

(K-디지털트레이닝) 기업맞춤형 AI-X 융복합 인재 양성과정

- 이 소 영 강사

yisy0703@naver.com

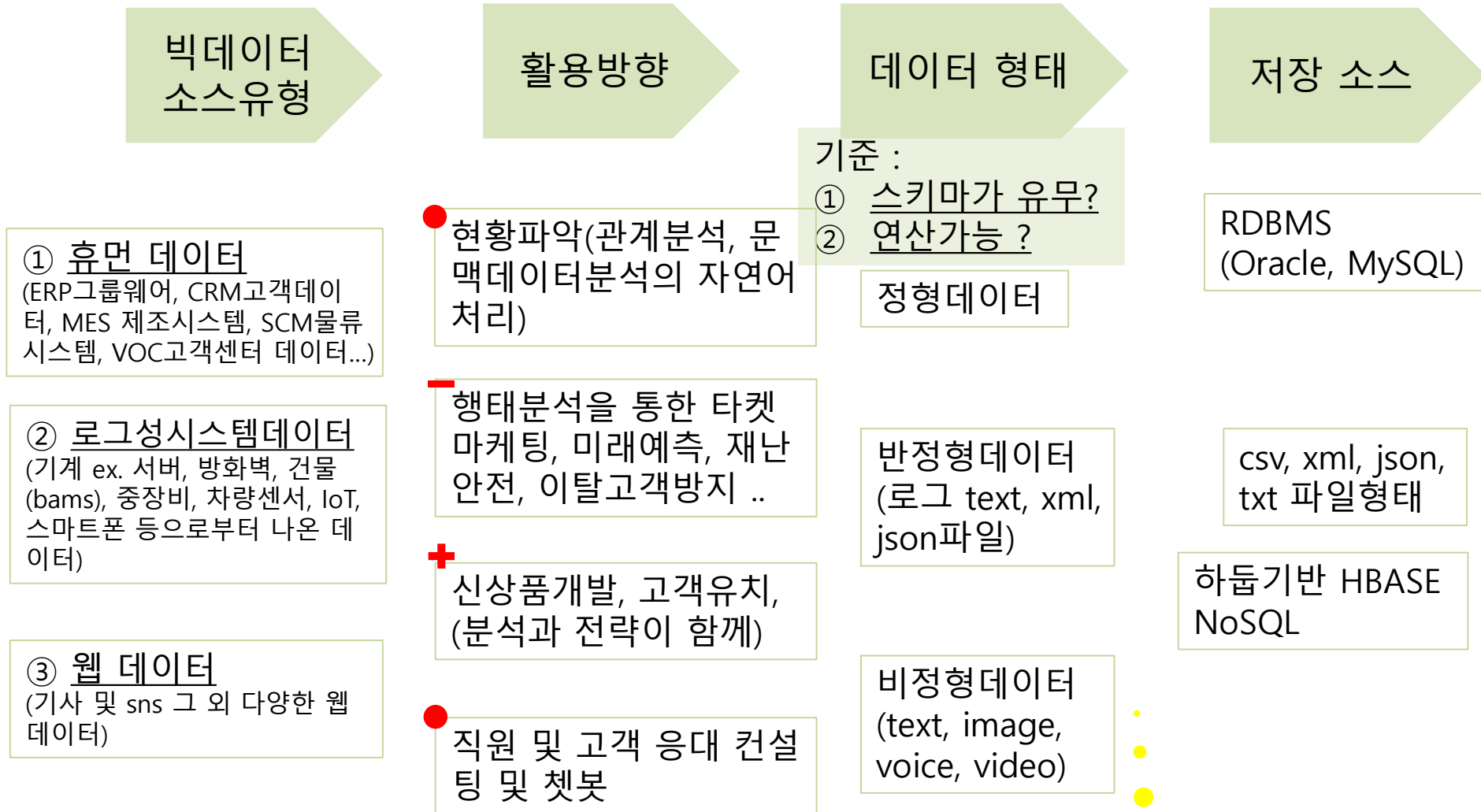
<https://github.com/yisy0/ai> (source, lecNote)

