

ONLINE LEARNING SYSTEM FOR PYTHON SCRAPPER

– TEAM 6 –

김상민, 김제현, 김호연, 박휘찬, 신동휘, 이지섭,

CONTENTS

- Backend
- Frontend
- IDE
- Deployment

BACKEND – NODE.JS

- Server configured with `express` module
- Communicates with **MySQL DBMS** using `sequelize` module



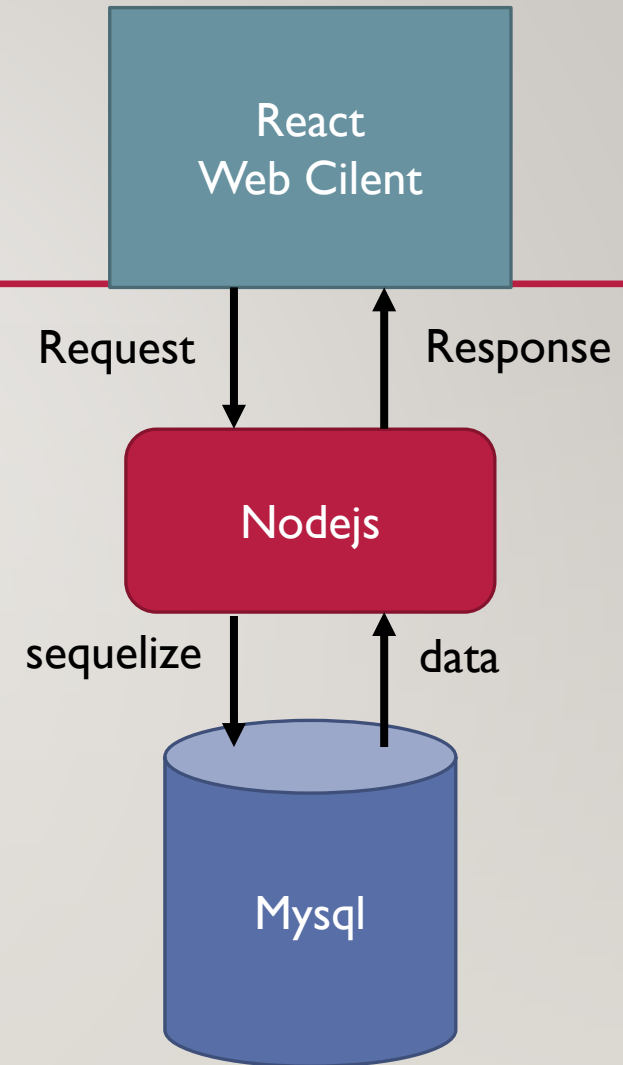
BACKEND – MYSQL

- Database table structure
- User
 - id (PK): bigint, email: string, password: string
- Tutorial
 - id (PK): bigint, chapter: string, title: string, content: string
- Problem
 - id (PK): bigint, chapter: string, problem: string, answer: string
- UserProblemSet
 - id (PK): bigint, user (user.id), problem (problem.id), is_solved: enum ["Y", "N"]



