학습동아리 <1회차>

## 컴파일 언어 vs 인터프리티 언어

1. 컴파일 언어
   1. 원시코드(소스코드)를 모두 기계어로 변환 후 머신상에서 기계어 코드를 직접 실행
   2. 기계어 코드 생성을 위한 빌드 과정이 필요(빌드를 위한 시간 소요)
   3. 기계어 코드를 런타임시 직접 실행하기 때문에 실행 속도가 빠르다
   4. 대표적인 언어 : C, C++
2. 인터프리터 언어
   1. 원시코드(소스코드)를 기계어로 변화하지 않고 바로 실행
   2. 인터프리터라는 중간 매개체가 필요
   3. 인터프리터가 한줄씩 읽고 기계어로 바로 변환해서 실행, 빌드가 필요없음.
   4. 런타임시 실시간으로 읽어서 실행하기 때문에 컴파일 언어에 비해 속도가 느리다.
   5. 대표적인 언어 : Python, R, Ruby

<참고- 빌드 과정 (c언어 기준)>

소스코드(확장자: .c) 🡪 **[전처리기]** 🡪 전처리된 코드파일(확장자: .i] 🡪 **[컴파일러]** 🡪 어셈블리 코드(확장자: .s) 🡪 **[어셈블러]** 🡪 오브젝트 파일(확장자: .o) 🡪 **[링커]** 🡪 실행파일(확장자: .exe)

## Python 언어의 특징

1. 대화 기능의 인터프리터 언어
2. 동적인 데이터 타입 결정
3. 개발 기간 단축에 초점을 둔 언어
4. 간단하고 쉬운 문법
5. 고기능의 내장 객체 자료형 제공 : 리스트, 듀플, 딕셔너리
6. 메모리 자동 관리 (Garbage Collection) 기능 제공
7. 짧은 코드 : 데코레이터 기능 등 사용 가능
8. 넓은 확장성 : 다른 언어에서도 호출 시켜 사용 가능함.

## Python 개발 환경

1. 추천 에디터
   1. PyCharm : 파이썬 전용 에디터, community 버전 개인 무료
   2. Visual Studio Code : 파이썬 이외 다른 언어 개발에도 사용, 무료
   3. Sublime Text : 간단한 고성능 text 에디터, 무료
   4. Atom : 리눅스 계열 O/S의 대표적 에디터, 무료
   5. Spyder : Anaconda 설치시 기본 설치, 무료
2. Anaconda 란?
   1. Python 기본 패키지 제고
   2. 수학/과학 라이브러리 패키지 추가 기본 설치
   3. 인공지능, 빅데이터 관련 개발 할 경우 기본 python 보다 Anaconda 설치를 추천함.