

(HL 만도 및 HL Klemove)

Intelligent Vehicle School

– 이 소 영 강사

yisy0703@naver.com

<https://github.com/comlec/rapa>

1. 프로그래밍 언어

- 프로그래밍 언어는 컴퓨터와 사람이 소통할 수 있는 언어
- 사람이 컴퓨터에게 어떤 명령을 실행시킬 목적으로 설계되어서 컴퓨터와 의사소통을 할 수 있도록 해주는 언어

사람이 사용하는 언어

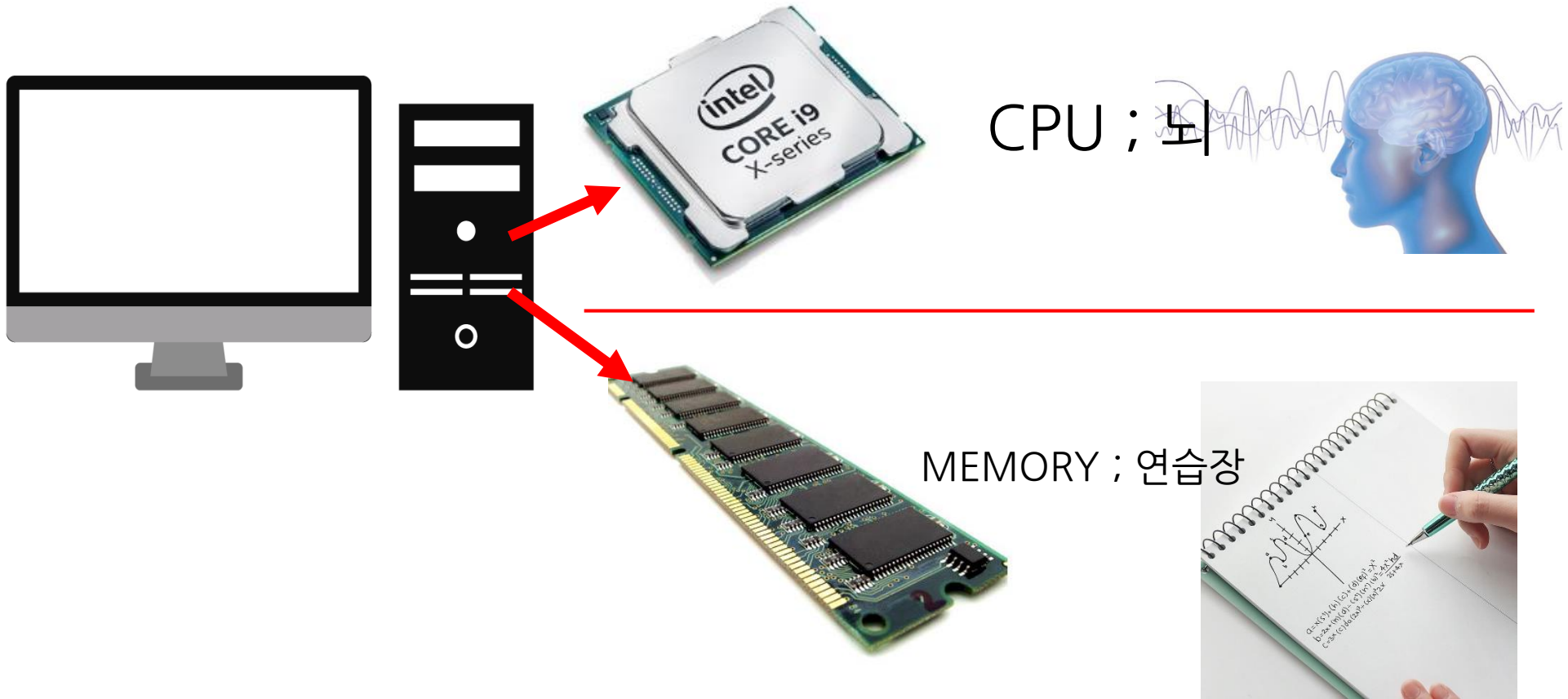


프로그래밍 언어



vs.

