

# 빅데이터시스템

BIG DATA ANALYTICS  
TECHNOLOGY

과학기반의 빅데이터분석



# 목 차

## 01 빅데이터 산업

빅데이터산업용어, 빅데이터플랫폼, 빅데이터에코시스템,  
빅데이터서비스프레임워크

## 02 빅데이터 분석방법과 접근법

분석방법, 분석접근법

## 03 데이터 과학 방법론

데이터과학, 연구목표설정, 데이터수집, 데이터준비, 데이터탐색,  
데이터모델링, 결과발표및분석자동화

## 04 Flask를 활용한 웹서버 구축

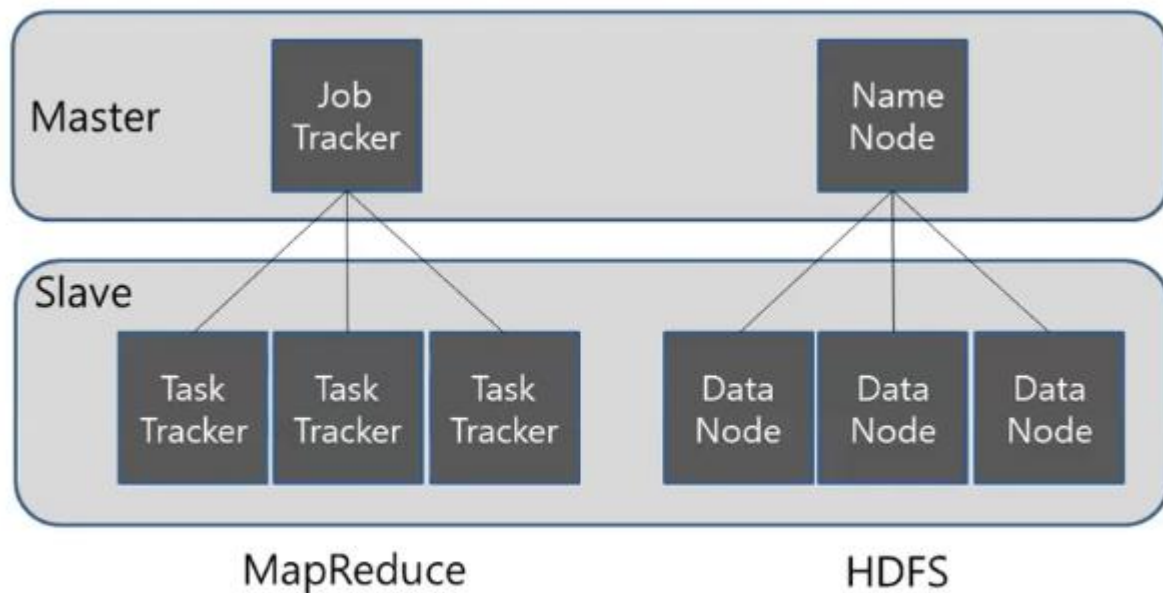
Flask란, 디렉토리구성, 디렉토리생성, html파일생성, 이미지파일생성,  
css파일생성, js파일생성, 라이브러리파일생성, 엑셀파일생성,  
Flask Main파일생성, 웹서버실행

# \*. 전수업리뷰

## ⌚ 하둡(Hadoop) 개념

※ Hadoop(High-Availability Distributed Object-Oriented Platform) : Java 로 개발되었으며, 클러스터에서 사용할 수 있는 분산파일시스템과 분산처리시스템을 제공하는 아파치 소프트웨어 재단의 오픈 소스 프레임워크

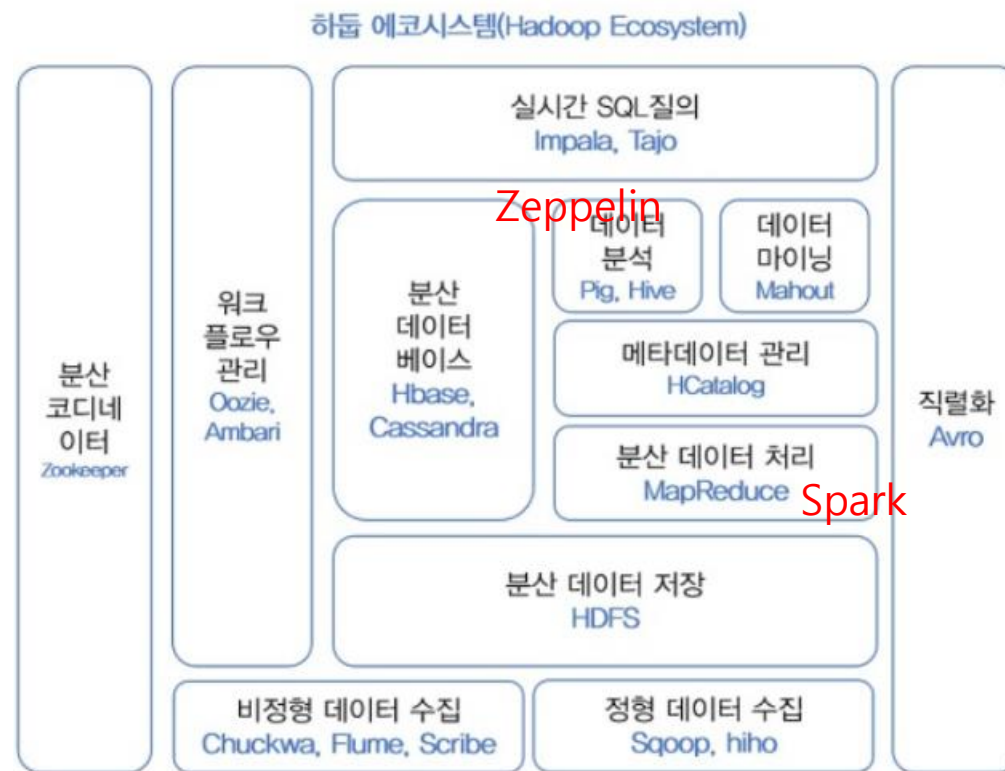
### 기본골격



한기철, K-ICT 빅데이터 교육교재 중 발췌

- 맵리듀스는 일을 어떻게 분배할 것인지 결정
- HD파일시스템은 데이터를 어떻게 분산저장할지를 결정

### 기능확장



하둡 프로그래밍(위키북스)

# \*. 전수업리뷰

## ⌚ 환경설정파일

파일 구분	내 용	비 고
hdfs-site.xml	<ul style="list-style-type: none"> <li>하둡 파일시스템 환경설정</li> </ul>	
core-site.xml	<ul style="list-style-type: none"> <li>HDFS, MapReduce 환경설정</li> </ul>	
yarn-site.xml	<ul style="list-style-type: none"> <li>Resource Manager 및 Node Manager 환경설정</li> </ul>	
mapred-site.xml	<ul style="list-style-type: none"> <li>MapReduce 어플리케이션 환경설정</li> </ul>	
hadoop-env.sh	<ul style="list-style-type: none"> <li>하둡이 구동되는 데 필요한 환경 설정</li> </ul>	
workers	<ul style="list-style-type: none"> <li>하둡의 worker 로 동작할 서버 호스트 이름 설정</li> </ul>	<ul style="list-style-type: none"> <li>slaves</li> </ul>
masters	<ul style="list-style-type: none"> <li>하둡의 master 로 동작할 서버 호스트 이름 설정</li> </ul>	

# \*. 전수업리뷰

## ⌚ 가상머신 (wsl : windows subsystem for linux)

Powershell 관리자권한으로 실행  
디렉토리생성 : \_seok

Wsl -l -o

```
PS C:\>
PS C:\> cd \
PS C:\> mkdir _seok

디렉터리 : C:\

Mode                LastWriteTime
----                -
d-----         2023-09-08    오전 4:42

PS C:\> cd _seok
PS C:\_seok>
PS C:\_seok>
```

