



스파르타코딩클럽 7주차



매 주차 강의자료 시작에 PDF파일과 영상 링크를 올려두었어요!

▼ PDF 강의자료 다운받기

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5da717b3-1c8a-43fd-8f77-bd9002772b44/week07.pdf>

▼ 영상강의 참고하기

- 곧 업로드 될 예정입니다!



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **option** + **t**

목차



수업 목표



체크인



배우고 적용하기 - 전반부



5분 꿀팁 - 가르치면서 배우기



웹 기초 동작 원리 - 총 복습

[2시간] 수업

[1시간] 프로젝트 코딩 & Todo 리스트 점검



배우고 적용하기 - 후반부

[2시간]: 프로젝트 개발

[1시간] 튜터 피드백 - Todo 리스트

[숙제 - 단짠단짠 🍷🍰]

숙제 1. 개발일지 작성

숙제 2. 내 프로젝트 자랑하기



체크아웃

[설치] - 다음 시간을 위해 미리 설치해와야 할 것들



수업 목표

1. 내 서버에 mongoDB를 설치하기
2. 서버에서 나홀로 메모장 실행하기
3. 진짜 웹 서비스처럼 실행하기
 - 1) SSH 접속을 끊어도 서버가 계속 프로젝트를 실행할 수 있게 하기
 - 2) IP 주소 뒤에 붙는 포트 번호(5000)을 없애기

전반 3시간



체크인



튜터님은 체크인과 함께 **출석 체크(링크)**를 진행해주세요!

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

▼ "15초 체크인"을 진행합니다.

- 튜터님은 타이머를 띄워주세요! (링크)
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.
 - 예. 지난주에 등산을 했었는데요. 갑자기 아, 가보고 싶은 산들을 모아놓는 나만의 메모장을 만들고 싶더라고요! 제 프로젝트로 그거 해보려고요!



배우고 적용하기 - 전반부



5분 꿀팁 - 가르치면서 배우기

▼ 가르치면서 배우기 Learning By Teaching

- 앞으로 스스로 공부를 위해 Learning By Teaching! 바로 가르치면서 배우기 학습방법을 소개하려고 합니다.
- 시험기간에 친구들을 가르쳐주면서 나도 자연스럽게 학습이 되었던 기억 한 번 쯤은 있으셨을거예요. 다른 사람에게 가르치면서 나만의 언어로 정리가 되고, 내가 모르는 부분을 깨닫게 되죠.
- ▼ 참고. Learning By Teaching 이 왜 효과적인지(교육 심리학 연구 요약 / 영문)

Learning by teaching others is extremely effective - a new study tested a key reason why

By Christian Jarrett The learning-by-teaching effect has been demonstrated in many studies. Students who spend time teaching what they've learned go on to show better understanding and knowledge retention than students who simply spend the same time re-studying. What remains unresolved, however,

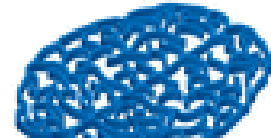
<https://digest.bps.org.uk/2018/05/04/learning-by-teaching-others-is-extremely-effective-a-new-study-tested-a-key-reason-why/>



The learning benefits of teaching: A retrieval practice hypothesis

Teaching educational materials to others enhances the teacher's own learning of those to-be-taught materials, although the underlying mechanisms remain largely unknown. Here, we show that the learnin...

<https://onlinelibrary.wiley.com/doi/abs/10.1002/acp.3410?campaign=wolearlyview>



- 특히 이 과정에서 **내가 무엇을 모르는지** 깨닫는 게 된답니다. 내용은 몰라도 괜찮아요. 뭘 모르는지 알게 되면, 이제 알아보면 되니까요 😊
- 웹을 관통하는 개념! **웹 기초 동작 원리**를 20분 안에 무려 4번 복습할 수 있는 꿀팁을 알려드릴게요! 스터디에서도 응용할 수 있겠죠?
 1. 튜터님이 설명하는 웹 기초 동작원리를 열심히 듣는다.
 2. 빈 종이에 웹 동작원리를 스스로 그려본다.
 3. 옆 짝꿍 튜티에게 서로 웹 동작원리를 설명해준다.
 4. 칠판에 나와서 웹 동작원리를 그리면서 동료 튜티에게 설명해준다.