1.1. 프로그램 실행시



1.2. 컴파일 언어와 인터프리터 언어

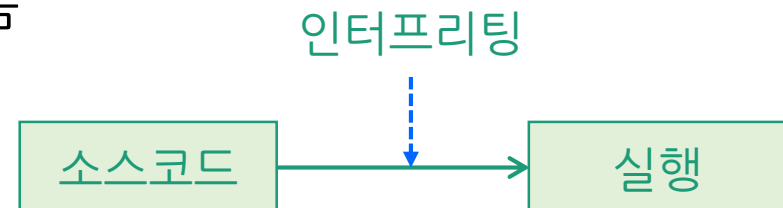
● 컴파일 언어

- 원시 소스코드를 컴파일이라는 과정을 통해 기계어로 번역한 파일을 만들고 이 파일을 통해 실행시키는 언어
- 컴파일 과정이 오래 걸리고 실행시키는 컴퓨팅 환경에 영향을 많이 받지만 실행 시 컴파일 된 실행 파일만 있으면 프로그램을 실행시킬 수 있으며 매우 빠른 속도로 실행되는 장점
- C, C++ 등



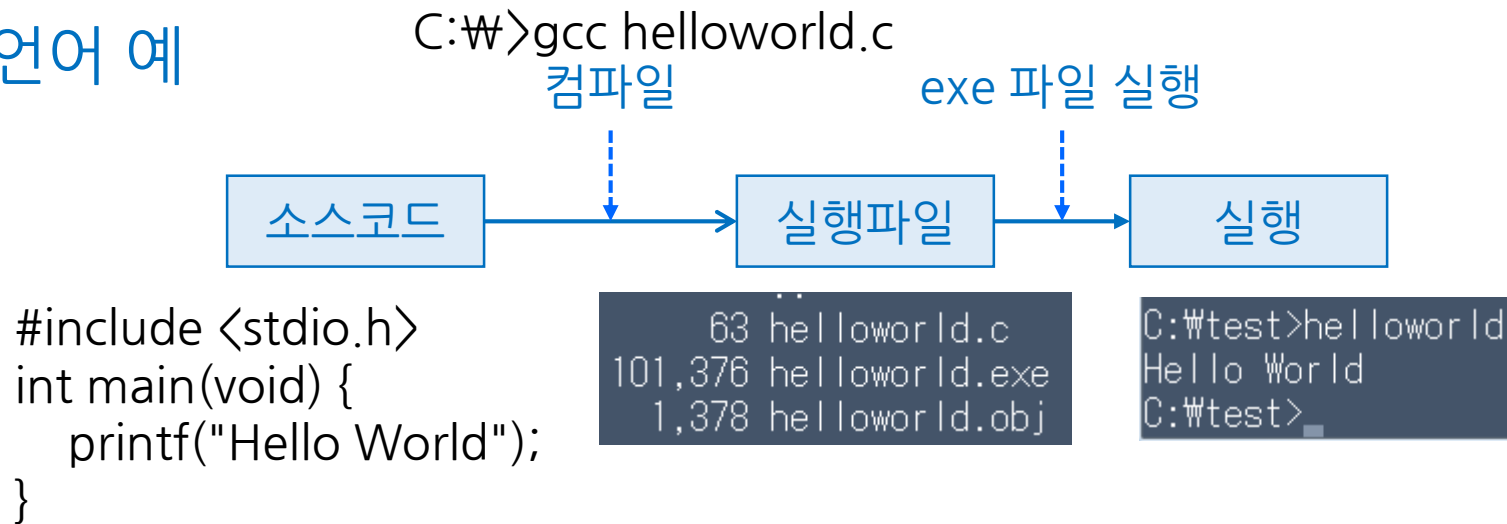
● 인터프리터 언어

- 인터프리터(해석기)에 의해 원시 소스코드를 한 줄씩 읽어 실행하는 언어
- 컴파일 언어보다 더 느리게 실행
- 코드를 빠르게 테스트해 볼 수 있고, 프로그래밍을 대화식으로 할 수 있기 때문에 교육용으로 사용되는 경우가 많음
- Python, JavaScript, R 등