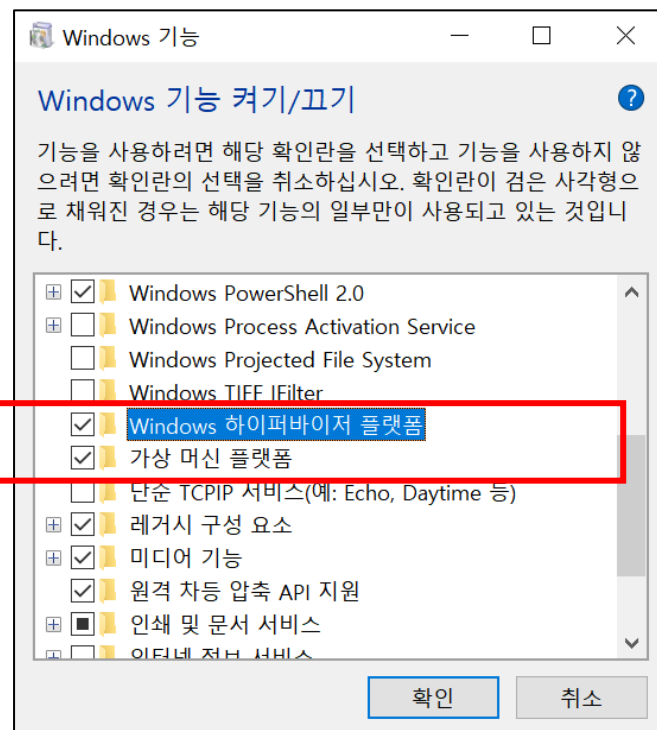
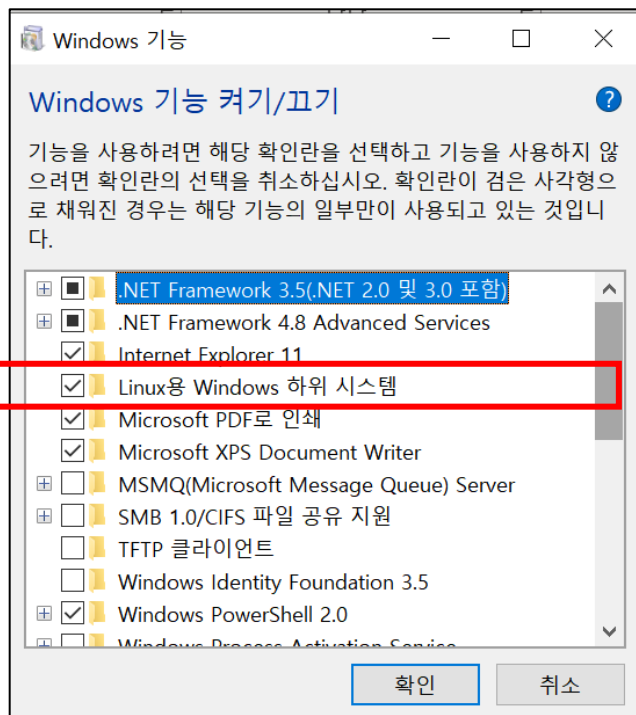
```
PS C:\_seok> wsl -l -o
다음은 설치할 수 있는 유효한 배포 목록입니다.
기본 배포는 '*' 로 표시됩니다.
'wsl --install -d <Distro>'을(를) 사용하여 설치하세요.

NAME                                FRIENDLY NAME
* Ubuntu                            Ubuntu
Debian                              Debian GNU/Linux
kali-linux                          Kali Linux Rolling
Ubuntu-18.04                        Ubuntu 18.04 LTS
Ubuntu-20.04                        Ubuntu 20.04 LTS
Ubuntu-22.04                        Ubuntu 22.04 LTS
OracleLinux_7_9                     Oracle Linux 7.9
OracleLinux_8_7                     Oracle Linux 8.7
OracleLinux_9_1                     Oracle Linux 9.1
openSUSE-Leap-15.5                  openSUSE Leap 15.5
SUSE-Linux-Enterprise-Server-15-SP4 SUSE Linux Enterprise Server 15 SP4
SUSE-Linux-Enterprise-15-SP5        SUSE Linux Enterprise 15 SP5
openSUSE-Tumbleweed                 openSUSE Tumbleweed
PS C:\_seok>
```

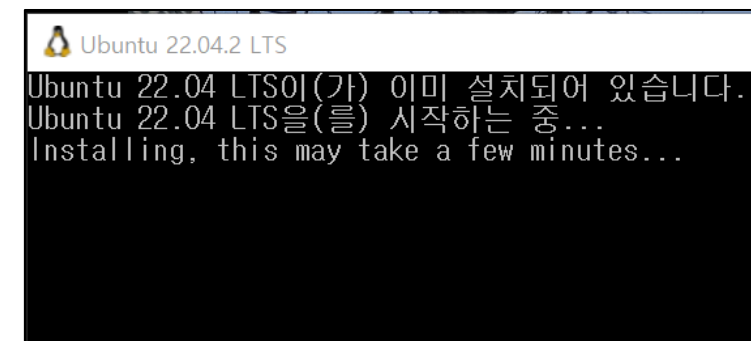
# \*. 전수업리뷰

⌚ wsl

제어판-프로그램및기능-Windows 기능 켜기/끄기



PC 리부트



왼쪽화면이 수정이 되면, PC를 리스타트 한다는 화면이 뜬다

세 곳이 체크되어야 하며, 만약 모두 되어 있다면, 일부러 체크를 한번 끄고, <확인>한 다음, 다시 체크를 켜고 <확인>하면, PC를 리스타트 한다는 화면이 나올 것임



# \*. 전수업리뷰



<https://www.oracle.com/kr/java/technologies/downloads/#jdk19-linux>

※ 어제 수업 기준으로,  
jvm이 jdk-21 로 업그레이드 됨

이 페이지 이후로는  
jdk-21 로 이름을 바꾸어서  
환경설정을 해야 함

Java downloads Tools and resources Java archive

JDK 20 JDK 17 GraalVM for JDK 20 GraalVM for JDK 17

### JDK Development Kit 20.0.2 downloads

JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions](#).

JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

Linux macOS Windows

Product/file description	File size	Download
ARM64 Compressed Archive	181.55 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.tar.gz">https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.tar.gz</a>
ARM64 RPM Package	181.27 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.rpm">https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.rpm</a>
x64 Compressed Archive	183.11 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz">https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz</a>
x64 Debian Package	155.91 MB	<a href="https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.deb">https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.deb</a>

오른쪽 마우스 버튼  
링크 주소 복사

# \*. 전수업리뷰

## ⌚ python3

```
[root@linux ~]# python -V
Command 'python' not found, did you mean:
  command 'python3' from deb python3
  command 'python' from deb python-is-python3
[root@linux ~]# python3 -V
Python 3.10.12
[root@linux ~]# apt-get install -y python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (22.0.2-0)
0 upgraded, 0 newly installed, 0 to remove and 52 not installed.
```

python -V  
python3 -V

apt-get install -y python3-pip

만약, python 이 없다면,

```
#-- python 3.10
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt-get update
##### python 3.10 확인
apt list | grep python3.10
##### 설치
sudo apt-get install python3.10
##### alternatives 에 등록
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.8 1
sudo update-alternatives --install /usr/bin/python3 python3 /usr/bin/python3.10 2
##### 기본 호출을 3.10 으로 변경
sudo update-alternatives --config python3

##### 오류 발생 시
sudo apt-get remove python3-apt
sudo apt-get install python3-apt
sudo apt-get install --reinstall python3-apt
```



# \*. 전수업리뷰

## ⌚ Hadoop

- <https://hadoop.apache.org/releases.html>

다운로드

Hadoop은 편의를 위해 해당 바이너리 타르볼과 함께 소스 코드 타르볼로 릴리스됩니다. 다운로드는 미리 사이트를 통해 배포되며 GPG 또는 SHA-512를 사용하여 변조 여부를 확인해야 합니다.

버전	출시일	소스 다운로드	바이너리 다운로드	릴리스 노트
3.3.6	2023년 6월 23일	<a href="#">소스 (체크섬 서명)</a>	<a href="#">바이너리 (체크섬 서명)</a> <a href="#">bin-aarch64 (체크섬 서명)</a>	<a href="#">발표</a>
3.2.4	2022년 7월 22일	<a href="#">소스 (체크섬 서명)</a>	<a href="#">바이너리 (체크섬 서명)</a>	<a href="#">발표</a>
2.10.2	2022년 5월 31일	<a href="#">소스 (체크섬 서명)</a>	<a href="#">바이너리 (체크섬 서명)</a>	<a href="#">발표</a>

GPG를 사용하여 Hadoop 릴리스를 확인하려면:

1. [미리 사이트](#)에서 `hadoop-XYZ-src.tar.gz` 릴리스를 다운로드합니다.
2. [Apache](#)에서 서명 파일 `hadoop-XYZ-src.tar.gz.asc`를 다운로드합니다.
3. [Hadoop KEYS](#) 파일을 다운로드합니다.
4. `gpg -키` 가져오기
5. `gpg -hadoop-XYZ-src.tar.gz.asc` 확인

SHA-512를 사용하여 빠른 검사를 수행하려면:

1. [미리 사이트](#)에서 `hadoop-XYZ-src.tar.gz` 릴리스를 다운로드합니다.
2. [Apache](#)에서 체크섬 `hadoop-XYZ-src.tar.gz.sha512` 또는 `hadoop-XYZ-src.tar.gz.mds`를 다운로드합니다.

- 최신버전의 <바이너리(체크섬 서명)> 클릭

커뮤니티 주도 개발 "THE APACHE WAY"

다음 사이트에서 다운로드하는 것이 좋습니다.

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz>

대체 다운로드 위치는 아래에 제안되어 있습니다.

PGP 서명(파일) 또는 해시(또는 파일)를 사용하여 다운로드.