[한 걸음 더]

- 앞서 배운 웹 동작원리 복습 꿀팁 을 실제로 해본다!

📖 웹 기초 동작 원리 - 총 복습

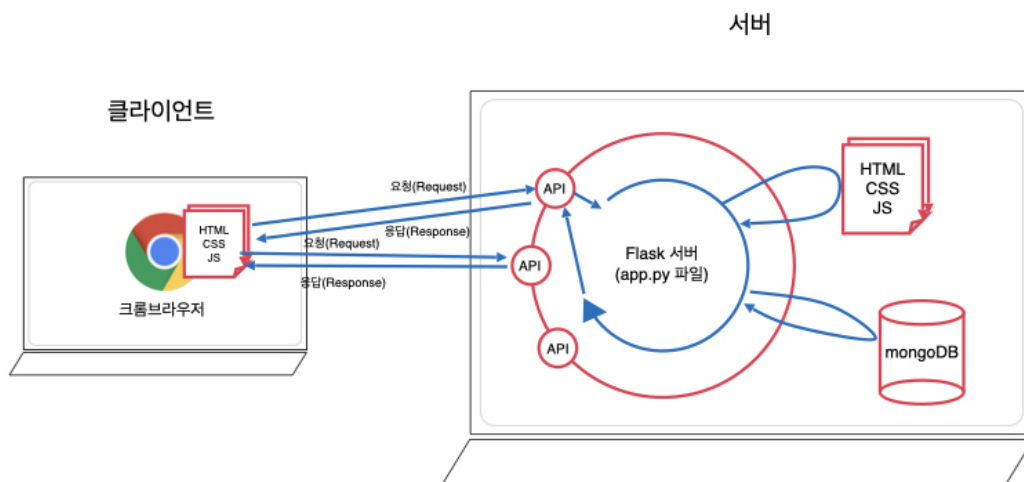
▼ 총 복습 - HTTP 요청(GET/ POST), HTTP 응답, API, Flask

- HTTP: 웹은 HTTP라는 규약(규칙)을 따릅니다. url에서 **http://** 가 바로 HTTP 라는 규약을 따른다는 표시예요.
- 클라이언트 : HTTP에서 요청을 하는 쪽

- 서버 : HTTP에서 요청을 받아 응답하는 쪽입니다. 응답하는 데이터로는 화면 코드(HTML, CSS, JS) 와 JSON같은 데이터 파일 등이 있습니다.
- 클라이언트 요청 방식 - GET, POST
 - GET → 통상적으로! 데이터 조회(Read)를 요청할 때
예) 영화 목록 조회
→ 데이터 전달 : URL 뒤에 물음표를 붙여 key=value로 전달
→ 예: google.com?q=북극곰
 - POST → 통상적으로! 데이터 생성(Create), 변경(Update), 삭제(Delete) 요청 할 때
예) 회원가입, 회원탈퇴, 비밀번호 수정
→ 데이터 전달 : 바로 보이지 않는 HTML body에 key:value 형태로 전달
- API(Application Programming Interface): 응용 프로그램(application, 애플리케이션)에서 기능을 사용하거나 데이터를 주고 받기 위한 기능
 - API 를 사용할 땐, 미리 정해진 약속 을 따라야 작동합니다. 약속들은 API 페이지(문서)에 적혀있습니다.
 - 우리가 사용하는 API는 아래 적힌 모든 것을 미리 약속해두고 그대로 동작합니다.
 1. 요청 정보 : 요청 URL, 요청 방식 (GET / POST /...)
 2. 서버가 제공할 기능 : 데이터 조회(Read), 데이터 생성(Create) 등
 3. 응답 데이터 : 응답 데이터 형식 (어떤 key 로 어떤 데이터를 줄지, 예. response['img'])
- Flask : 웹을 만들고 서버를 구동시키기 편하게 하는 프레임워크(Framework)(Flask 공식 문서 / 비공식 한글 번역문서)
 - `@app.route('/')` 부분을 수정해서 URL을 부여할 수 있습니다
 - templates 폴더 : HTML 파일을 담아둡니다. 실행할 때에는 이 폴더에서 화면을 불러오죠.
 - static 폴더 : 이미지나 css파일과 같은 정적 파일을 담아두는 역할을 하지요!