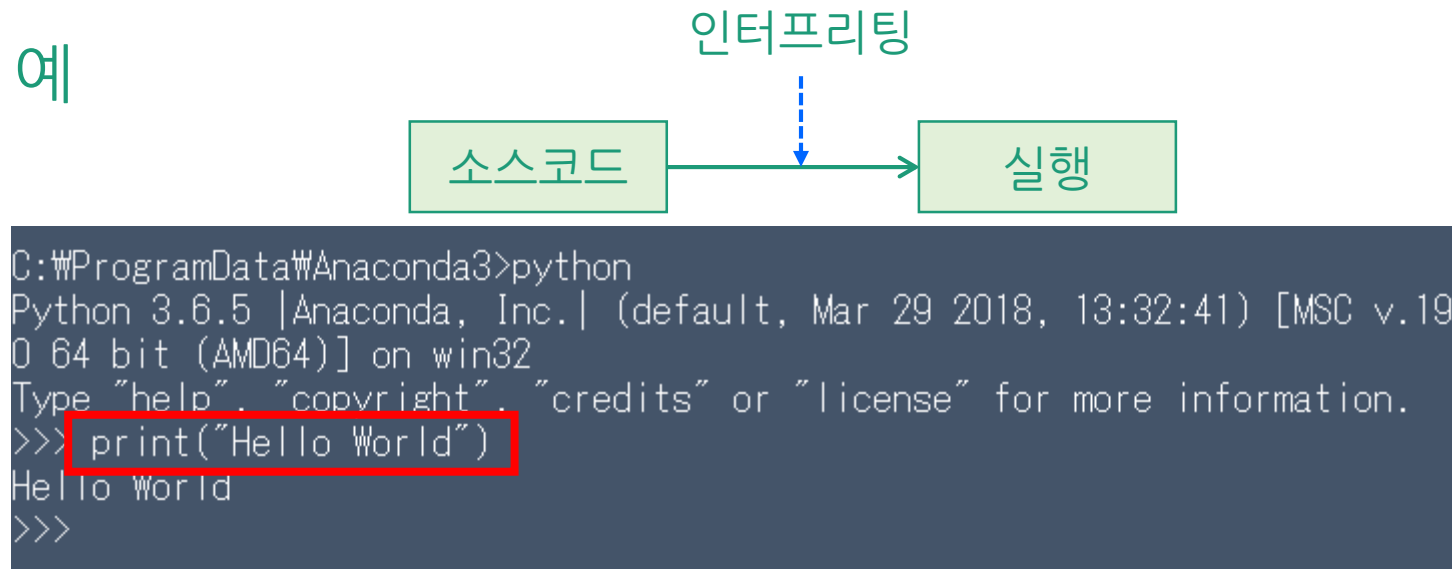


1.2. 컴파일 언어와 인터프리터 언어

● C언어 예



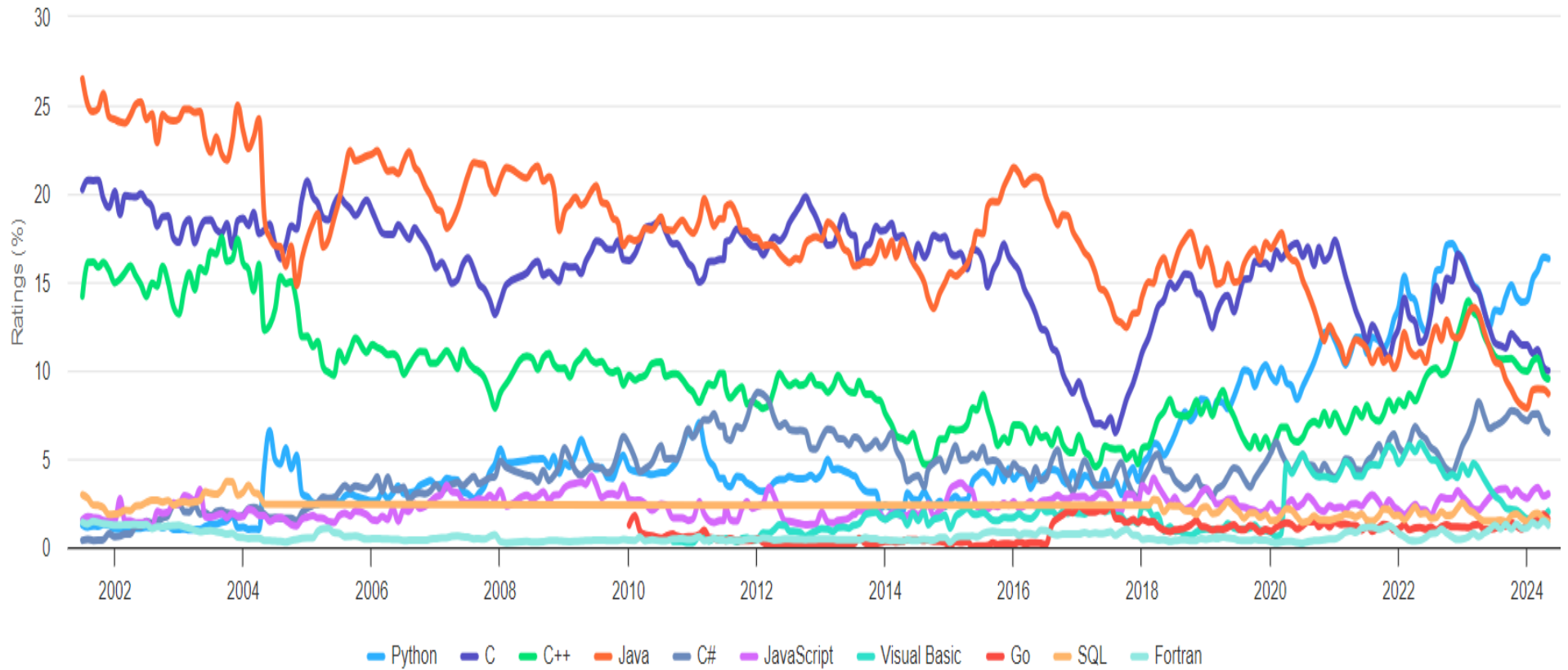
● 파이썬 예



프로그래밍 커뮤니티 인덱스

TIOBE Programming Community Index

Source: www.tiobe.com



source : <https://tiobe.com/tiobe-index>

프로그래밍 언어 참고자료 (도서구입은 비추)

- C
 - 나도코딩의 C언어 입문(나도코딩지음, 길벗, 2023)
 - 혼자 공부하는 C언어(서현우 지음, 한빛미디어, 2024)
- Python
 - The Python(허진경 지음, 부크크, 2021)
 - 파이썬 데이터 전처리 및 탐색 라이브러리(허진경 지음, 부크크, 2021)

추천 사항

- 이력서 자료

- Git
 - ① 매일 공부 정리 내용 및 소스
 - ② 프로젝트 내용
- 모의 프로젝트를 활용한 포트폴리오
- 저작권 등록(<https://www.cros.or.kr>) : 프로젝트 등록해서 이력서에 쓰면 반응이 좋음

1.내 소스 코드 저장

2.버전 관리 도구

3.소스 코드 공유

4.협업 공간

깃허브의 레파지터리(원격저장소) 생성하고 올리기

1. <http://www.github.com>에 가입하기(sign up) 후 코드가 포함된 메일 확인
2. Sign in
3. Create repository(올릴 공간명)
4. 소스를 내컴퓨터에서 github 인터넷으로 올리고 git 명령어를 사용하기 위해 git-scm을 다운받고 install (<https://git-scm.com/downloads>)
5. Git Bash에서 git 환경 설정
 - > git config --global user.name "comlec0703"
 - > git config --global user.email "comlec0703@gmail.com"