**HTTP**

<https://dlcdn.apache.org/hadoop/common/hadoop-3.3.4/hadoop-3.3.4.tar.gz>

새 탭에서 링크 열기  
새 창에서 링크 열기  
시크릿 창에서 링크 열기  
다른 이름으로 링크 저장...  
**링크 주소 복사**  
검사

- 링크에서 오른쪽 마우스 <링크 주소 복사> 클릭

# \*. 전수업리뷰

## ⌚ Hadoop

```
[root@linux util]#
[root@linux util]# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.18.55.150 netmask 255.255.240.0 broadcast 172.18.63.255
    inet6 fe80::215:5dff:fe12:9d47 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:12:9d:47 txqueuelen 1000 (Ethernet)
```

ifconfig

```
[root@linux util]#
[root@linux util]#
[root@linux util]# vi /etc/hosts
```

vi /etc/hosts

본인의 우분투 IP주소로  
변경해야 함

```
root@DESKTOP-28CEK7O: ~
# This file was automatically generated by WSL
# [network]
# generateHosts = false
127.0.0.1 localhost
127.0.1.1 DESKTOP-28CEK7O. DESKTO
172.18.55.150 linux.home
172.18.55.150 nn1
172.18.55.150 nn2
172.18.55.150 dn1
172.18.55.150 dn2
172.18.55.150 dn3
```

<vi 명령어 사용>

j 로 맨 아래로 이동  
o 누르면, 한줄 아래부터 키 입력 가능

```
172.18.55.150 linux.home
172.18.55.150 nn1
172.18.55.150 nn2
172.18.55.150 dn1
172.18.55.150 dn2
172.18.55.150 dn3
```

ESC 콜론(:) wq EnterKey 입력

# \*. 전수업리뷰

## ⌚ Hadoop

```
[root@linux util]#  
[root@linux util]#  
[root@linux util]# mv hadoop-3.3.6 hadoop  
[root@linux util]#  
[root@linux util]#  
[root@linux util]# cd ~  
[root@linux ~]#  
[root@linux ~]# vi .bashrc
```

mv hadoop-3.3.4 hadoop

cd ~

vi .bashrc



source .bashrc

```
#----- PYTHON -----  
export PYTHONPATH=/usr/bin/python3  
export PYSPARK_PYTHON=/usr/bin/python3  
  
#----- HADOOP -----  
export HADOOP_HOME=/util/hadoop  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

<vi 명령어 사용>

j 로 맨 아래로 이동  
o 누르면, 한줄 아래부터 키 입력 가능

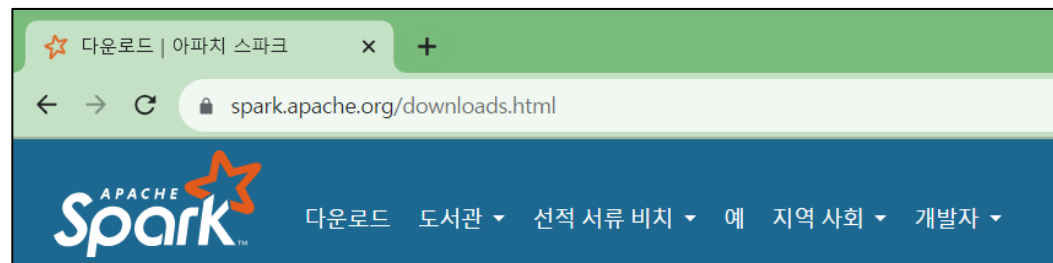
```
#----- HADOOP -----  
export HADOOP_HOME=/util/hadoop  
export HADOOP_COMMON_HOME=$HADOOP_HOME  
export HADOOP_HDFS_HOME=$HADOOP_HOME  
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop  
export HADOOP_YARN_HOME=$HADOOP_HOME  
export HADOOP_MAPRED_HOME=$HADOOP_HOME  
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

ESC 콜론(:) wq EnterKey 입력

# \*. 전수업리뷰

- <https://spark.apache.org/downloads.html>

## ⌚ Spark



### Apache Spark™ 다운로드

1. Spark 릴리스를 선택하세요. 3.4.1(2023년 6월 23일) ▾
2. 패키지 유형을 선택하세요: Apache Hadoop 3.3 이상용으로 사전 구축됨

3. Spark 다운로드: Spark-3.4.1-bin-hadoop3.tgz

4. 다음 절차에 따라 3.4.1 서명, 체크섬 및 프로젝트 릴리스 KEYS를 사용하여 이 릴리스를 검증하십시오.

Spark 3은 일반적으로 Scala 2.12로 사전 구축되었으며 Spark 3.2+는 Scala 2.13으로 사전 구축된 추가 배포판을 제공합니다.

### 스파크와 연결

Spark 아티팩트는 [Maven Central](#)에서 호스팅됩니다. 다음 좌표를 사용하여 Maven 종속성을 추가할 수 있습니다.





# \*. 전수업리뷰

## ⌚ Spark

- 스팍의 기본 변수 및 웹페이지 설정

또는

vi spark-env.sh

gnome-text-editor spark-env.sh

```
[root@localhost conf]#  
[root@localhost conf]# vi spark-env.sh
```

```
Open ▾ + spark-env.sh  
/util/spark/conf  
#!/usr/bin/env bash  
export SPARK_WORKER_INSTANCES=3  
export SPARK_HOME=/util/spark  
export SPARK_CONF_DIR=/util/spark/conf  
export JAVA_HOME=/usr/jdk-20.0.2  
export SPARK_MASTER_WEBUI_PORT=8080  
#export HADOOP_HOME=/util/hadoop  
#export HADOOP_CONF_DIR=/util/hadoop/etc/Hadoop  
  
#  
# Licensed to the Apache Software Foundation (ASF) under one  
# contributor license agreements. See the NOTICE file distributed
```

<vi 명령어 사용>

j 로 아래로 한칸 이동  
o 누르면, 한줄 아래부터 키 입력 가능

```
export SPARK_WORKER_INSTANCES=3  
export SPARK_HOME=/util/spark  
export SPARK_CONF_DIR=/util/spark/conf  
export JAVA_HOME=/usr/jdk-20.0.2  
export SPARK_MASTER_WEBUI_PORT=8080  
#export HADOOP_HOME=/util/hadoop  
#export HADOOP_CONF_DIR=/util/hadoop/etc/Hadoop
```

ESC 콜론(:) wq EnterKey 입력

# \*. 전수업리뷰

## ⌚ Spark

- 콘솔과 마스터, 제플린 Port 방화벽 열기

```
[root@linux sbin]# apt install firewalld
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
[root@linux sbin]# firewall-cmd --permanent --add-service=http
success
[root@linux sbin]# firewall-cmd --permanent --add-port=80/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=9090/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=7077/tcp
success
[root@linux sbin]# firewall-cmd --permanent --add-port=10101/tcp
success
[root@linux sbin]#
[root@linux sbin]# firewall-cmd --reload
Error: COMMAND_FAILED: 'python-nftables' failed: internal:0:0:0:
```

```
[root@linux sbin]# firewall-cmd --list-all
public
target: default
icmp-block-inversion: no
interfaces:
sources:
services: dhcpv6-client http ssh
ports: 80/tcp 9090/tcp 7077/tcp 10101/tcp
protocols:
forward: yes
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
[root@linux sbin]#
```

apt install firewalld -y

firewall-cmd --permanent --add-service=http

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

firewall-cmd --permanent --add-port=8080/tcp

firewall-cmd --permanent --add-port=7077/tcp

firewall-cmd --permanent --add-port=9090/tcp

firewall-cmd --permanent --add-port=8081/tcp

firewall-cmd --permanent --add-port=8082/tcp

firewall-cmd --permanent --add-port=8083/tcp

firewall-cmd --permanent --add-port=4040/tcp

firewall-cmd --list-all

아파치용

Spark 콘솔

Spark Master

zeppelin notebook

worker-1

worker-2

worker-3

공유 프로세스

# \*. 전수업리뷰

## ⌚ Spark

- PC의 크롬에서 우분투의 Spark Master 접속 확인 <http://linux.home:8080>

Spark Master at **spark://DESKTOP-28CEK7O.:7077**

URL: spark://DESKTOP-28CEK7O.:7077  
 Alive Workers: 0  
 Cores in use: 0 Total, 0 Used  
 Memory in use: 0.0 B Total, 0.0 B Used  
 Resources in use:  
 Applications: 0 Running, 0 Completed  
 Drivers: 0 Running, 0 Completed  
 Status: ALIVE

▼ **Workers (0)**

Worker Id	Address	State	Cores	Memory	Resources

▼ **Running Applications (0)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

▼ **Completed Applications (0)**

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

이 부분을 마우스로 클릭해서 복사함

아직 Workers 는 살아 있지 않음

# \*. 전수업리뷰

## ⌚ Spark

```
[root@linux sbin]#  
[root@linux sbin]# cd /util  
[root@linux util]#  
[root@linux util]# mkdir test  
  
[root@linux util]#  
[root@linux util]# cd test  
[root@linux test]#  
[root@linux test]#  
  
[root@linux test]#  
[root@linux test]#  
[root@linux test]# vi pyspark-test.py
```

```
cd /util  
mkdir test  
cd test  
vi pyspark-test.py
```