<https://zrr.kr/eTi7Ms> (downloads)

www192.168.0.52

www192.168.0.244

AI 빅데이터의 속성 정리

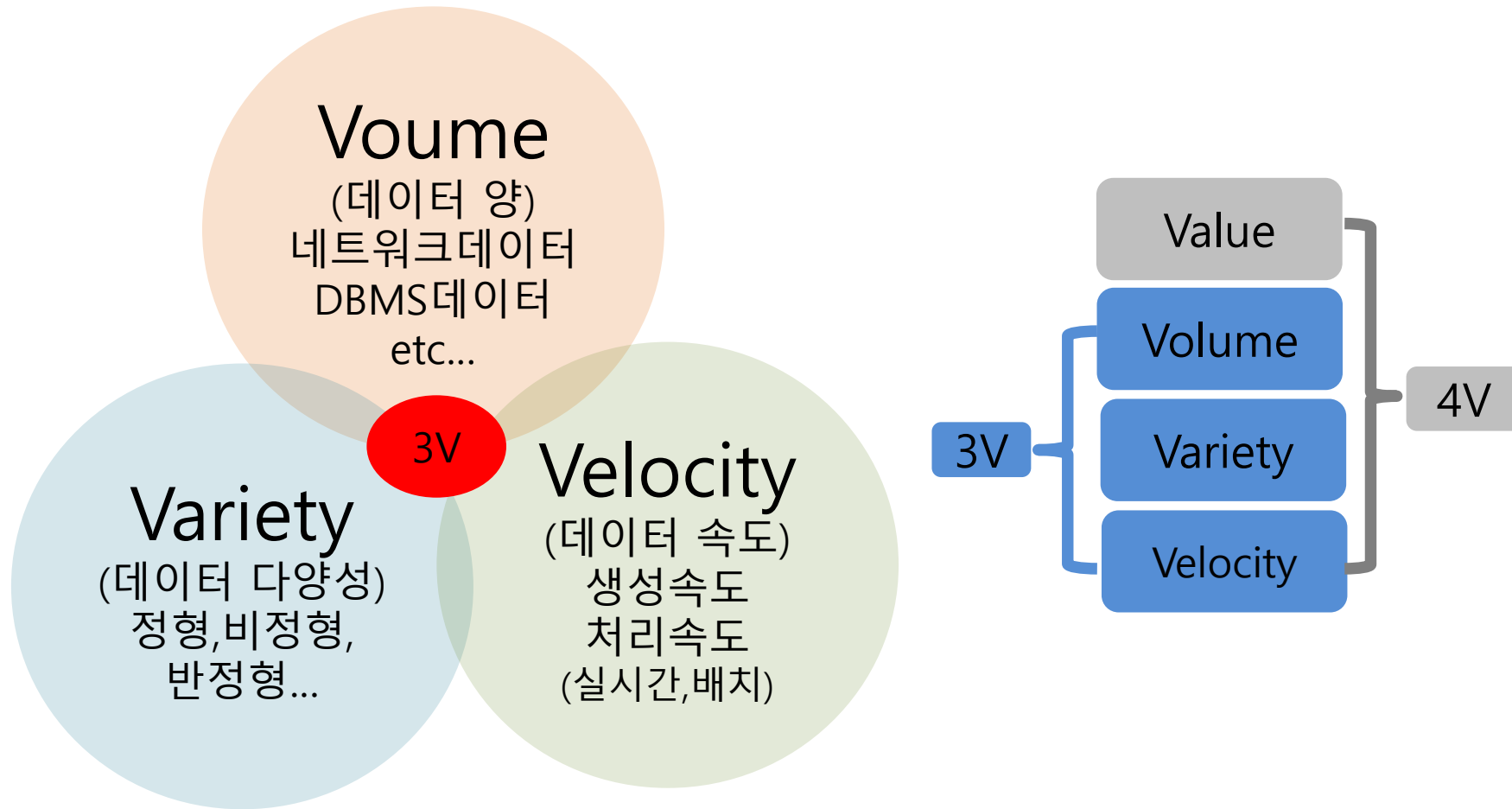


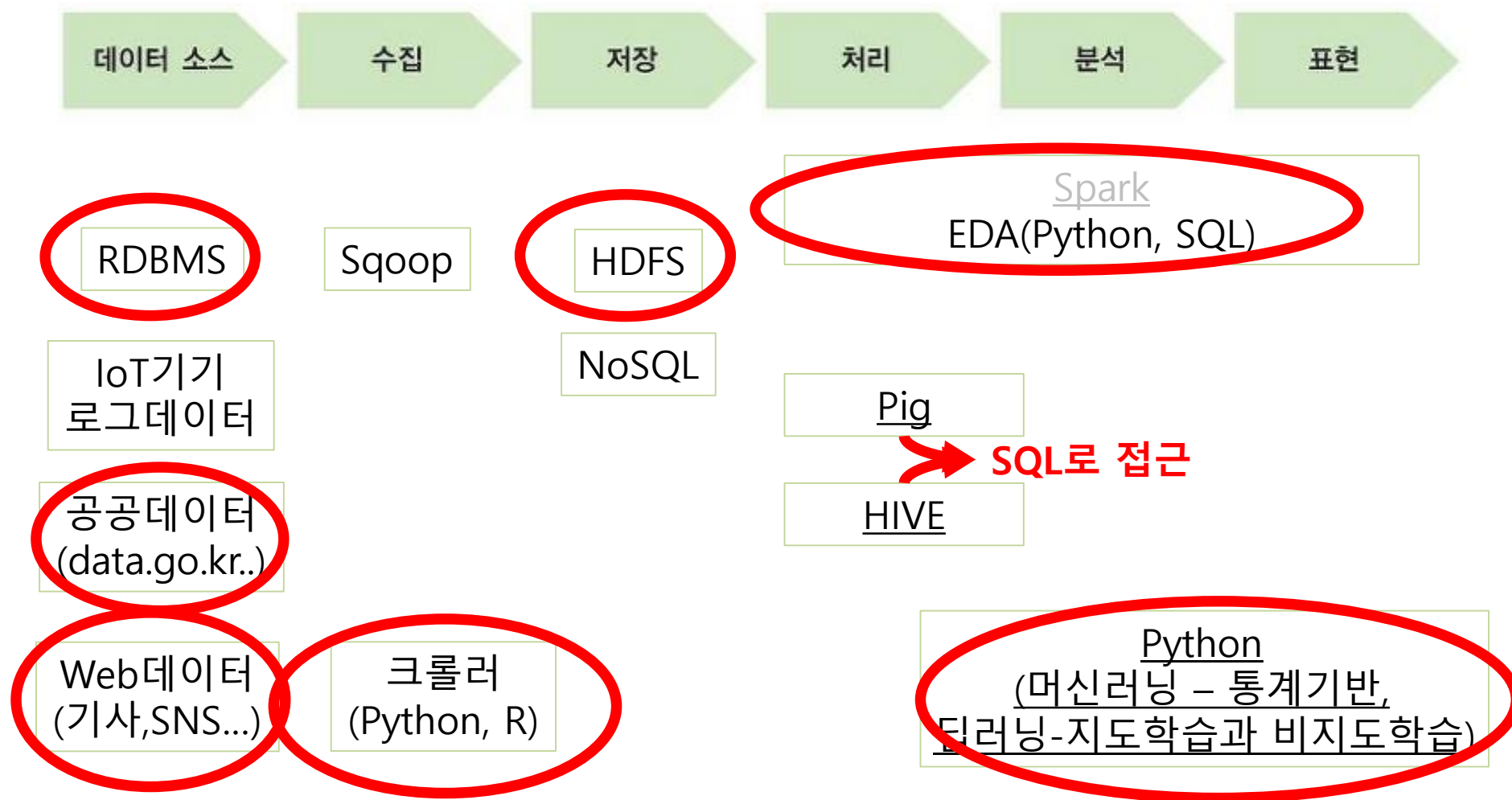
비정형데이터는 정형화 어플리케이션의 성능이 중요(java, R, Python)

