

Python 1

컴퓨터 OS

- Operating System, 운영체제
- 우리의 프로그램이 동작할 수 있는 구동 환경
- 프로그램은 OS에 의존적(dependent) => OS에 맞춰서 개발

파일 시스템

- OS에서 파일을 저장하는 **트리구조** 저장 체계
- 컴퓨터 등의 기기에서 의미 있는 정보를 담는 논리적인 단위
- 모든 프로그램은 파일로 구성되어 있고, 파일 사용
- **디렉토리 (Directory)**
폴더, 파일과 다른 디렉토리 포함 가능
- **파일 (File)**
컴퓨터에서 정보를 저장하는 논리적인 단위
파일명, 확장자로 식별
실행, 쓰기, 읽기 가능 (chmod 777)
- 파일시스템 Root 디렉토리로부터 시작하는 트리구조
- **경로**: 컴퓨터 파일의 고유한 위치, 트리구조상 노드의 연결
 - **절대 경로**: 루트 디렉토리부터 파일위치까지의 경로
 - **상대 경로**: 현재 있는 디렉토리부터 타겟 파일까지의 경로

터미널

- 키보드로 명령을 입력 프로그램 실행
- **Graphical User Interface, GUI**
- **Command Line Interface, CLI**: text를 사용하여 컴퓨터에 명령 입력하는 인터페이스 체계
 - Windows: CMD window, Windows Terminal
 - Mac, Linux: Terminal
- Console = Terminal = CMD창
어원: 디스플레이, 키보드 조합된 장치
현재: CLI로 입력하는 화면
- 각 터미널에서 프로그램을 작동하는 shell 존재
=> shell마다 다른 명령어 사용

Python

- 1991년 귀도 반 로섬이 발표
- 플랫폼 독립적
- 인터프리터 언어
- 객체지향
- 동적타이핑언어
- 처음 C언어로 구현되었음
- 자비로운 종신 독재자: 파이썬의 개선사항(PEP)에 대한 최종 의사결정자
- python(피톤)은 그리스 신화 속의 괴물뱀
- 몬티 파이썬이라는 코메디 그룹의 이름에서 유래

Python의 특징

- 플랫폼: 윈도우, 리눅스, 안드로이드, 맥OS, iOS 등 프로그램이 실행되는 운영 체제를 플랫폼이라고 함
- 독립적인: OS에 상관없이 한번 프로그램을 작성하면 사용가능
- 인터프리터: 소스코드를 바로 실행할 수 있게 지원하는 프로그램 실행 방법

컴파일러		인터프리터
소스코드를 기계어로 먼저 번역 / 해당 플랫폼에 최적화되어 프로그램 실행	작동 방식	별도의 번역과정 없이 소스 코드를 실행시점에 해석해서 컴퓨터가 처리할 수 있도록 함
실행속도가 빠름	장점	간단히 작성, 메모리가 적게 필요
한번의 많은 기억장소 필요	단점	실행속도가 느림
C, 자바, C++, C#	주요 언어	파이썬, 스칼라

- 사람이 알 수 있는 고급 언어를 기계만 알 수 있는 저급 언어로 변환
- 객체 지향적 언어

실행 순서가 아닌 단위 모듈(객체) 중심으로 프로그램 작성

하나의 객체는 어떤 목적을 달성하기 위한 행동(method)와 속성(attribute) 가지고 있음
- 동적 타이핑 언어

프로그램이 실행하는 시점에 프로그램이 사용해야할 데이터에 대한 타입 결정
- 파이썬 쓰는 이유
 - 쉽고 간단, 다양
 - 이해하기 쉬운 문법
 - 다양한 라이브러리 - 무엇이든 구현 가능, 대부분의 라이브러리가 이미 다른 사용자에 의해 구현되어 있음(특히 통계, 데이터 분석)
 - 이미 널리 쓰임