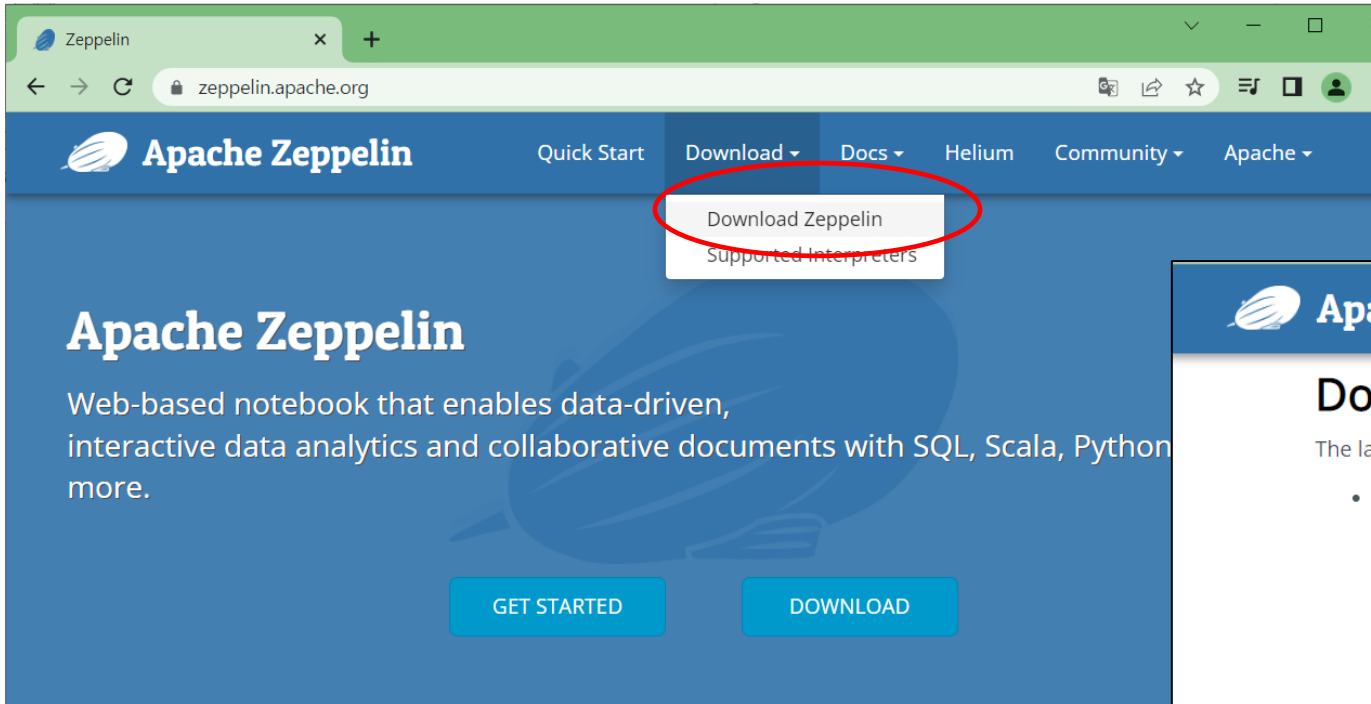
- pyspark-test.py

```
from pyspark import SparkContext, SparkConf  
  
conf = SparkConf()  
conf.setMaster("spark://DESKTOP-28CEK7O.:7077")  
conf.setAppName("seokill")  
sc = SparkContext(conf=conf)  
  
print("="*50, "\n")  
print("안녕하세요~스파크님~")  
print(99 * 1000000)  
print(sc)  
print("="*50, "\n")
```



# \*. 전수업리뷰

## ⌚ Zeppelin



<https://zeppelin.apache.org/>



# \*. 전수업리뷰

## ⌚ Zeppelin

```
<configuration>
<property>
<name>zeppelin.server.addr</name>
<value>127.0.0.1</value>
<description>Server binding address</description>
</property>
<property>
<name>zeppelin.server.port</name>
<value>8080</value>
<description>Server port.</description>
</property>
</configuration>
```

↓

```
<configuration>
<property>
<name>zeppelin.server.addr</name>
<value>192.168.121.128</value>
<description>Server binding address</description>
</property>
<property>
<name>zeppelin.server.port</name>
<value>9090</value>
<description>Server port.</description>
</property>
</configuration>
```

<vi 명령어 사용>

j 로 아래로 이동  
 l 로 오른쪽으로 → 127 문자 까지  
 x 로 문자 삭제 → 127.0.0.1 모두 삭제  
 i 를 누르고, 172.18.55.150 입력

ESC 누르고,

j, h, l 키로 아래로 이동 → 8080 문자까지  
 x 로 문자 삭제 → 8080 모두 삭제  
 i 를 누르고, 9090 입력

ESC 콜론(:) wq EnterKey 입력

# \*. 전수업리뷰

## ⌚ Zeppelin

- 맨 아래로 이동 하여 아래의 내용 입력 (맨 아래로 이동 \$G)

```
##### Zeppelin impersonation configuration
# export ZEPPELIN_IMPERSONATE_CMD      # Optional, when
# {ZEPPELIN_IMPERSONATE_USER} bash -c '
# export ZEPPELIN_IMPERSONATE_SPARK_PROXY_USER #Optional
# use --proxy-user option with Spark interpreter when imp
```

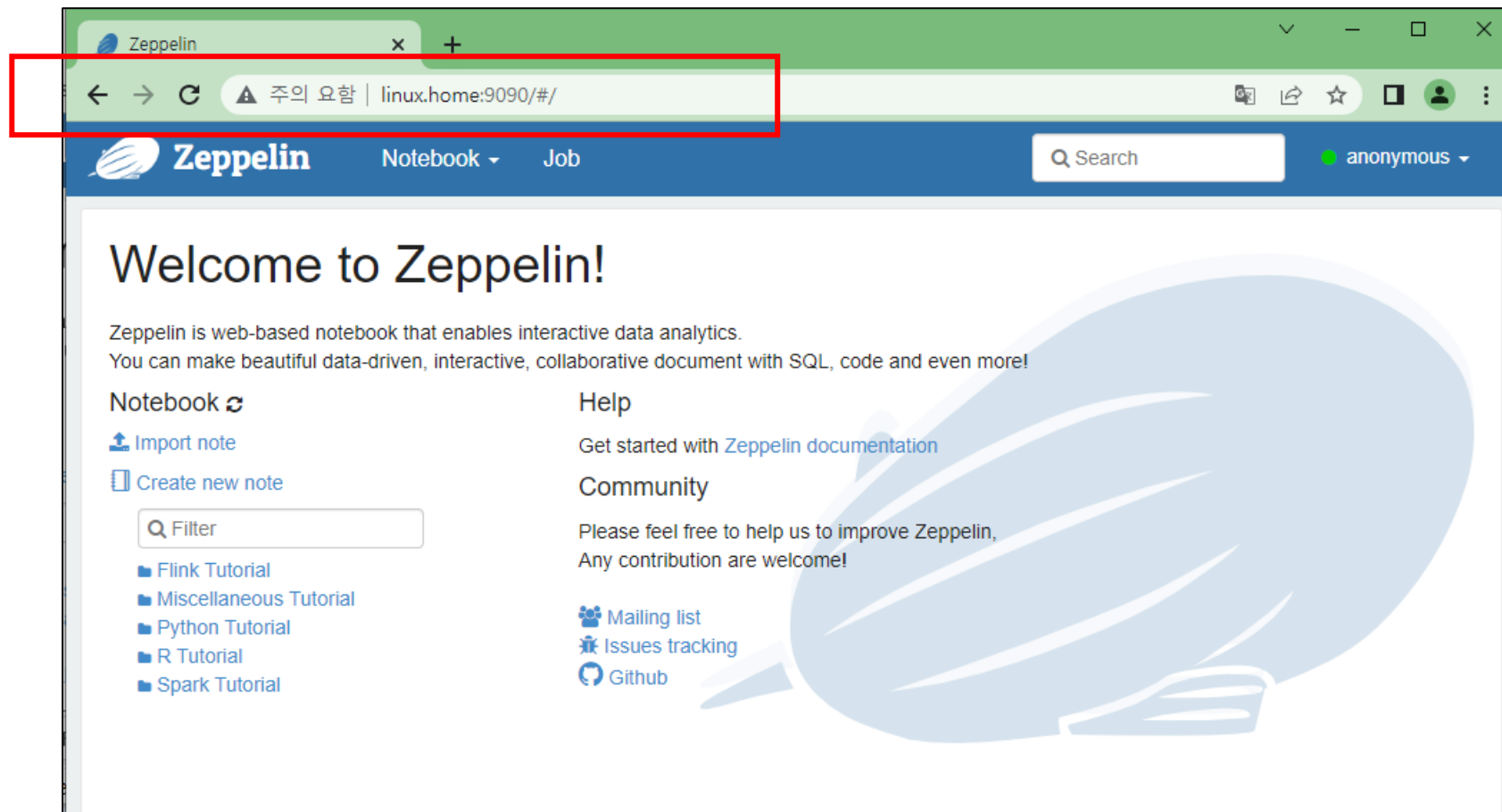
```
export JAVA_HOME=/usr/jdk-19.0.1
export SPARK_MASTER=spark://linux.home:7077
export SPARK_HOME=/util/spark
#export HADOOP_HOME=/util/hadoop
#export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
#export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
#export MASTER=spark://linux.home:7077
#export ZEPPELIN_PORT=9090
#export PYTHONPATH=/usr/bin/python3
#export PYSARK_PYTHON=/usr/bin/python3
#export PYSARK_DRIVER_PYTHON=/usr/bin/python3
```

```
export JAVA_HOME=/usr/jdk-20.0.2
export SPARK_MASTER=spark:// DESKTOP-28CEK7O.: 7077
export SPARK_HOME=/util/spark
#export HADOOP_HOME=/util/hadoop
#export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
#export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
#export MASTER=spark://linux.home:7077
#export ZEPPELIN_PORT=9090
#export PYTHONPATH=/usr/bin/python3
#export PYSARK_PYTHON=/usr/bin/python3
#export PYSARK_DRIVER_PYTHON=/usr/bin/python3
```