빅데이터의 특성

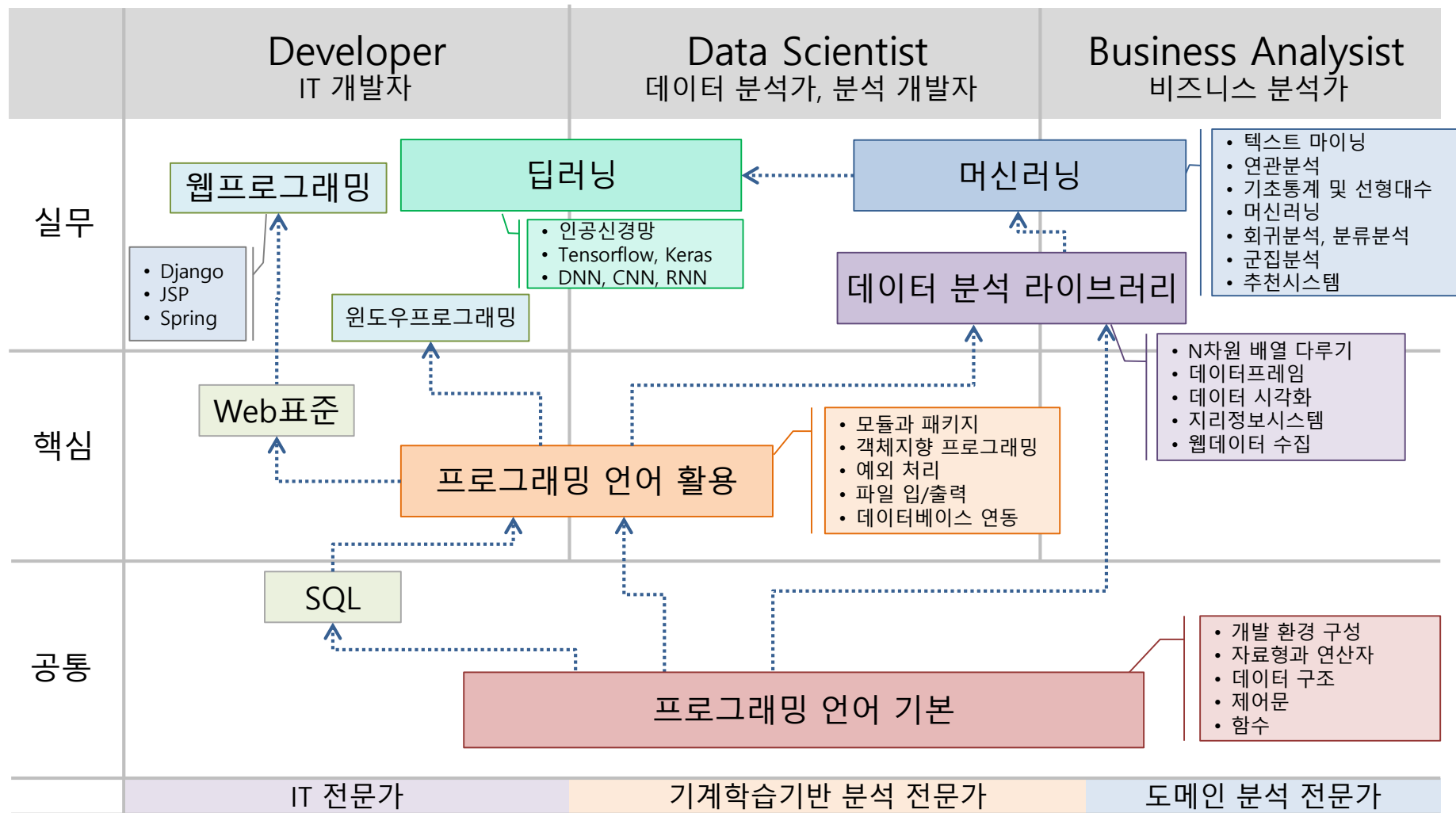
킬로(10^3)-메가(10^6)-기가(10^9)-테라(10^{12})-페타(10^{15})-엑사(10^{18})-제타(10^{21})-요타(10^{24})

- 빅데이터의 특성 : 3V + Value
 - ✓ 초기 가트너에서 정의한 빅데이터의 3V에 가치를 더한 개념





로드맵



일정(824h + 348h)

- ✓ 파이썬 기반 데이터분석 및 시각화 5.5주
- ✓ 웹표준 UI(HTML, CSS, JavaScript, jQuery) 2주
- ✓ 인공지능 모델링(딥러닝과 자연어처리) 1.5주
- ✓ OpenAI API를 활용한 생성형 AI 2주
- ✓ DBMS(Oracle, MySQL) 기본 1주
- ✓ Flask 및 dJango(파이썬 웹 개발 프레임워크) 2주
- ✓ 머신러닝(회귀분석 제외) 1주
- ✓ 프로젝트 (총 3차) 5주
- ✓ 취업지원팀 : 이력서 특강, 취업특강 등

일정(824h + 348h)

- ✓ CNN과 전이학습 1주
- ✓ OpenCV (이미지 전처리를 위한 라이브러리) 1 주
- ✓ YOLO 이미지 머신러닝 1주
- ✓ FastAPI 웹 프레임워크 (DB ORM포함) 1주
- ✓ 프로젝트 5주
- ✓ 취업지원팀 : 이력서 특강, 취업특강 등

이 과정을 마치면...

기본 라이브러리 불러오기

```
In [1]: 1 import tensorflow as tf
2 print(tf.__version__)
3
4 from tensorflow import keras
5
6 import numpy as np
7 import pandas as pd
8 import matplotlib.pyplot as plt
```

1.14.0

💡 딥러닝 라이브러리 Keras를 이용한 분류 예

데이터셋 불러오고 학습 데이터와 검증 데이터로 나누기

```
In [2]: 1 redwine = pd.read_csv("winequality-red.csv", sep=";")
2 redwine_X = redwine.iloc[:, :-1]
3 redwine_y = redwine.iloc[:, -1] - 3
4
5 from sklearn.model_selection import train_test_split
6 train_X, test_X, train_y, test_y = train_test_split(redwine_X,
7 redwine_y,
8 test_size=0.3)
```