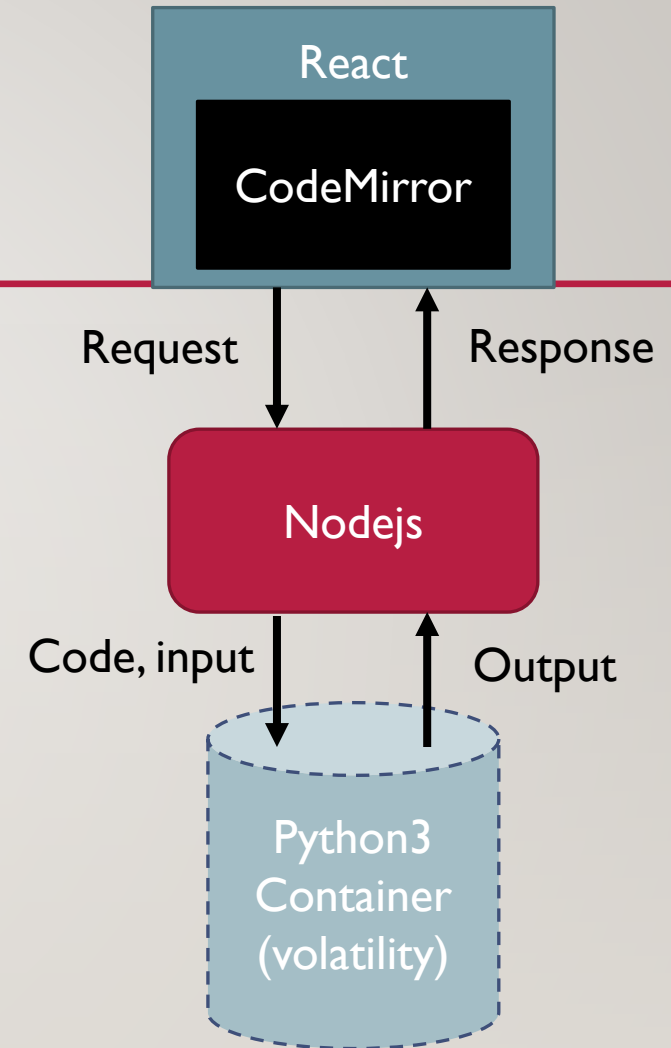
HOW TO COMMUNICATE FRONT & BACKEND

- Backend nodejs express provide api router with DB manager
- Frontend can request **GET** or **POST** method with that url like **/api/auth/login**
- If request income, backend response with some data to frontend client
- This makes **development pallely**



IDE

- **CodeMirror**-based Python IDE
- **Python:3.10 Docker container**
 - User-submitted codes run on this
- If user click run or submit button, code and input will send to server
- **/api/runcode** will execute a code
- Result returned and shown in web page



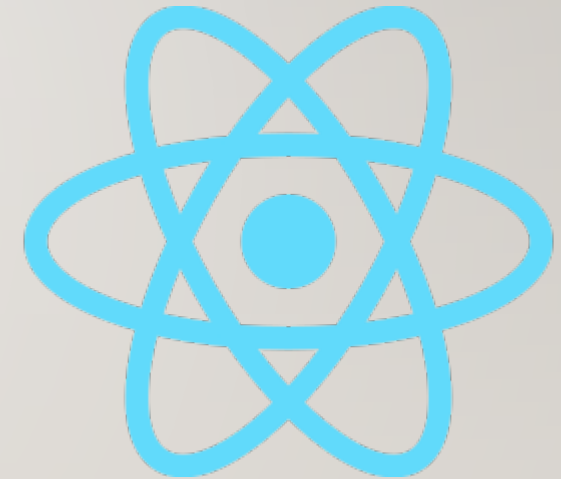
IDE

- (modified) Using nodejs Python-shell for code executing
 - ➔ secure issue, server attack, access to other user's code...
 - ➔ Virtualizing with Docker container
- (added) Timeout 10 sec, Long output denied
- (fixed) Post request to server with code
- (fixed) Code-Mirror IDE design
- (fixed) run with input, grading correct answers
- (fixed) Code save and load for user

FRONTEND



- Based on **Next.js** and **React**
- Provides web page component structures
 - Such as log-in page and problem pages
- Communicates with backend to fetch contents with **API**



로그인

Email

Password

회원가입

로그인

회원가입

Email

Email address
kijen98@gmail.com

Email Confirm

Email address
kijen98@gmail.com

Password

Password
.....

Password Confirm

Retype Password
.....

뒤로가기

회원가입

localhost:3060 내용:

ERROR!

확인

로그인

Email

Email address
kijen98@gmail.com

Password

Password
.....

회원가입

로그인

컨텐츠

문제풀기

1-1

Chapter 1-1

Chapter 1-1 자료형 인트로

모든 컴퓨터는 정보를 1과 0으로 이루어진 2진수로 처리합니다. 그 중에서 특히 문자는 2진수로 바뀌어 ASCII, UTF-8등 여러 포맷을 통해 처리하게 됩니다. 예를 들어 ASCII에서 알파벳 A는 숫자 65로 표현하게 됩니다. 그렇다면 컴퓨터는 이진수로 바뀐 65라는 숫자가 알파벳 A를 나타내는지 단순히 숫자 65를 나타내는지 어떻게 알 수 있을까요? 이를 데이터에 표시해 준 것을 자료형이라고 합니다.

자료형은 컴퓨터가 여러 종류의 데이터를 식별하는 분류입니다. 보통 python의 기초를 다룰 때 숫자형, 문자열, 리스트, 튜플, 딕셔너리, 집합, 불을 배웁니다. 이번 챕터에서는 이들이 무엇이고, 다루기 위한 함수들을 배울 것입니다.