# \*. 전수업리뷰

## ⌚ Zeppelin

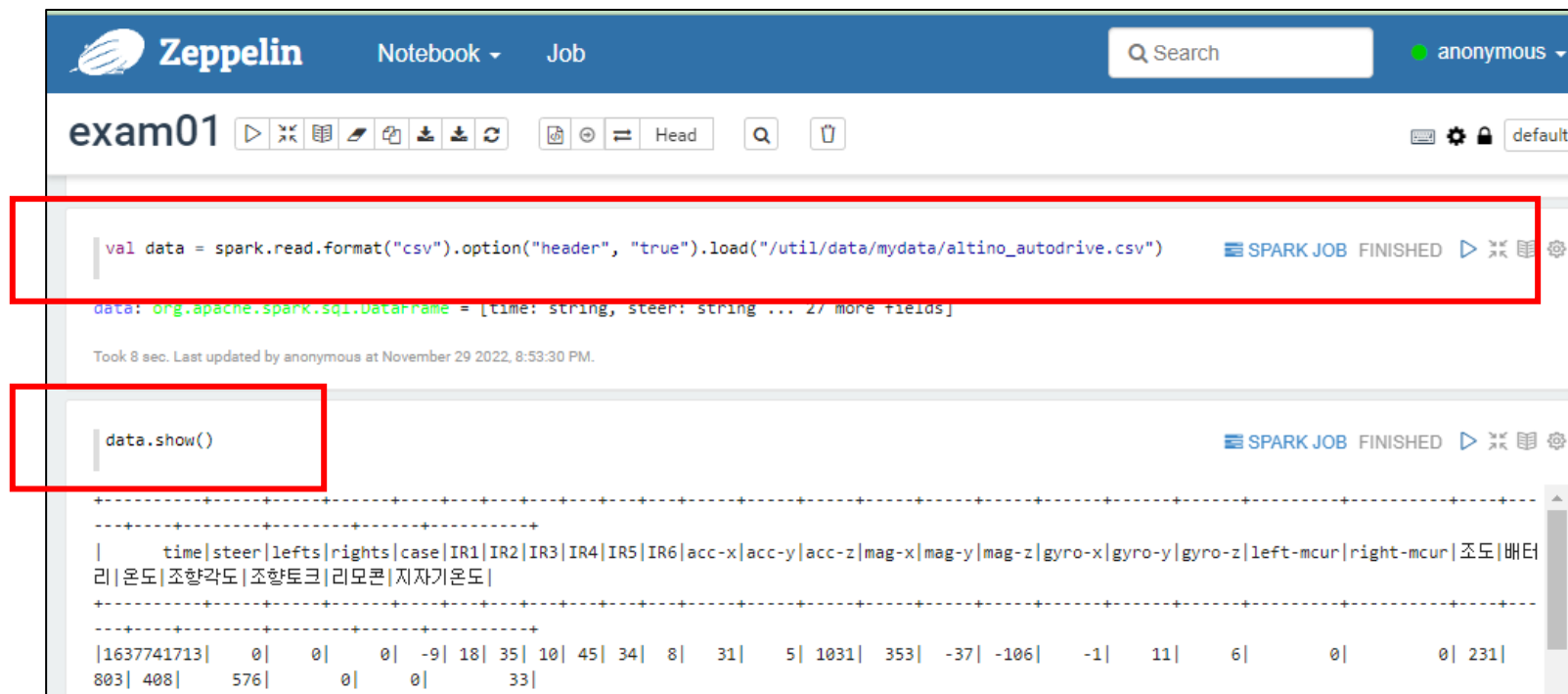
http://linux.home:9090





# \*. 전수업리뷰

## ⌚ Zeppelin



The screenshot shows the Zeppelin Notebook interface. The top bar includes the Zeppelin logo, 'Notebook' and 'Job' tabs, a search bar, and a user profile dropdown. The notebook title is 'exam01'. Below the title bar, there are two code blocks, each with a red border. The first code block contains the following Spark SQL command:

```
val data = spark.read.format("csv").option("header", "true").load("/util/data/mydata/altino_autodrive.csv")
```

The second code block contains the following command:

```
data.show()
```

Below the code blocks, the output of the first command is displayed as a table with columns: time, steer, lefts, rights, case, IR1, IR2, IR3, IR4, IR5, IR6, acc-x, acc-y, acc-z, mag-x, mag-y, mag-z, gyro-x, gyro-y, gyro-z, left-mcur, right-mcur, 조도, 배터리 온도, 조향각도, 조향토크, 리모콘, 지자기온도. The table shows data for a specific time point.

```
val data = spark.read.format("csv").option("header", "true").load("/util/data/mydata/altino_autodrive.csv")
```

```
data.show()
```

# \*. 전달 사항



교재

주교재

- PowerPoint 로 만든 pdf 자료
- 데이터 과학 기반의 파이썬 빅데이터 분석 (이지영 지음, 한빛아카데미)

부교재

- 필요 시, 영상 공유



# \*. 전달 사항

## RoadMap

### Hadoop설치

- ✓ VM 셋업
- ✓ JDK
- ✓ Python
- ✓ Hadoop Engine
- ✓ Spark Engine
- ✓ Zeppelin

### 빅데이터분석

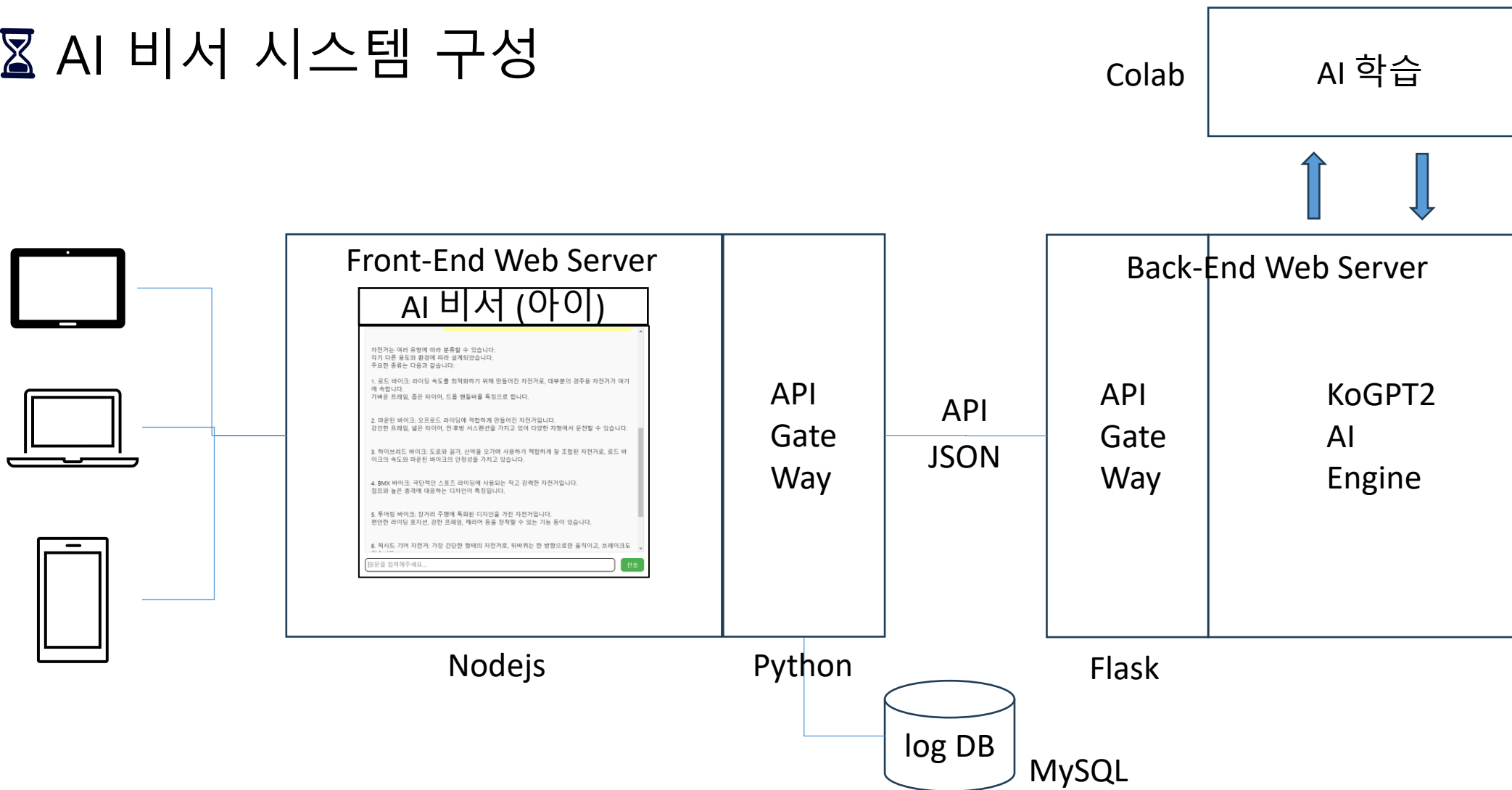
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

### AI 비서학습

- ✓ 챗봇 데이터 수집
- ✓ Flask 웹서버
- ✓ Nodejs API 연동
- ✓ KoGPT2 환경구성
- ✓ Colab을 이용한 학습
- ✓ 말풍선생성기 활용
- ✓ MySQL
- ✓ 챗봇 비서 만들기

# \*. 전달 사항

## ⌚ AI 비서 시스템 구성





# 1. 빅데이터 산업

## ⌚ 빅데이터 산업 용어

- 빅데이터 산업은 관련된 여러 분야가 유기적으로 결합된 시스템

구 분	내 용	비 고
빅데이터 플랫폼	데이터 관점에서 빅데이터를 수집·저장·분석하는 프로세스와 그에 필요한 자원의 유기적 결합	
빅데이터 에코시스템	빅데이터 플랫폼에 서비스 산업을 결합하여 고객에게 가치를 전달하는 유기적 공동체	
빅데이터 서비스 프레임워크	빅데이터 에코시스템에서 서비스 공급자를 분류하고 서비스 유형과 수준을 파악	