인공신경망 모델 정의하기

```
In [3]: 1 model = keras.Sequential([
2     keras.layers.Flatten(input_shape=(11,)),
3     keras.layers.Dense(100, activation='sigmoid'),
4     keras.layers.Dropout(0.2),
5     keras.layers.Dense(200, activation='relu'),
6     keras.layers.Dropout(0.4),
7     keras.layers.Dense(50, activation='tanh'),
8     keras.layers.Dropout(0.1),
9     keras.layers.Dense(6, activation='softmax')
10 ])
11
12 model.compile(optimizer='adam',
13               loss='sparse_categorical_crossentropy',
14               metrics=['accuracy'])
```

손실함수와 옵티마이저 정의하기

```
In [4]: 1 history = model.fit(train_X, train_y.to_numpy(), epochs=300)
```

Epoch 1/300
1119/1119 [=====] - 0s 161us/sample - loss: 1.2863 - acc: 0.4343

학습시키기

평가하기

```
In [5]: 1 test_loss, test_acc = model.evaluate(test_X, test_y, verbose=1)
2 print('\n테스트 정확도:', test_acc)
```

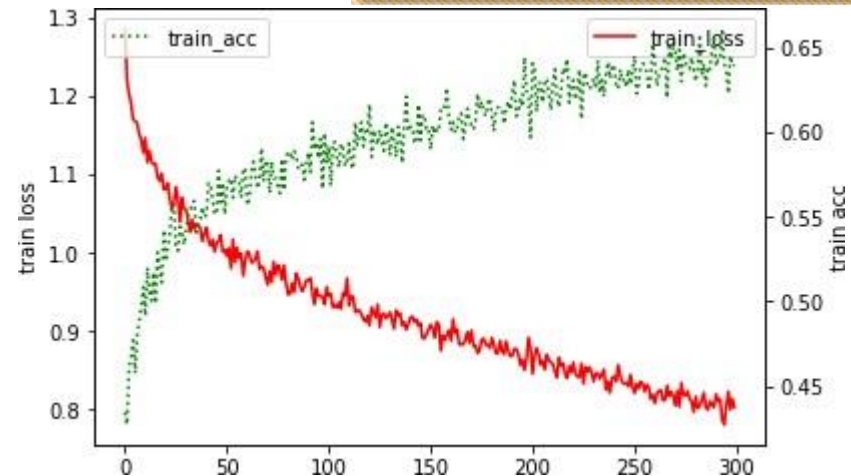
480/480 [=====] - 0s 161us/sample - loss: 1.0884 - acc: 0.6000

테스트 정확도: 0.6

학습 모델 시각화하기

```
In [6]: 1 import matplotlib.pyplot as plt
2 from matplotlib import pyplot as plt
3 plt.rcParams['figure.figsize'] = (6,4)
4 ax1 = plt.subplot(2,1,1)
5 ax1.plot(history.history['loss'], 'r-', label="train_loss")
6 ax1.legend()
7 ax2 = ax1.twinx()
8 ax2.plot(history.history['acc'], 'g:', label="train_acc")
9 ax2.set_ylabel("train acc")
10 ax2.legend()
11 plt.show()
```