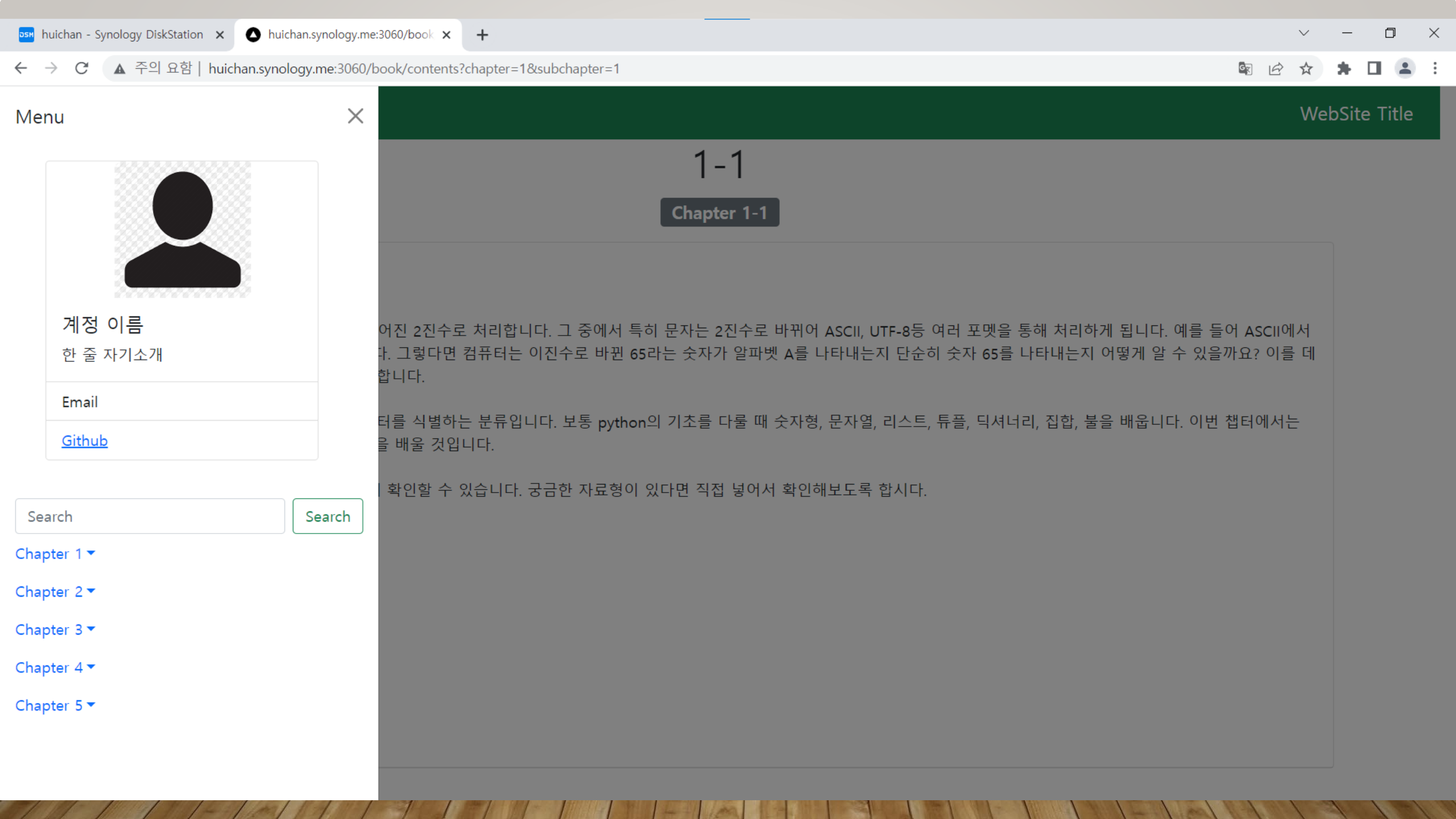
Type()함수를 이용하여 자료형을 쉽게 확인할 수 있습니다. 궁금한 자료형이 있다면 직접 넣어서 확인해보도록 합니다.

```
>>> type(6)           # 정수
```


```
<type 'int'>
```

```
>>> type('A')         # 문자열
```

```
<type 'str'>
```

Menu ✕



계정 이름

한 줄 자기소개

Email

[Github](#)

Search

Search

- Chapter 1 ▾
- Chapter 2 ▾
- Chapter 3 ▾
- Chapter 4 ▾
- Chapter 5 ▾

1-1

Chapter 1-1

어진 2진수로 처리합니다. 그 중에서 특히 문자는 2진수로 바뀌어 ASCII, UTF-8등 여러 포맷을 통해 처리하게 됩니다. 예를 들어 ASCII에서 다. 그렇다면 컴퓨터는 이진수로 바뀐 65라는 숫자가 알파벳 A를 나타내는지 단순히 숫자 65를 나타내는지 어떻게 알 수 있을까요? 이를 데 합니다.

터를 식별하는 분류입니다. 보통 python의 기초를 다룰 때 숫자형, 문자열, 리스트, 튜플, 딕셔너리, 집합, 불을 배웁니다. 이번 챕터에서는 을 배울 것입니다.

확인할 수 있습니다. 궁금한 자료형이 있다면 직접 넣어서 확인해보도록 합시다.

