**Day 03 (●’◡’●) 디버깅 중요!**

모니터 – 문자 기반, GUI(그래픽) 기반으로 출력됨.

1차원 정보 : 벡터 Vector

2차원 정보 : 행렬 Metrics(x, y) -> 관계형 데이터 모델

문자 기반 : 콘솔 Console

SW 공학 : 최소한의 비용으로 최대한의 효과를 만들기 위한 학문 (생성 -> 폐기까지의 비용을 줄이기 위해서)

초기에는 개발비용, 생성비용이 높았다. 하지만 코드를 공개하면서 생성 비용이 줄어들게 됨.

따라서 최근에는 유지보수비용이 훨씬 많이 발생하게 되었다. 그래서 유지보수 비용을 줄이기위해 여러 방법이 생겨났다.

* 프로그래밍 방법론 : SW 공학적 관점에서 명령어를 만들어내는 방법을 효율적으로 하기 위함
  + 초기 함수 지향형 : 생성 비용 줄이기 = **코드**의 재사용성 높이기 = 라이브러리 기술 발달 : 모듈화(**함수** 단위) 즉, 코드의 재사용성이 높아짐
  + 객체 지향형 : 유지보수 비용 줄이기 = 데이터 변화에 둔감해지게 하기 = 모듈화(object(data+code) 단위)
* 프로젝트 방법론 : 폭포수, 애자일, 칸반, 스크럼반, 하이브리드 등등
* 함수 (function)

기능, 기능을 수행하다.

수학, 프로그래밍 등 사용

* 객체 (object)

단일 책임 = 단일 역할 = 관련된 다수의 기능 + 관련된 다수의 데이터

독립적 실행 단위

* 추상화 = 개념적인 것 (구체적인 것의 반대)

함수를 구체적으로 잘 몰라도 함수를 사용할 수 있음. 내부적 함수 코드를 몰라도 함수 사용 가능

꼭 필요한 근본적인 특성에 집중해 여타의 부분이나 상세 내역을 없애 나가는 과정

공통적 부분에 집중해 일반화하는 과정

등등 객체 지향에서는 더 다양한 역할이 있음.

* 다형성 : 여러 곳에서 사용할 수 있는 것