서버와 클라이언트가 하나의 컴퓨터에 있던 로컬 개발 환경에서 한 걸음 더 나아가
오늘은 클라우드에서 구입한 서버에서 mongoDB 를 설치하고, Flask 프로젝트를 실행시켜 보겠습니다.



[2시간] 수업

- ▼ 1) 7주차 수업 설명: "웹서비스를 런칭하기 위한 작업들"
- 6주차: 서버 구입해서 프로젝트 실행시켜보기
 - 클라우드 서비스 AWS에서 사용할 서버(EC2) 구매하기
 - 구입한 서버에서 간단한 flask 서버 실행해보기
- 7주차: 클라우드 서버에 mongoDB 설치하기 + 나홀로 메모장 완성본을 서버에서 실행하기

- 8주차: 도메인 붙이기

- 숫자로 된 IP 주소 대신 영문으로 된 주소 (www.spartacodingclub.co.kr 처럼요!) 로 접속할 수 있게 됩니다.

▼ 2) EC2에 접속하기

▼ Mac OS: Mac은 ssh가 있어서, 명령어로 바로 접근 가능!

- 터미널을 열기 (spotlight에 terminal 입력)
- 방금 받은 내 Keypair의 접근 권한을 바꿔주기

```
sudo chmod 400 받은키페어를꺼다놓기
```

- SSH로 접속하기

```
ssh -i 받은키페어를꺼다놓기 ubuntu@AWS에적힌내아이피
```

예) 아래와 비슷한 생김새!

```
ssh -i /path/my-key-pair.pem ubuntu@13.125.250.20
```

▼ Window: ssh가 없으므로, git bash라는 프로그램을 이용!

- gitbash를 실행하고, 아래를 입력!

```
ssh -i 받은키페어를꺼다놓기 ubuntu@AWS에적힌내아이피
```

예) 아래와 비슷한 생김새!

```
ssh -i /path/my-key-pair.pem ubuntu@13.125.250.20
```

- Key fingerprint 관련 메시지가 나올 경우 Yes를 입력해주세요!
- git bash를 종료할 때는 exit 명령어를 입력하여 ssh 접속을 먼저 끊어주세요.

▼ 3) 나홀로메모장 실행 준비 - mongoDB를 설치



mongoDB를 설치하고,
→ 실행해보고,
→ 계정(접속 아이디/비밀번호)을 만들고,
→ 외부에 열어놓는 것까지 해볼게요!

- 시작 전, 오늘 실행할 파일을 저장하는 폴더를 만들어볼게요. 바탕화면 - sparta안에 week07 폴더를 만들어주세요.

▼ 1. mongoDB 설치하기

- 아래 파일을 week07 폴더에 다운받습니다. 그런 후 파일질라에서 드래그 드롭으로 EC2 인스턴스의 home/ubuntu 폴더에 업로드합니다.

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/88798a9d-e694-4420-811e-066ef30a3a8/installmongodb.sh
```

- 위 파일은 쉘 명령어를 적어둔 파일입니다. 파일을 실행시키면 안에 적어둔 명령어들이 실행됩니다. 위 파일에는 mongoDB를 설치하는 명령어들을 모아두었습니다.



위 파일 내용은 mongoDB 공식자료를 참고해서 정리했습니다. (링크)
외울 필요는 전혀 없습니다! 사용 설명서 보듯이 자료를 찾아서 그대로 실행하면 됩니다.

- EC2 에 접속한 터미널(윈도우즈는 GitBash) 에 아래와 같이 입력합니다. 쉘에서는

`installmongodb.sh` 파일이 있는 `ubuntu` 폴더에 위치하고 있어야합니다.

```
. installmongodb.sh
```

▼ 2. mongoDB 실행하기

- EC2 에 접속한 터미널(윈도우즈는 GitBash) 에 아래와 같이 입력합니다.