문제 풀기

챕터 1

챕터 2

챕터 3

챕터 4

Chapter 1 문제 풀기

1. 1-1	Solved
2. 1-2	UnSolved
3. 1-3	UnSolved
4. 1-4	UnSolved
5. 1-5	UnSolved
6. 1-6	UnSolved
7. 1-7	UnSolved
8. 1-8	UnSolved
9. 1-9	UnSolved
10. 1-10	UnSolved
11. 1-11	UnSolved
12. 1-12	UnSolved
13. 1-13	UnSolved
14. 1-14	UnSolved
15. 1-15	UnSolved
16. 1-16	UnSolved
17. 1-17	UnSolved
18. 1-18	UnSolved
19. 1-19	UnSolved

2 번 문제 풀기

챕터 1-2 2번 문제

유클리드 호제법이라는 방법을 통해 두 정수의 최대 공약수를 찾을 수 있습니다. 큰 수를 작은 수로 나누어서 나온 나머지로 다시 작은 수를 나누는 과정을 나머지가 0이 나올 때까지 반복하면, 직전의 작은 수가 최대공약수가 됩니다. 예시로 126과 75의 최대공약수를 구해 보면, 3이 나옵니다:

$126 \% 75 \Rightarrow 51$

$75 \% 51 \Rightarrow 24$

$51 \% 24 \Rightarrow 3$

$24 \% 3 \Rightarrow 0$

이렇게 유클리드 호제법을 이용해서 354748과 491512의 최대공약수를 구하는 코드를 작성해 보세요.

사용을 추천하는 함수: `max()`, `min()`

1

Please edit your input

RUN

SUBMIT

2 번 문제 풀기

챕터 1-2 2번 문제

유클리드 호제법이라는 방법을 통해 두 정수의 최대 공약수를 찾을 수 있습니다. 큰 수를 작은 수로 나누어서 나온 나머지로 다시 작은 수를 나누는 과정을 나머지가 0이 나올 때까지 반복하면, 직전의 작은 수가 최대공약수가 됩니다. 예시로 126과 75의 최대공약수를 구해 보면, 3이 나옵니다:

126 % 75 => 51

75 % 51 => 24

51 % 24 => 3

24 % 3 => 0

이렇게 유클리드 호제법을 이용해서 354748과 491512의 최대공약수를 구하는 코드를 작성해 보세요.

사용을 추천하는 함수: max(), min()

```
1 def getGCF(a, b):
2     if a < b:
3         tmp = a
4         a = b
5         b = tmp
6
7     while b != 0:
8         n = a % b
9         a = b
10        b = n
11    return a
12
13 a = int(input())
14 b = int(input())
15 print(getGCF(a, b))
```

126

75

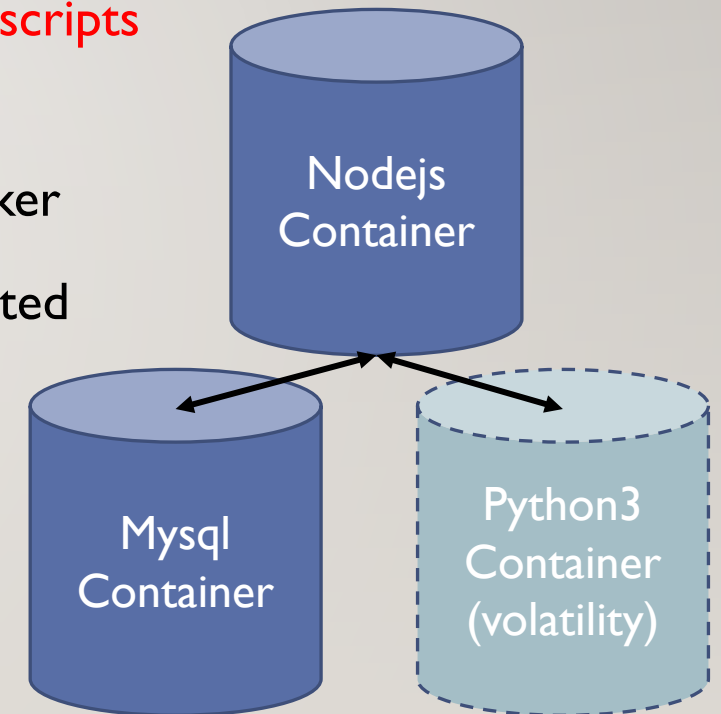
3

RUN

SUBMIT

CONTAINER DEPLOYMENT

- Dependency and Docker containers maintained by two **shell scripts** *run.sh* and *stop.sh*
- node:18, mysql:8, and python:3.10 **container** executed by docker
- Python container is **volatile** alive, when code should be executed



SERVER ENVIRONMENT

- Deployed and tested on a personal **NAS-Synology**