학습 시 손실과 정확도를 그래프로 표현



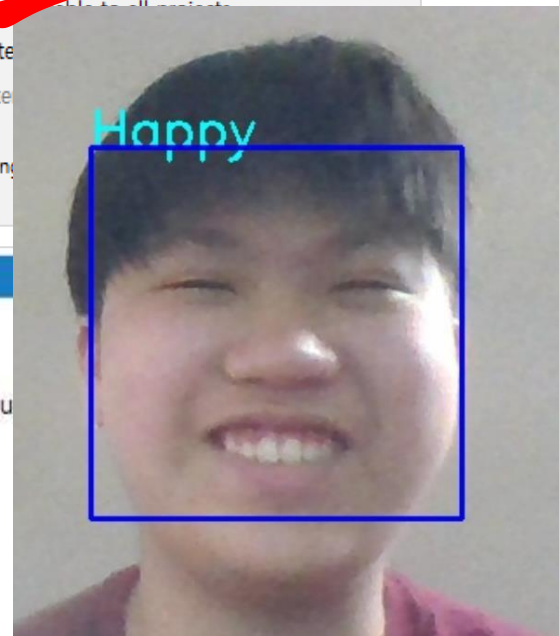
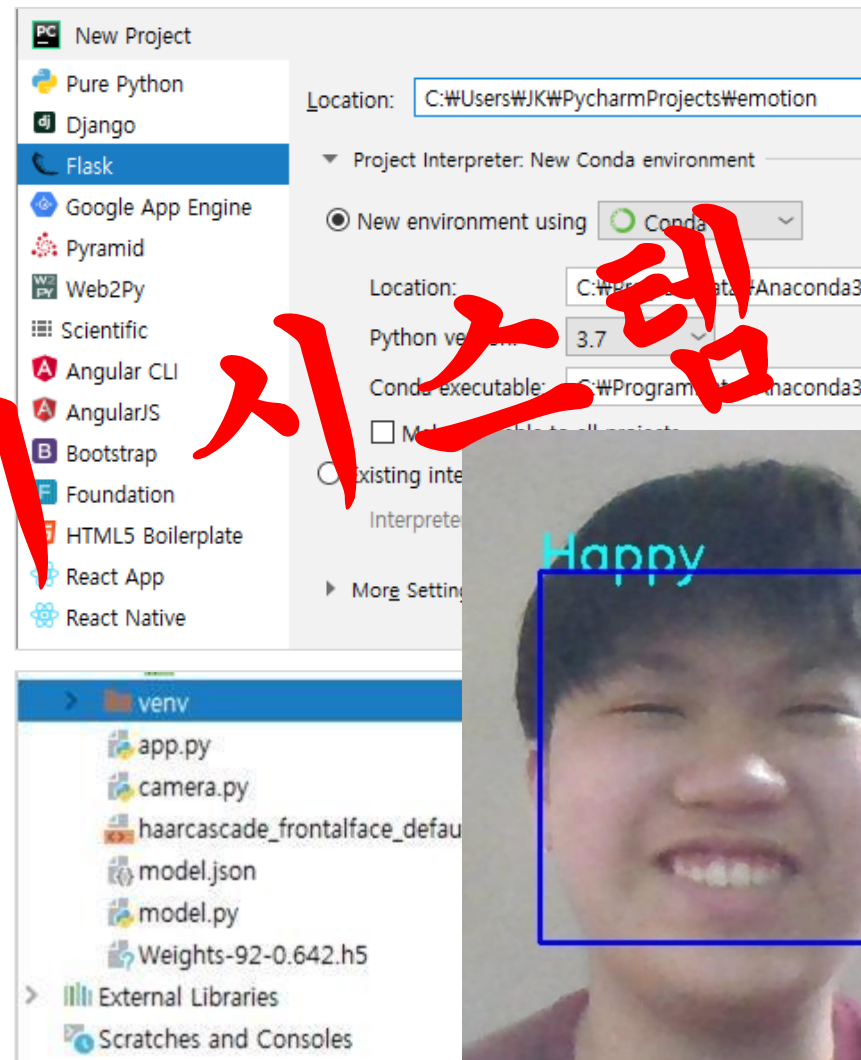
이 과정을 마치면...



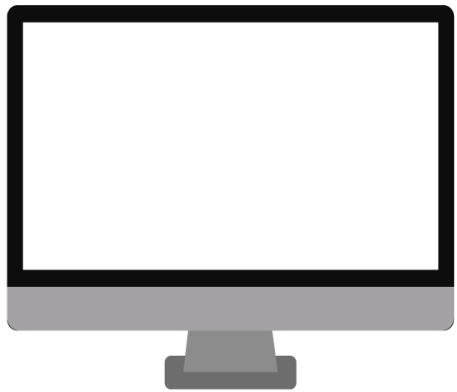
이 과정을 마치면...



시스템
시각인식



컴퓨터 프로그래밍이란?



컴퓨터



C
C++
Java
Python
JavaScript
...
언어



프로그래밍언어를

이용해서

프로그램을

만드는

행위

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



vs.

프로그래밍 언어



저급 프로그래밍 언어와 고급 프로그래밍 언어

● 저급 프로그래밍 언어

- 사람보다 컴퓨터가 이해하기 쉽게 작성된 프로그래밍 언어
- 기계어, 어셈블리어

```
00000000 7f 45 4c 46 01 00 00 00 00 00 00 00 00 00 05
00000010 02 00 03 00 1b 00 00 05 1b 00 00 05 04 00 00
00000020 b2 0c b9 31 00 00 05 cd 80 25 20 00 01 00 40 cd
00000030 80 48 65 6c 6c 6f 20 77 6f 72 6c 64 0a
```

기계어

```
054000000    add    eax, 0x00000004
B20C         mov    dl, 12
B92E00005    mov    ecx, msg
CD80         int    0x80
2520000100   and    eax, 0x00010020
40          inc    eax
CD80         int    0x80
```

어셈블리어

● 고급 프로그래밍 언어

- 사람이 사용하기 편하도록 직관적인 표현으로 작성된 언어
- 대부분 개발자들이 사용하고 있는 언어
- 파이썬, 자바, C언어 등

```
print('Hello World')
```

파이썬

컴퓨터 프로그래밍

```
System.out.println(2);  
System.out.println('Hello');
```

...

컴퓨터 프로그래밍 소스

컴파일
or
인터프리팅



컴파일러 결과
(프로그래밍)

```
0101 0010 1010 1000  
0000 0101 1011 1010
```

...

