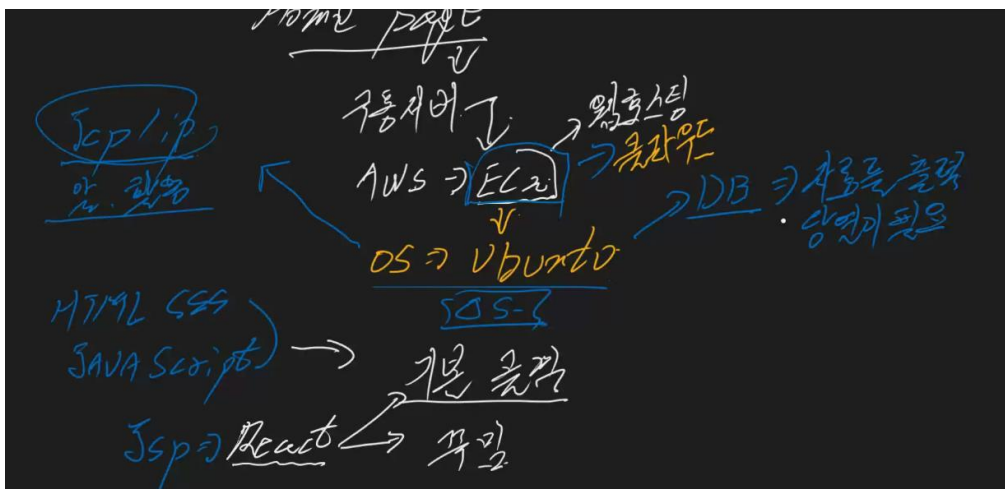
우리가 왜 학습해야 할까?

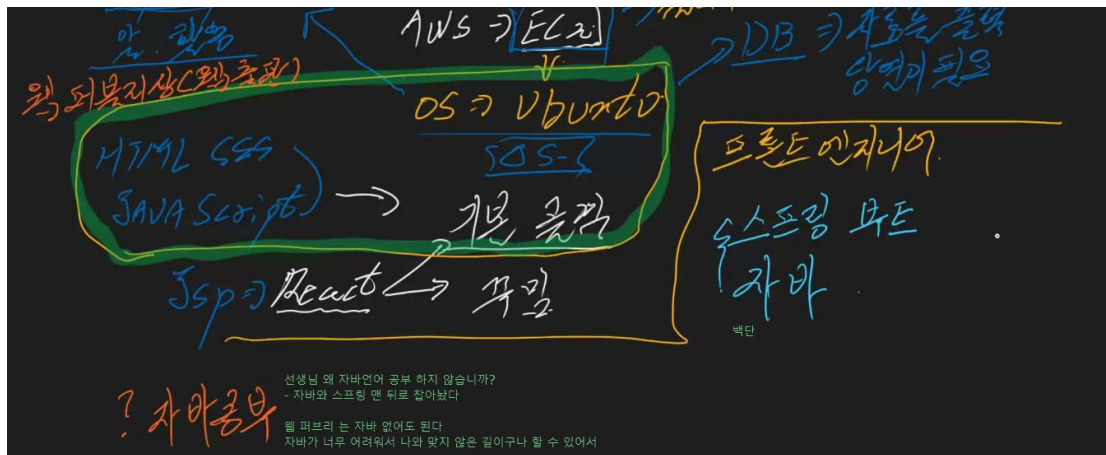
1. 웹 페이지 구현 위하여 학습 (PC 기준)

홈페이지만드는데 필요한 것

Homepage

- A. 구동되기 위한 서버 필요 -> AWS에서 할 것 -> EC2 (웹호스팅, 클라우드) -> OS => ubuntu (TCP/IP 알아야 활용) -> 웹페이지 ip 동일해야지 사람들이 들어온다[여기까지 프론트 엔지니어가 끝남] -> DB 자료를 출력하는 것 당연히 필요





HTML CSS

JAVA SCRIPT

// 위에 두개가 기본 골격

JSP => React

// 기본골격 과 꾸밈

- 와이어샤크 : 나중에 알도록 하면 좋다 통신 보안 때문에
- 사이트 : stack over flow
(오류나, 질문 같은 것 여기서 질문하기, Document를 기준으로 보기 때문에 여기서 확인하는 것이 정확하고 좋음.)