```
# 실행. 아무 반응이 없으면, 잘 실행된 것!
# 리눅스는 보통 잘 되면 아무것도 안나와요!^^;
sudo service mongod start

# 아래 명령어를 입력해서 127.0.0.1:27017 이 있는지 확인하기
# 있다면 mongoDB가 정상 작동 중이라는 뜻입니다.
netstat -tnlp
```

▼ 3. mongoDB 접속 계정 생성하기



우리가 만든 mongoDB를 외부 접속할 수 있게 하기 위해,
접속에 필요한 아이디와 비밀번호를 세팅해봅시다!
설정을 하지 않으면 누구나 DB정보를 볼 수 있게 됩니다.

```
# mongoDB 쉘에 들어가기
mongo
```



좌측에 '>' 표시가 나오면 성공적으로 MongoDB에 접속한 것입니다! 다음 명령어를 순차적으로 입력해주세요.

눈치채셨겠지만,
`test, test` 자리에 내가 넣고 싶은 아이디/비밀번호를 넣으면 됩니다. (영어로만요!)

```
# admin으로 계정 바꾸기
use admin;

# 계정 생성하기
db.createUser({user: "test", pwd: "test", roles:["root"]});
```



아래와 같은 화면을 보면 완성!

```
> use admin;
switched to db admin
> db.createUser({user: "test", pwd: "test", roles:["root"]});
Successfully added user: { "user" : "test", "roles" : [ "root" ] }
```

```
# mongoDB 쉘에서 나오기
exit

# MongoDB 재시작
sudo service mongod restart
```

▼ 4. mongoDB를 외부에 열어주기 - mongoDB 설정 업데이트



mongoDB는 기본적으로 같은 IP 안에서만 접속을 허용하고 있습니다.
앞으로 할 것은 외부에서 접근이 가능하도록 잠금을 풀어주는 작업입니다.

(1) 아래 mongoDB의 설정파일을 **week07** 폴더에 다운받습니다. 그런 후 파일질라에서 드래그 드롭으로 EC2 인스턴스의 /etc 폴더에 업로드합니다.

- mongoDB는 이 이름이어야 설정파일로 인식하므로, 파일이름을 바꾸지 않아야합니다. EC2 인스턴스에는 이미 같은 파일 이름이 있으므로 **overwrite(덮어쓰기)** 해줍니다.

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/51f56f28-2a6d-4ddb-9d10-3783281051c3/mongod.conf>

▼ 기본 설정파일과 다르게 바뀐 내용은 아래와 같습니다.

- 0.0.0.0 : 모든 IP 에서 접속할 수 있게 해주고,
- 인증(authorization)을 활성화시킵니다.

```
# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0

# how the process runs
processManagement:
  timeZoneInfo: /usr/share/zoneinfo

security:
  authorization: enabled
```

(2) 바뀐 설정을 적용하기 위해 mongoDB를 재시작 시키고, 제대로 수정되었는지 확인합니다.

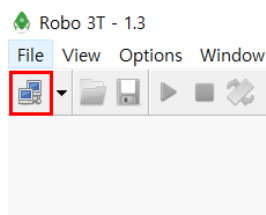
```
# mongoDB 재시작
sudo service mongod restart