컴퓨터가 이해할 수 있는 형태

컴파일 언어와 인터프리터 언어

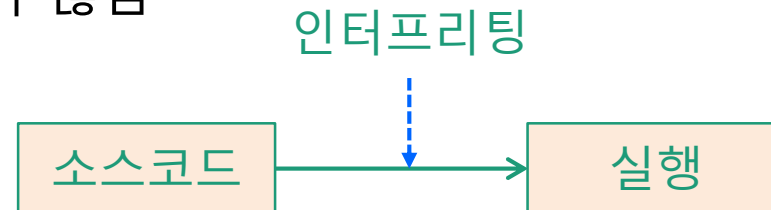
● 컴파일 언어

- 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



어떤 언어를 해야 할까?

- CPU 종속적인 언어

Intel 사 CPU



어셈블리 PG A

AMD사 CPU



어셈블리 PG A (탑재불가)

어떤 언어를 해야 할까?

CPU 독립적인 언어 , OS 종속적인 언어

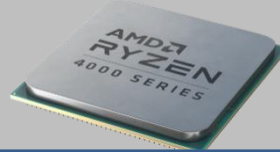
Intel 사 CPU



OS (Windows)

a.c -> a.exe

AMD사 CPU



OS (Windows)

a.c -> a.exe

Intel 사 CPU



OS (MacOS)

a.c -> a.exe 탑재불가

어떤 언어를 해야 할까?

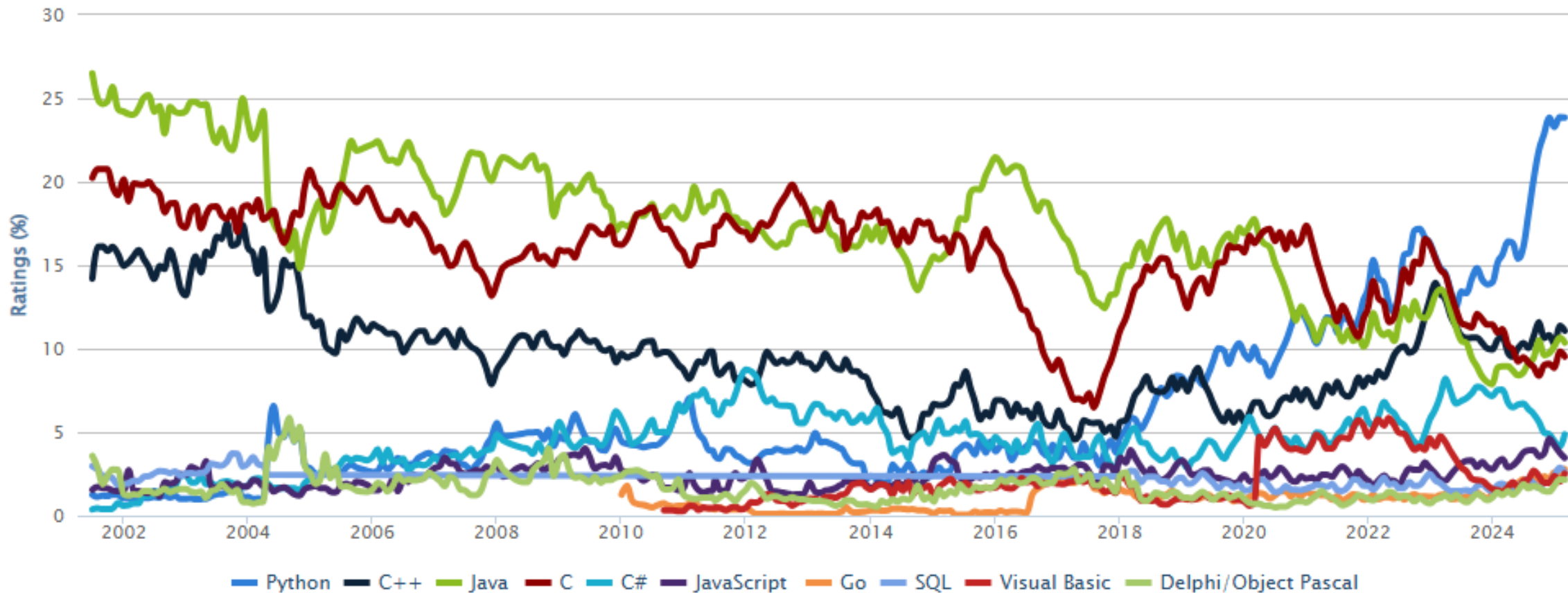
OS 독립적인 언어(=플랫폼 독립적인 언어)