 - > git config --list (확인)

깃허브의 레파지터리(원격저장소) 생성하고 올리기

6. 내 pc의 repository 폴더에 다음의 파일 추가

① README.md (git 소개)

② .gitignore (git 공간에 올리지 않을 파일과 폴더 리스트)

(d:/rapa/.gitignore d:/rapa/source/1_C/.gitignore)

7. git 공간에 자료 올리기

> git init (초기화)

> git add . (공유 추가할 파일 더하기)

> git status (상태 확인 - 선택)

> git commit -m "first commit" (히스토리 버전 이름)

> git branch -M main (branch 생성 - branch 생성을 하지 않을 경우 master로)

> git remote add origin 본인깃주소 (연결고리)

> git remote -v (확인 - 선택)

> git push -u origin main (branch에 올리기 branch를 생성하지 않았다면 master)

깃허브의 레파지터리 생성

8. 내 pc와 git이 동기화된 거 확인 후, 추가 파일 만든 후 추가 올리기

- git add .

- git commit -m "second"

- git push -u origin main

9. 추가한 파일 수정

- git add .

- git commit -m "third"

- git push -u origin main

깃허브의 레파지터리 생성

- 10.협업 (웹에서 수정하거나 다른 장소에서 수정 후 pull 필수)
- git clone 주소 폴더이름 (다른 pc에 git 내용 그대로 받기)
 - git pull origin main (git 저장소와 내 PC간 동기화)

수정 후

- git add .
- git commit -m "last"
- git push origin main