# 아래 명령어를 입력하여 0.0.0.0:27017 이 목록에 있음을 확인합니다.
# 만약 127.0.0.1:27017 이 있다면 설정 파일이 제대로 수정되지 않은 것입니다.
netstat -tnlp
```

▼ 5. Robo3T를 이용해서, 내 컴퓨터에서 EC2서버에 있는 mongoDB에 원격으로 접속하기

▼ (1) id/pw를 넣고 접속해보기

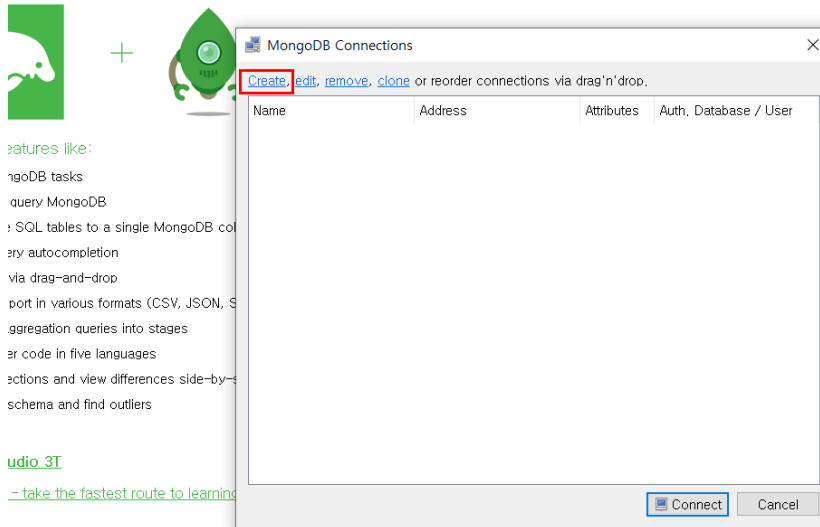
- 좌측 상단 빨간 상자 내 아이콘을 클릭합니다.



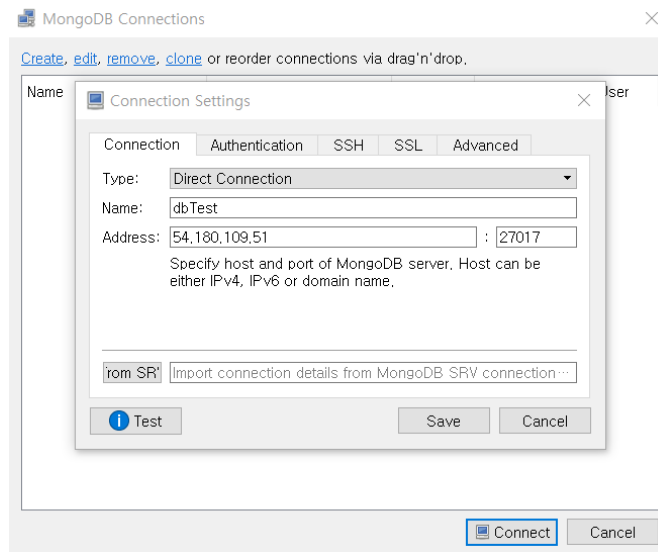
- Create 클릭!

more powerful GUI? Try Robo 3T's sibling, Studio 3T

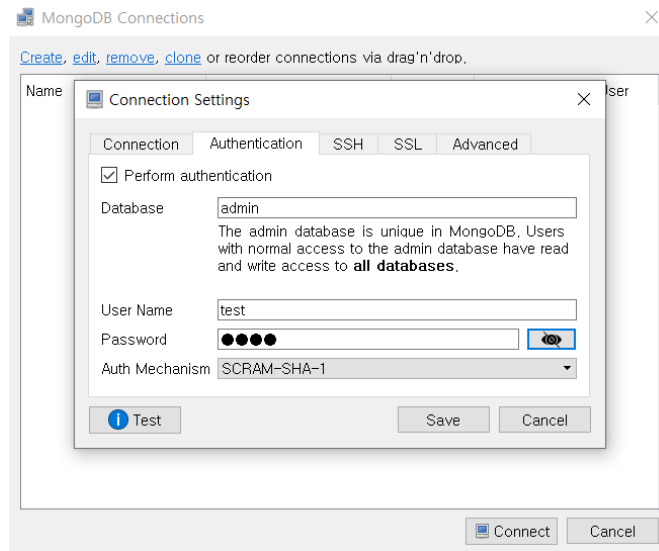
Blog Posts



- 접속 정보를 세팅합니다.



- 상단 Authentication 탭을 클릭합니다.
 1. Perform authentication 체크박스를 클릭합니다.
 2. 생성한 계정의 아이디와 비밀번호를 입력하고, 'save'를 클릭합니다.



▼ (2) 만약에 안된다면 다음 사항을 확인해보세요.

1. AWS에 접속합니다.

```
ssh -i 키페어 ubuntu@내AWS아이피
```

2. MongoDB 상태 확인

```
netstat -tnlp
```

아래처럼 "127.0.0.1:27017" 을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000           0.0.0.0:*               LISTEN      13341/python
tcp        0      0 127.0.0.1:27017        0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
```

3. 설정 확인

- 위 4. mongoDB를 외부에 열어주기 - mongoDB 설정 업데이트 에서 설정이 변경되었는지 다시 한 번 확인합니다.