Java 웹 개발자 양성과정

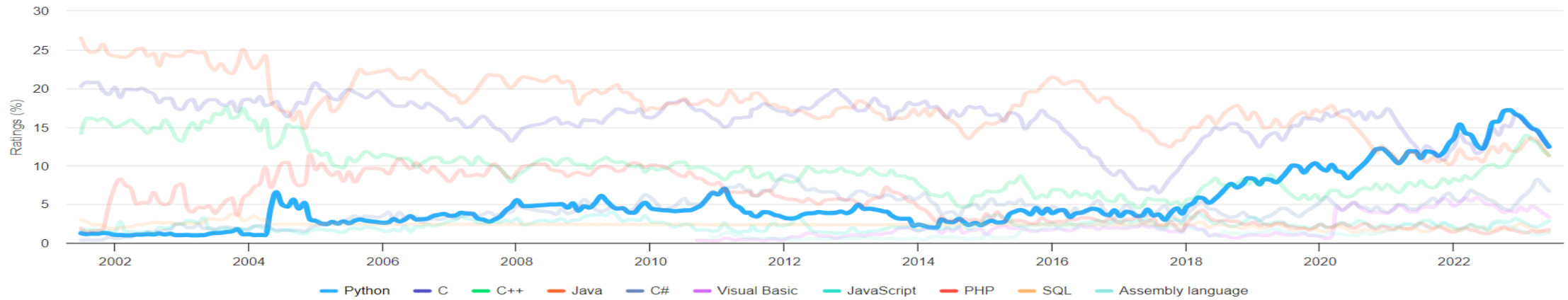
TIOBE Programming Community Index

Source: www.tiobe.com

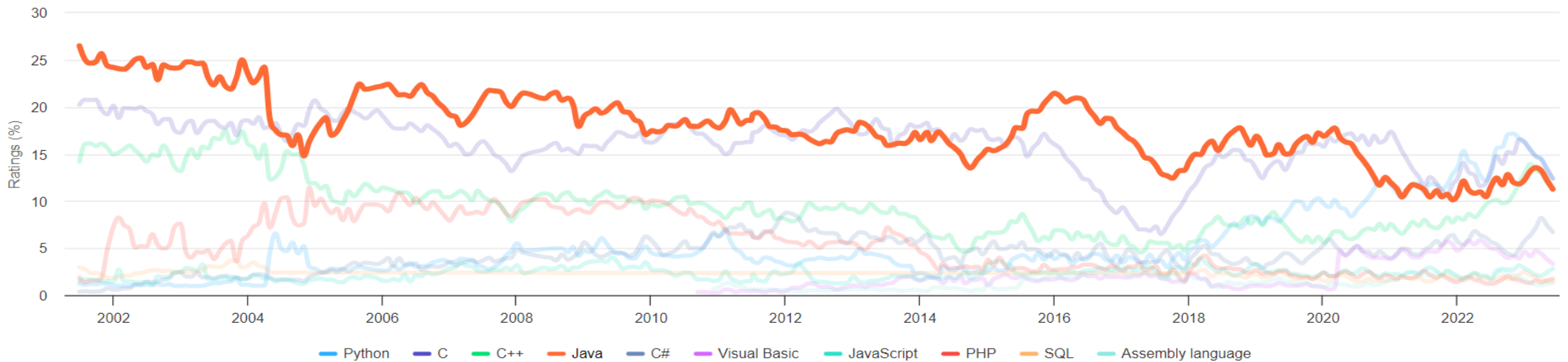


source : <https://tiobe.com/tiobe-index>

Source: www.tiobe.com



Source: www.tiobe.com



초보 개발자는 토끼와 거북이 중 거북이가 되자

시스템 폴더 관리

ai\downloads : 개발환경을 위한 다운로드 파일

ai\source : 수업시간 개발 소스

ai\letNote : 교안 노트

ai\note : 그날 그날 수업한 내용을 10~20줄로 요약한 파일

ai\취업 : 취업에 관련된 정보 및 이력서 포트폴리오

시스템 설치

1. Chrome 브라우저 <https://www.google.com/chrome/browser>
2. Git <https://git-scm.com>
3. 파이썬

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 "yisy0"
 - > git config --global user.email "yisy0703@naver.com"
 - > git config --list (확인)

깃허브의 레파지터리(원격저장소) 생성하고 올리기

6. 내 pc의 폴더에 다음의 파일 추가

① README.md (git 소개)

② .gitignore (git 공간에 올리지 않을 파일과 폴더 리스트) ex (d:/ai-x/.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)

*** 처음 commi의 경우만 인증 로그인 필요 ***

생성된 리파지터리에 데이터 공유하기

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

생성된 리파지터리에 데이터 공유하기

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