# 1. 빅데이터 산업

## ⌚ 빅데이터 플랫폼

### ■ 데이터 플랫폼의 발전

- 데이터 플랫폼은 정형화된 형태로 데이터를 저장하는 파일 시스템으로 시작
- 다수가 동시에 사용할 수 있는 데이터베이스와 데이터 웨어하우스(DW)로 발전
- 폭발적으로 증가하는 데이터를 저장 및 유통하기 위한 빅데이터 플랫폼으로 진화

### ■ 빅데이터 플랫폼의 개념

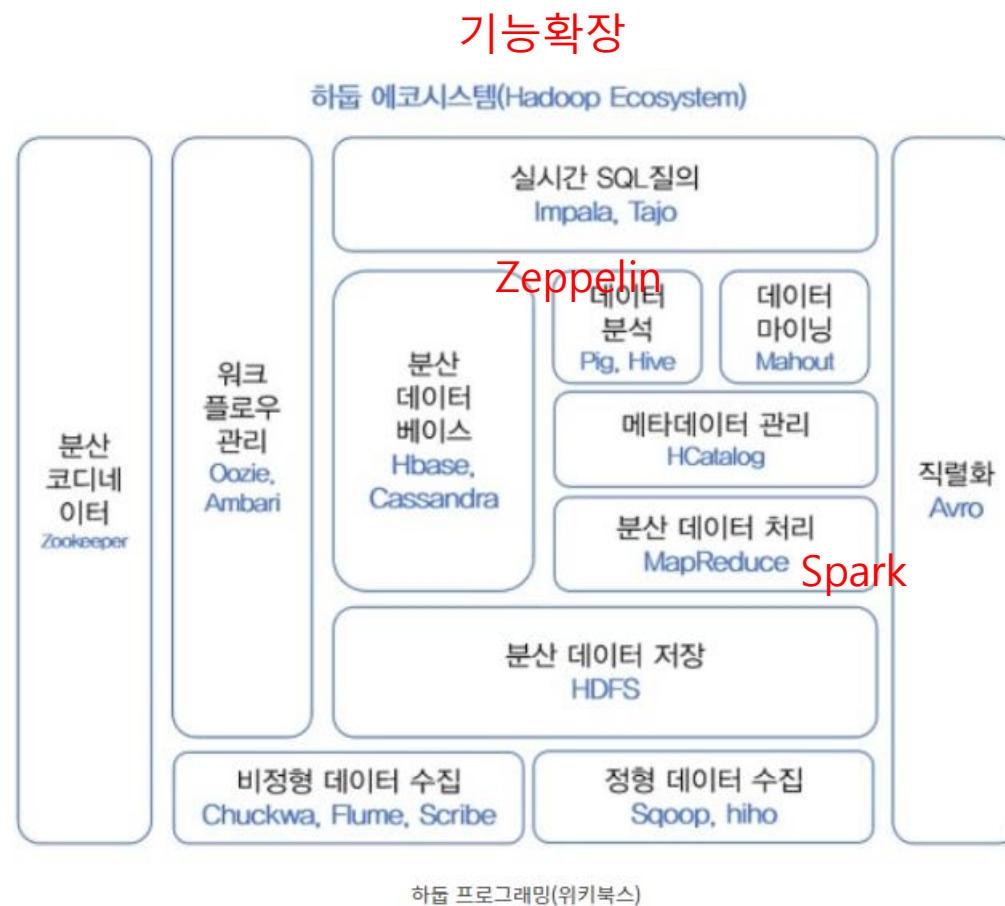
- 대량의 데이터를 저장 및 분석, 처리할 수 있는 대용량의 고속 저장 공간 보유
- 고성능 계산 능력과 실시간으로 발생하는 빅데이터를 처리 및 분석하여 일관성을 유지
- 빅 데이터에서 발생하는 개인 정보를 위한 정보 보안 관리체계 지원도 필요
- 빅데이터 플랫폼은 오픈 소스인 하둡을 근간으로 많이 사용

# 1. 빅데이터 산업

## ⌚ 빅데이터 에코시스템

### ■ 빅데이터 생태계

- 플랫폼을 기반으로 단독으로 구성되지 않고, 다양한 비즈니스와 결합
- 종합적인 관점에서 고객에게 가치를 전달할 수 있는 환경 구성
- 데이터를 다양한 경로를 통해 수집하고 다양한 파이프라인을 만드는 영역이 존재
- 수집된 데이터를 정제하고 체계적으로 저장 및 관리하도록 다양한 데이터 분석 인프라 구성
- 실제 비즈니스에 활용되는 시각화 모듈 구성



# 1. 빅데이터 산업

## ⌚ 빅데이터 서비스 프레임워크

### ■ 공급자 분류

- 빅데이터 서비스 프레임워크는 빅데이터 시장을 효율적으로 이해하기 위한 것
- 에코시스템 안에서 서비스 공급자를 분류하고 서비스 유형과 수준을 파악하는 것이 필요
- 공급하는 서비스의 유형/수준에 따라 빅데이터 서비스 공급자와 애플리케이션 공급자로 분류

공급 서비스 유형에 따른 분류	공급 서비스 수준에 따른 분류
<ul style="list-style-type: none"> <li>■ 하드웨어 공급자 <ul style="list-style-type: none"> <li>- 자체 데이터센터 및 클라우드 시스템을 통해 빅데이터 서비스를 위한 인프라를 공급</li> </ul> </li> <li>■ 처리 소프트웨어 공급자 <ul style="list-style-type: none"> <li>- 서비스 소비자가 저장한 빅데이터를 효과적으로 저장 및 처리할 수 있는 소프트웨어를 제공</li> </ul> </li> <li>■ 분석 소프트웨어 공급자 <ul style="list-style-type: none"> <li>- 서비스 소비자의 빅데이터를 분석할 소프트웨어를 제공</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>■ 인프라 계층 <ul style="list-style-type: none"> <li>- 빅데이터를 위한 기초 작업을 담당하는 하드웨어나 운영체제를 제공(가상화 컴퓨팅 서비스 포함)</li> </ul> </li> <li>■ 플랫폼 계층 <ul style="list-style-type: none"> <li>- 클라우드 컴퓨팅 서비스나 하드웨어에 종속되지 않는 처리 및 분석 소프트웨어 등을 제공</li> </ul> </li> <li>■ 애플리케이션 계층 <ul style="list-style-type: none"> <li>- 소비자가 빅데이터와 소통하는 매커니즘을 제공</li> <li>- 빅데이터 처리 결과를 바탕으로 소비자가 원하는 분석 결과를 제공하거나 시장에 유통</li> </ul> </li> </ul>

# 1. 빅데이터 산업

## ⌚ 빅데이터 서비스 프레임워크

### ■ 서비스 유형

서비스유형

서비스수준

	하드웨어	처리소프트웨어	분석 소프트웨어
인프라	<b>하드웨어-인프라 유형</b> <ul style="list-style-type: none"> <li>기업 등에서 자체 데이터센터를 구축할 수 있게 해주는 서비스 유형</li> <li>사적 데이터를 중심으로 하는 기업형 솔루션과 공적 데이터를 중심으로 하는 플랫폼 서비스로 구분</li> <li>IBM, HP, 오라클 등의 기업용 하드웨어 솔루션 제품이 여기에 해당</li> </ul>	<b>처리 소프트웨어-인프라 유형</b> <ul style="list-style-type: none"> <li>하드웨어와 소프트웨어를 함께 제공하는 서비스 유형</li> <li>대용량 데이터를 다루기 위해 필요한 분산 저장 및 병렬 처리 인프라에 처리 솔루션제공</li> <li>기업용 솔루션 사업을 하는 오라클, IBM, HP, EMC 등의 기업에서 자사의 하드웨어와 특화된 소프트웨어를 통합해서 제공</li> </ul>	
플랫폼	<b>하드웨어-플랫폼 유형</b> <ul style="list-style-type: none"> <li>클라우드를 기반으로 서비스를 제공하는 유형</li> <li>기존의 클라우드 컴퓨팅 시스템을 사용해 빅데이터 서비스를 제공</li> </ul>	<b>처리 소프트웨어-플랫폼 유형</b> <ul style="list-style-type: none"> <li>오픈 소스 기반의 소프트웨어 플랫폼을 제공하는 서비스 유형</li> <li>공급자는 오픈 소스를 기반으로 하는 빅데이터 처리 프로그램을 공급</li> <li>소비자는 공급자가 제공하는 클라우드 서비스를 통해 빅데이터 처리 서비스를 이용</li> </ul>	<b>분석 소프트웨어-플랫폼 유형</b> <ul style="list-style-type: none"> <li>일반 소비자를 위한 분석 소프트웨어를 제공하는 서비스 유형</li> <li>빅데이터를 솔루션으로 상품화하고 클라우드 컴퓨팅과 결합하여 제공</li> <li>소비자는 자체 서버와 솔루션을 구축하는 대신에 클라우드 컴퓨팅 인프라에서 데이터를 저장 및 분석하는 프로그램을 이용할 수 있음</li> </ul>
애플리케이션			<b>분석 소프트웨어-애플리케이션 유형</b> <ul style="list-style-type: none"> <li>고객 맞춤형 솔루션 서비스 유형으로 데이터의 의미를 파악하고 이를 분석해서 활용하는 서비스를 제공</li> <li>축적된 데이터를 바탕으로 분석 후 결과의 의미를 파악/제공</li> <li>소비자의 검색 패턴을 이용해 독감 확산을 예측했던 구글 분석이 대표적 사례</li> </ul>