4. 재시작

```
sudo service mongod stop
```

```
netstat -tnlp
```

위 명령어를 차례대로 입력하여 27017 포트에 아무것도 떠있지 않음을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ sudo service mongod stop
ubuntu@ip-172-31-34-31:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000           0.0.0.0:*               LISTEN      13341/python
tcp6       0      0 :::22                  :::*                    LISTEN      -
```



```
sudo service mongod start

netstat -tnlp
```

위 명령어를 차례대로 입력하여 0.0.0.0:27017 이 목록에 있음을 확인합니다.

```
ubuntu@ip-172-31-34-31:~$ sudo service mongod start
ubuntu@ip-172-31-34-31:~$ netstat -tnlp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:5000           0.0.0.0:*               LISTEN      13341/python
tcp        0      0 0.0.0.0:27017          0.0.0.0:*               LISTEN      -
tcp6       0      0 :::22                  :::*                    LISTEN      -
```

다시 위로 돌아가서 (1) id/pw를 넣고 접속해보기 과정을 시도합니다.

▼ 4) 나홀로메모장 완성본을 filezilla로 EC2에 업로드해봅니다.

1. 나홀로 메모장 완성 파일을 **week07** 폴더에 다운받습니다.



참고.

아래 코드에서는 모바일 화면에서도 화면 폭이 잘 맞아 보이도록 반응형(responsive) 스타일로 살짝 바뀌어있어요. 반응형 스타일의 가장 큰 특징은 핸드폰, 데스크탑 등 웹 브라우저 창 크기에 따라 화면이 달라진다는 것입니다.

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/551d8b62-43dd-416d-a046-f203a5a4344b/alonememo.zip>

2. MongoDB 접속 세팅을 바꿔주세요!



EC2서버의 MongoDB에 아이디와 비밀번호를 추가했으니, 우리의 pymongo에도 아이디와 비밀번호를 입력 해주어야 합니다! 그래야 pymongo가 올바르게 DB에 접근할 수 있습니다.

위 나홀로메모장 `app.py` 파일을 열어서,
아래 분홍 형광펜 부분을 바꿔주세요!

```
from pymongo import MongoClient
client = MongoClient('mongodb://test:test@localhost', 27017)
db = client.dbsparta
```

- 이 코드는 아래와 같은 뜻이에요!

```
client = MongoClient('mongodb://설정한아이디:비밀번호@내AWS아이피', 27017)
```

3. 파일질라에서 alonememo 폴더 째로 드래그 드롭으로 EC2 인스턴스의 home/ubuntu 폴더에 업로드합니다.

▼ 5) 완성본을 실행해봅니다.

1. 파일을 실행하기 전, 사용하는 패키지를 설치합니다.

```
# 패키지 설치하기
pip install requests beautifulsoup4 pymongo
```

2. 실행하기

```
# home 디렉토리로 이동
cd ~
# alonememo 폴더로 이동
cd alonememo
# 파이썬으로 코드를 실행합니다.
python app.py
```

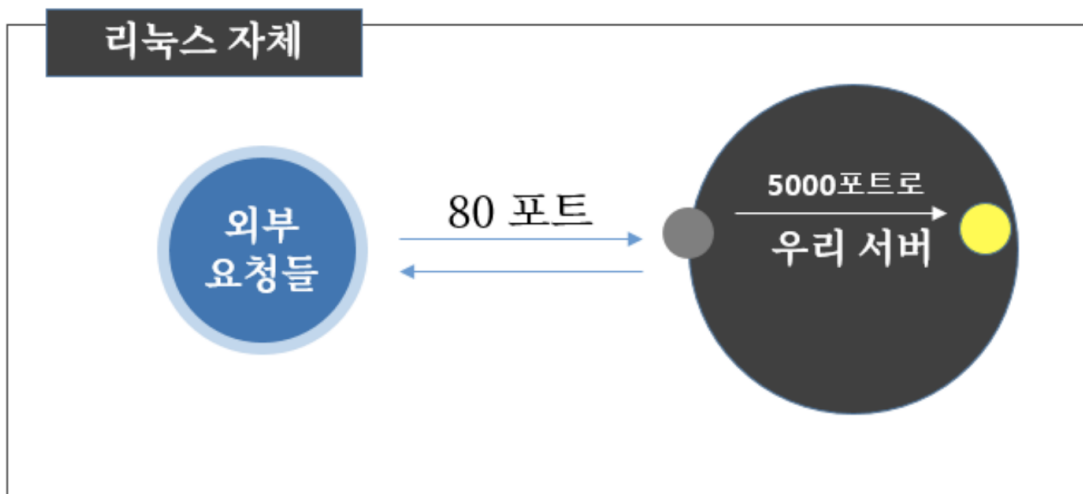
▼ 6) 접속해봅시다!

- 크롬 브라우저에서 접속하기

