

- 최근 F/E¹ 에서 데이터 처리시 빠른 응답과 구현 및 적응이 빨라 on-boarding에 장점이 생긴다.
- 한계는 분명하다 결국 Scalability 문제에 도달하므로 데이터는 원점대로 B/E 에서 다뤄야한다.
- Computing Science(CS, 컴퓨터과학?) 는 1개의 요리를 터득하면 나머지 9개는 연결된다.
- **Communication(의사소통)** : 책을 많이 읽는 습관이 중요하다. 명확한 전달력이 용이해진다.
- 강사의 강의내용을 변형없이, 하나도 빠짐없이 적어라. 이후 요약을 따로하는 것이 복습이다.
- **Compatibility(호환성)** : 어떤 환경에서든 연동, 연결되는 것이다. 이는 팀원들간에도 적용된다.
- **Engineering(기술)** : 각 분야에 발생한 기술적인 결함을 해결하는 것
- **Technical(기능)** : 배경지식과, 컨텍스트(문법)에 맞게 사용되는 것, 잘 작동되는 것이다.
 - NCS(국가직무능력표준) 기반의 직업훈련은 기능에 해당한다.
 - 배경지식과 개념정립이 잘되어 있으면 기능적 구현은 쉽다.
 - 컴퓨터는 0or1의 반복이다.
 - 기능(Technical)적 반복 : 좋아하거나 쉬운것을 많이 반복해야 한다
 - 코드와 로직은 Attitude (태도, 자세) 를 가지고 수없이 반복하길 권장한다..
 - 눈으로 알지만 손으로 구르지 않으면 차후 개발단계에서 차이가 현격하게 벌어진다.
 - 실무에선 초반에 기대감이 낮다. 일단 반복하되, 6개월 배운 것의 변형일 것이다.
- **Project Based Learning(PBL, 조별과제)** : 10~30일 간격 변동으로, 소통 및 호환을 위함이다.
- 6개월간의 전체 강의 진행은 다음과 같다.
 - CS(Computing Science) : 자료 구조에 관하여
 - DB(Data Base) : 자료 저장, 자료->의미->추출(정보처리)->정보
 - 데이터를 규칙에 맞게 정리(Array), 의미가 있으면 정보가 된다.
 - OS(Operating System) : 서버용도 / PC - window, Linux, Mobile phone - IOS, Android
 - Linux 는 일반적용도에서는 타이핑이 필요하므로 사용이 불편하다. 하지만 용량이 적어 CPU 사용에 따른 요구가 적어 용이하다.
 - Ubuntu 20.04 lt (서버용)
 - Server(질문에 대답) <-> Client (질문을 요청자)
 - Host <-> Guest / 회사마다 부르는 용어차이
 - HTML/CSS/Javascript : 웹 페이지의 기본, F/E
 - 미니 프로젝트 1
 - 리액트, 타입스크립트, 뷰 등
 - 미니프로젝트 2
 - Java, Spring
 - 최종프로젝트
- **Integrated Development Environment(IDE, 통합개발환경)³ :**
 - IntelliJ : IDE중 하나로 java,kotlin 같은 안드로이드 앱을 위한 최적화 앱
 - IntelliJ 는 유료지만 추가설정이 적은 편이다. 많이 사용하는 Visual Studio 는 무료지만 VS Code 는 유료이며, eclipse 는 추가 설정할 사항이 많은 편이다.
 - Learning Management System(LMS, 학습관리시스템) : 온라인으로 학생들의 성적, 진도, 출석등을 관리해주는 시스템
- PBL 1차 준비
 - 맛의 레일로드(팀명, 제품명)
 - 기획회의 : 계획서 -(타인에게 전달시-> 기안서 -> 제안서 -> 요구사항분석서(ROC)

¹ 코딩에 많이 쓰이는 영어

² 개발자라면 CS 의 의미가 Common Sense 일 정도로 빠삭해져야 한다.

³ 코딩, 디버그, 컴파일, 배포 등 프로그램 개발에 관련된 모든 작업을 하나의 프로그램 안에서 처리하는 환경을 제공하는 소프트웨어