## 2. 빅데이터 분석방법과 접근법

### ⌚ 분석 방법

#### ■ 분석 목적에 따른 구분

구분	내용
통계 분석	<ul style="list-style-type: none"> <li>통계 기법에 의한 분석 방법으로 가장 대표적인 유형</li> </ul>
예측 분석	<ul style="list-style-type: none"> <li>과거의 데이터와 변수 간의 관계를 이용하여 새로운 변수를 추정</li> </ul>
데이터 마이닝 분석	<ul style="list-style-type: none"> <li>많은 데이터 속에 숨겨진 유용한 패턴을 추출하여 분류, 군집, 연관, 이상 탐지 분석 등을 수행</li> </ul>
최적화 분석	<ul style="list-style-type: none"> <li>주어진 제한 조건을 만족하면서 목적 함수를 최대화 또는 최소화하는 방법을 찾는다</li> </ul>



## 2. 빅데이터 분석방법과 접근법

### ⌚ 분석 접근법

#### 하향식 접근법

- 문제 해결 방법을 찾기 위해 필요한 데이터를 수집 및 분석하는 방식
- 문제 해결을 위해 근본 원인을 파악하고 분석 과제를 도출한 뒤 해결 방안을 도출
- 도출된 해결 방안에 대한 실현 가능성과 우선순위를 결정하기 위해 데이터를 수집, 가공, 분석하는 접근법
- 분석 과제를 도출하기 위해 '수요 기반 분석 과제 도출 방식'을 사용
- 데이터 분석은 문제 해결을 가능하게 하는 실행 동인 역할

#### 상향식 접근법

- 현재 보유하고 있는 데이터를 분석하여 의미 있는 관계나 패턴을 찾아 지식을 발견하고 문제를 해결하는 방식
- 정형 데이터는 물론이고 다양한 원천의 비정형 데이터를 조합 하고 시각화를 통해 의미 있는 패턴을 파악한 뒤 이를 적용하여 문제를 해결하는 데이터 기반의 접근
- 분석 과제를 도출하기 위해 '데이터 주도 분석 과제 도출 방식'을 사용

#### 프로토타이핑 접근법

- 빅데이터 환경의 불확실성을 고려한 방식
- 소비자의 요구 사항이나 데이터를 규정하기가 어렵고 데이터 원천도 명확히 파악하기 어려운 경우 사용
- 일단 프로토타입을 만들어 분석을 시도한 뒤 결과를 확인하고 개선하고 이를 반복

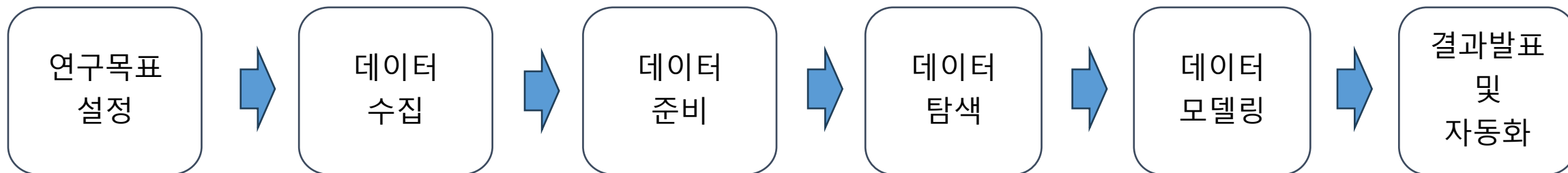
# 3. 데이터 과학 방법론

## ⌚ 데이터 과학

### ■ 의의

- 빅데이터를 다루고 그 안에서 가치를 도출하는 과정이 데이터 과학임
- 여기서 사용할 빅데이터 분석 프로젝트에 적용할 방법론은, 하향식접근법과 프로토타이핑 접근법을 융합한 것임
- 구조적이고 체계적인 단계 수행과 반복적인 모델 구축 작업을 통해 프로젝트 성공율을 높임

### ■ 데이터 과학 방법론의 6단계



# 3. 데이터 과학 방법론

## ⌚ 연구 목표 설정

### ■ 의의

- 프로젝트와 관련된 모든 참여자가 연구 목표를 함께 정의하고 산출물과 일정 등의 계획에 합의한 뒤 프로젝트 헌장 작성

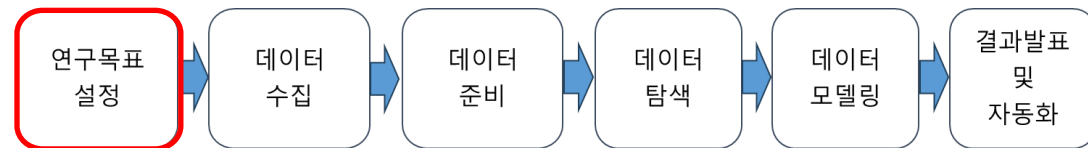
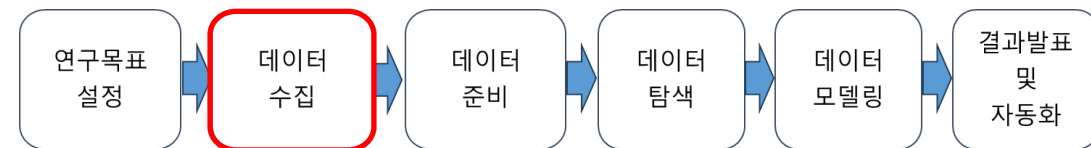


표 3-1 프로젝트 헌장 예시

프로젝트 헌장(Project Charter)													
프로젝트 명 (Project Name)													
프로젝트 설명 (Project Description)													
프로젝트 매니저 (Project Manager, PM)		승인 날짜 (Date Approved)											
프로젝트 스폰서 (Project Sponsor)		서명 (Signature)											
비즈니스 케이스(Business Case)		목표(Goals) / 산출물(Deliverables)											
<div>팀 구성원(Team Member)</div> <table border="1"> <thead> <tr> <th>이름(Name)</th> <th>역할(Role)</th> </tr> </thead> <tbody> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> <tr><td> </td><td> </td></tr> </tbody> </table>		이름(Name)	역할(Role)										
		이름(Name)	역할(Role)										
위험과 제약사항(Risk and Constraints)		주요 일정(Milestones)											

# 3. 데이터 과학 방법론



## ⌚ 데이터 수집

### ■ 데이터의 위치와 형태 확인 및 원시 데이터를 수집

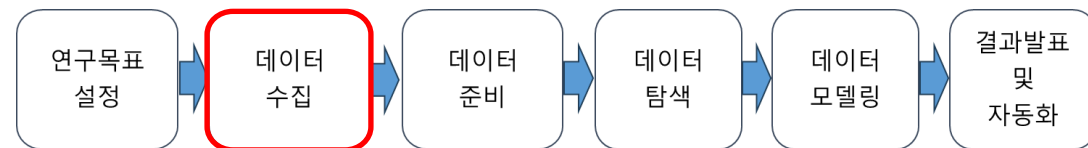
- 필요한 데이터를 수집할 때는 이미 가지고 있는 내부 데이터베이스나 데이터 저장소를 이용
- 외부에서 수집하는 경우 다양한 수집 기술을 활용할 수 있음
- 수집할 데이터의 유형과 종류를 파악한 뒤 그에 맞는 수집 기술을 선택해서 사용

사이트	설명
<a href="http://data.go.kr">http://data.go.kr</a>	한국 정부에서 제공하는 공공데이터
<a href="http://kostat.go.kr">http://kostat.go.kr</a>	한국 통계청에서 공개하는 데이터
<a href="http://opendata.hira.or.kr">http://opendata.hira.or.kr</a>	한국 보건 의료 빅데이터 개방 시스템
<a href="http://www.localdata.kr">http://www.localdata.kr</a>	한국 지방행정 인허가 데이터
<a href="https://www.mcst.go.kr">https://www.mcst.go.kr</a>	한국 문화체육관광부 문화 데이터

사이트	설명
<a href="http://data.seoul.go.kr">http://data.seoul.go.kr</a>	서울시 열린데이터 광장
<a href="https://data.gg.go.kr">https://data.gg.go.kr</a>	경기도 공공데이터 개방 포털
<a href="http://data.gov">http://data.gov</a>	미국 정부의 공공데이터
<a href="http://data.worldbank.org">http://data.worldbank.org</a>	세계 은행에서 제공하는 개방 데이터
<a href="http://open.fda.gov">http://open.fda.gov</a>	미국 식약청의 개방 데이터

# 3. 데이터 과학 방법론

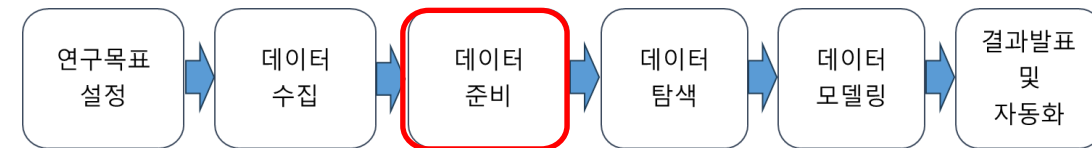
## ⌚ 데이터 수집



수집 기술	설명	수집 데이터
크롤링	• SNS, 뉴스, 웹 정보처럼 인터넷에서 제공하는 데이터를 수집할 수 있다.	웹 추출 데이터
FTP	• TCP/IP 프로토콜을 활용하는 인터넷 서버에서 각종 파일을 송수신할 수 있다. • 보안을 강화하려면 SFTP 사용을 고려해야 한다. • 서버 간 연동시에는 전용 네트워크 구축을 고려해야 한다.	파일
Open API	• 서비스, 데이터 등을 어디서나 쉽게 이용하도록 개방된 API로 데이터 수집 방식을 제공한다. • 다양한 애플리케이션을 개발할 수 있도록 개발자와 소비자에게 공개되어 있다.	실시간 수집 데이터
RSS	• 웹 기반의 최신 정보를 공유하기 위한 XML 기반의 콘텐츠 배급 프로토콜이다.	XML 기반 웹 콘텐츠
스트리밍	• 인터넷에서 실시간으로 음성/오디오/비디오 데이터를 수집하는 기술이다.	음성/오디오/비디오의 실시간 수집 데이터
로그 수집기	• 웹 서버 로그, 웹 로그, 트랜잭션 로그, 클릭 로그, DB 로그 등 각종 로그 데이터를 수집하는 오픈 소스 기술이다. • Chukwa, Flume, Scribe 등이 있다.	로그
RDB 수집기	• 관계형 데이터베이스에서 정형 데이터를 수집한 뒤 HDFS(하둡 분산 파일 시스템)나 HBase와 같은 NoSQL에 저장하는 오픈 소스 기술이다. • Sqoop, Direct JDBC/ODBC 등이 있다.	RDB 기반 데이터