```
http://내AWS아이피:5000/
```

▼ 7) 포트 번호 없애기 -기본 개념

- 지금은 5000포트에서 웹 서비스가 실행되고 있습니다. 그래서 매번 :5000 이라고 뒤에 붙여줘야 하죠. 뒤에 붙는 포트 번호를 없애려면 어떻게 해야할까요?
- http 요청에서는 80포트가 기본이기 때문에, 굳이 :80을 붙이지 않아도 자동으로 연결이 됩니다.
- 포트 번호를 입력하지 않아도 자동으로 접속되기 위해, 우리는 80포트로 오는 요청을 5000 포트로 전달하게 하는 포트 포워딩(port forwarding) 을 사용하겠습니다.
- 리눅스에서 기본으로 제공해주는 포트포워딩을 사용할 것입니다. 그림으로 보면 아래와 같습니다.



▼ 8) 포트 번호 없애기 - 리눅스 자체 포트포워딩을 작동시키기

- 띄워둔 터미널에서 새롭게 설정을 적용하기 전에 돌아가고 있던 서비스(Flask)는 일단 끄고! (터미널에서 ctrl + c)
- 포트포워딩 룰을 입력합니다.

```
sudo iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 5000
```

- 다시 서비스 시작

```
# project 가 있는 폴더로 이동
# cd ~/alonememo
python app.py
```

- 안되면 아래처럼 설정이 되어있는지 확인!

| | | | |
|----|------|-------|----|
| 설명 | 인바운드 | 아웃바운드 | 태그 |
|----|------|-------|----|

| |
|----|
| 편집 |
|----|

| 유형 ⓘ | 프로토콜 ⓘ | 포트 범위 ⓘ | 소스 ⓘ |
|---------------|--------|---------|-----------|
| HTTP ✓ | TCP | 80 | 0.0.0.0/0 |
| HTTP | TCP | 80 | ::/0 |
| SSH | TCP | 22 | 0.0.0.0/0 |
| 사용자 지정 TCP 규칙 | TCP | 5000 | 0.0.0.0/0 |
| 사용자 지정 TCP 규칙 | TCP | 5000 | ::/0 |

▼ 9) 포트 번호를 떼고 접속해봅시다!

- 웹 브라우저에서 접속하기

```
http://내AWS아이피/
```

▼ 10) SSH 접속을 끊어도 서버가 계속 돌게 하기

- 처음이라 헷갈릴 수 있으니 튜터님과 차근차근 진행해보세요.

- 현재 상황

Git bash 또는 맥의 터미널을 종료하면 (=즉, SSH 접속을 끊으면) 프로세스가 종료되면서, 서버가 돌아가지 않고 있습니다. 그러나 우리가 원격접속을 끊어도, 서버는 계속 동작해야겠죠?

- 원격 접속을 종료하더라도 서버가 계속 돌아가게 하기

```
# 아래의 명령어로 실행하면 된다
nohup python app.py &
```

- 서버 종료하기 - 강제종료하는 방법

```
#아래 명령어로 미리 pid 값(프로세스 번호)을 본다
ps -ef | grep 'python'

#아래 명령어로 특정 프로세스를 종료시킨다
kill -9 [pid값]
```

- 다시 켜기

```
nohup python app.py &
```

▼ 11) SSH 접속을 종료한 뒤, 접속해봅시다!

- 웹 브라우저에서 접속하기

```
http://내AWS아이피/
```

[1시간] 프로젝트 코딩 & Todo 리스트 점검

▼ 12) 개별 면담

- 튜터와 5분 씩 개별 면담을 진행합니다.

- 프로젝트의 범위와 우선순위를 점검합니다.

후반 3시간

[월수/화목반] : 체크인 & 출석체크



튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

▼ "15초 체크인" 진행합니다

- 튜터는 타이머를 띄워주세요! (링크)
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.



배우고 적용하기 - 후반부

[2시간]: 프로젝트 개발

▼ 13) 본격적으로 개발을 진행합니다.

- Todo 리스트에 따라 개발을 진행합니다.

[1시간] 튜터 피드백 - Todo 리스트

▼ 14) 개별 면담을 통해 Todo 리스트를 튜터와 함께 선정합니다.

- 각자 프로젝트 코딩을 하고, 튜터가 5분 씩 개별 면담을 진행하여 Todo 리스트를 함께 선정합니다.

[끝]

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)
- 다음 시간까지 해올 Todo 리스트에 대한 계획을 말해도 좋습니다!

[숙제 - 단짠단짠]

숙제 1. 개발일지 작성

- 튜터님과 함께 정한 Todo 리스트 순서대로 개발합니다. 즐겁게! 🧑🏻💻
- 개발하면서 익힌 내용들을 개발일지에 자유롭게 기록합니다.



리마인드!

개발일지는, 본인을 위해 과정을 기록하는 것입니다. 잘 안되는 것, 개발하면서 배운 것들을 편하게 작성해보세요. 이렇게 능동적으로 기록하면 배운 것들이 정리되면서 학습효율이 더 높아진다고 해요.

여러분이 해낸 것들 🎉을 꼭 기록해두세요!

- 개발일지에는 아래와 같은 내용이 들어가면 좋아요!



1. 한 주 동안의 회고
2. 한 주 동안의 배운 것들
3. 이번주의 목표

▼ 참고. 이전 기수 개발일지

- 매 번 개발할 때마다 개발일지를 쓰고,
<https://travel-hama.tistory.com/3?category=874450>
<https://medium.com/@annasim0318/til-스파르타코딩클럽200415기록-438e6e40e846>
<https://medium.com/@sujinhan.me/habit-tracker-2020-04-12-04-15-86d34872e274>
- TIL 이 모이면 나중엔 종합 개발일지가 되죠!
<https://medium.com/@sujinhan.me/habit-tracker-스파르타코딩클럽-8주차-회고-7382ff1f004d>
<https://blog.naver.com/dalgona92/222043274296>
- 이렇게 개발일지 쓰는 게 습관이 되면 또 새로운 프로젝트를 시도하기 쉬워질 수도 있고요!
https://blog.naver.com/yonnie_k/222037366116

[숙제 제출]

- 슬랙에 개발일지 링크를 공유합니다.

숙제 2. 내 프로젝트 자랑하기

- EC2 에서 실행되고 있는 나홀로 메모장을 친구와 가족에게 자랑하기! 🥳 페이스북, 트위터, 카카오톡 마음껏 자랑해보세요! 내가 스파르타코딩클럽에서 7주만에 웹 서비스 만들었다!
- 내 EC2 인스턴스의 ip 주소를 공유하면 되겠죠?



체크아웃

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! (링크)
- 체크인처럼, 현재 본인의 감정상태와 수업후기에 관해 이야기합니다.
 - 예. 오늘 내 프로젝트를 시작해서 재밌었어요!
 - 자유롭게 해주셔도 좋고, KPT에 맞춰해주셔도 좋아요.
 - Keep : 오늘 수업을 하면서 좋았던 것, 앞으로도 할 행동 / Problem : 아쉬워서 고쳐보면 좋을 것 / Try : 고치기 위해 내가 할 시도
 - 예) (Keep) 웹사이트를 빠르게 만들어서 좋았어요. (Problem) 아직 API 사용하는 부분이 헛갈려요. (Try) 집에 가서 숙제하면서 복습해볼게요!
- 튜터님은 체크아웃과 함께 출석 체크(링크)를 진행해주세요!
 - 출석체크 - (변동이 있다면 다시 제출해주세요!)

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

[설치] - 다음 시간을 위해 미리 설치해와야 할 것들

- 가비아 가입하기
 - 가입링크: <https://www.gabia.com>
 - 가비아에서 할인이벤트(500원/1년)를 진행하는 도메인을 구매해서 진행 할 예정입니다.