유형	종류	수집 기술
정형 데이터	RDB, 스프레드시트	ETL, FTP, Open API
반정형 데이터	HTML, XML, JSON, 웹 문서, 웹 로그, 센서 데이터	크롤링, RSS, Open API, FTP
비정형 데이터	소셜 데이터, 문서(워드, 한글), 이미지, 오디오, 비디오, IoT	크롤링, RSS, Open API, 스트리밍, FTP

# 3. 데이터 과학 방법론



## ⌚ 데이터 준비

### ■ 의의

- 수집한 원시 데이터의 품질을 높이기 위해 정제 후 사용 가능한 형태로 가공하는 단계
- 수집한 데이터를 다음 단계에서 사용할 수 있게 오류를 여과 하거나 수정하여 정제
- 필요에 따라서는 데이터를 통합하거나 형태를 변환

종류	설명
데이터 여과	• 오류 발견, 보정, 삭제, 중복성 확인 등의 과정을 통해 데이터 품질을 향상시킨다.
데이터 정제	• 결측치는 채워 넣고 이상치는 식별 또는 제거하고 잡음이 섞인 데이터는 평활화하여 데이터 불일치성을 교정한다.
데이터 통합	• 데이터 분석이 용이하도록 유사 데이터 및 연계가 필요한 데이터(또는 데이터베이스)를 통합한다.
데이터 축소	• 분석 시간을 단축하기 위해 분석에 사용하지 않는 항목은 제거한다.
데이터 변환	• 데이터 분석에 용이한 형태로 데이터 유형을 변환한다. • 정규화normalization, 집합화aggregation, 요약summarization, 계층 생성 등의 방법을 활용한다. • ETLExtraction, Transformation, Loading 도구를 제공한다.



# 3. 데이터 과학 방법론

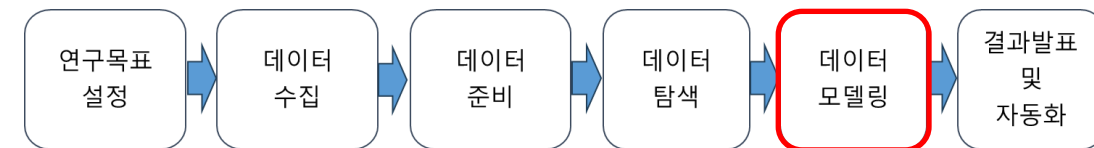


## ⌚ 데이터 탐색

### ■ 의의

- 데이터와 변수 간의 관계나 상호 작용을 이해하기 위한 단계
- 변수 간의 관련성, 데이터의 분포, 편차, 패턴 존재 여부를 확인하는 탐색적 데이터 분석이라고도 함
- 데이터를 쉽게 이해하기 위해 꺾은선 그래프나 히스토그램, 분포도 등과 같은 그래픽 기법을 많이 사용

# 3. 데이터 과학 방법론



## ⌚ 데이터 모델링

### ■ 의의

- 이전 단계에서 얻은 데이터 탐색 결과로 프로젝트에 대한 답을 찾는 단계
- 변수를 선택하여 모델을 구성하고 실행 및 평가하는 과정을 반복 수행하여 문제 해결 모델을 완성
- 이때 분석하려는 데이터의 특성과 목적에 따라 모델 유형을 선택할 수 있음

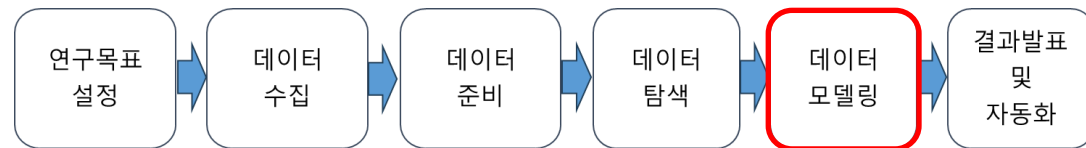
### ■ 데이터 분석 모델의 종류

유형	종류 및 설명
텍스트 마이닝 모델	텍스트 기반의 데이터로부터 새로운 정보를 발견할 수 있도록 정보 검색, 추출, 체계화, 분석을 모두 포함하는 텍스트 처리 과정 및 기법이다.
소셜 네트워크 분석 모델	언어 분석 기반의 정보 추출을 통해 대용량의 소셜 미디어 데이터에서 이슈를 탐지하고 시간 경과에 따라 이슈가 유통되는 전체 과정을 모니터링하고 향후 추이를 분석하는 기법이다.

# 3. 데이터 과학 방법론

## ⌚ 데이터 모델링

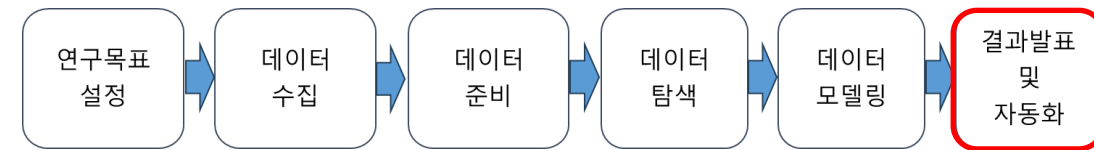
### ■ 데이터 분석 모델의 종류(계속)



유형	종류 및 설명	
통계 분석 모델	전통적인 분석 기법이다. 주로 수치형 데이터에 사용하며 확률을 기반으로 현상을 추정 및 예측한다.	
	기술 통계	대표적인 것으로 평균(산술평균, 중앙값, 최빈값), 분산, 표준편차가 있다.
	상관 분석	두 변수가 어떤 선형적 관계를 가지는지 분석하는 기법이다. 두 변수는 서로 독립적 관계일 수도 있고 상관된 관계일 수 있는데 이러한 관계의 강도를 상관관계라고 한다.
	회귀 분석	연속형 변수에 대해 독립 변수와 종속 변수 사이의 상관관계에 따른 수학적 모델인 선형적 관계식을 구하여 어떤 독립 변수가 주어졌을 때 이에 따른 종속 변수를 예측하거나 수학적 모델이 얼마나 잘 설명하고 있는지를 판별하기 위한 적합도를 측정하는 분석 기법이다.
	분산 분석	두 개 이상 다수의 집단을 비교할 때 집단 내의 분산, 총평균과 각 집단의 평균의 차이로 생긴 집단 간 분산의 비교를 통해 만들어진 F분포로 가설을 검증하는 기법이다.
	주성분 분석	다양한 변수를 분석하는 다변량 분석으로 많은 변수로부터 몇 개의 주성분을 추출하는 기법이다. 이때 주성분 분석은 차원 축소를 위한 것이다.

유형	종류 및 설명	
데이터 마이닝 모델	패턴 인식, AI, 머신러닝, 딥러닝 등을 이용하여 대용량 데이터에 숨겨진 데이터 간의 상호 관련성 및 유용한 정보를 추출하는 기법이다.	
	예측	대용량 데이터 집합 내의 패턴을 기반으로 미래를 예측한다(예: 수요 예측).
	분류	일정한 집단에 대해 특정한 정의로 분류 및 구분을 추론한다.
	군집화	구체적인 특성을 공유하는 자료를 분류한다. 미리 정의된 특성에 대한 정보를 가지지 않는다는 점에서 분류와 다르다(예: 유사 행동 집단의 구분).
	패턴 분석	동시에 발생한 사건 간의 상호연관성을 탐색한다(예: 장바구니 속 상품의 관계).
	순차 패턴 분석	연관 규칙에 시간 개념을 반영하여 시계열에 따른 패턴의 상호연관성을 탐색한다(예: 금융 상품 사용을 위한 반복 방문).

# 3. 데이터 과학 방법론



## ⌚ 결과 발표 및 분석 자동화

### ■ 의의

- 프로젝트 수행 결과가 연구 목표를 달성했는지를 이해 당사자, 특히 의사 결정자에게 이해시키고 가능하다면 이후의 유사 프로젝트 수행을 위해 분석 과정을 자동화하는 단계
- [연구목표설정]에서 작성한 프로젝트 헌장에 명시된 목표를 달성했는지 산출물이 제대로 작성되었는지, 일정과 예산은 계획대로 진행되었는지 여부를 확인
- 모든 참여자를 대상으로 분석 결과를 발표
- 분석 과정을 재사용할 수 있도록 자동화

## 4. Flask를 활용한 웹서버 구축

### Flask

#### ■ Flask 란

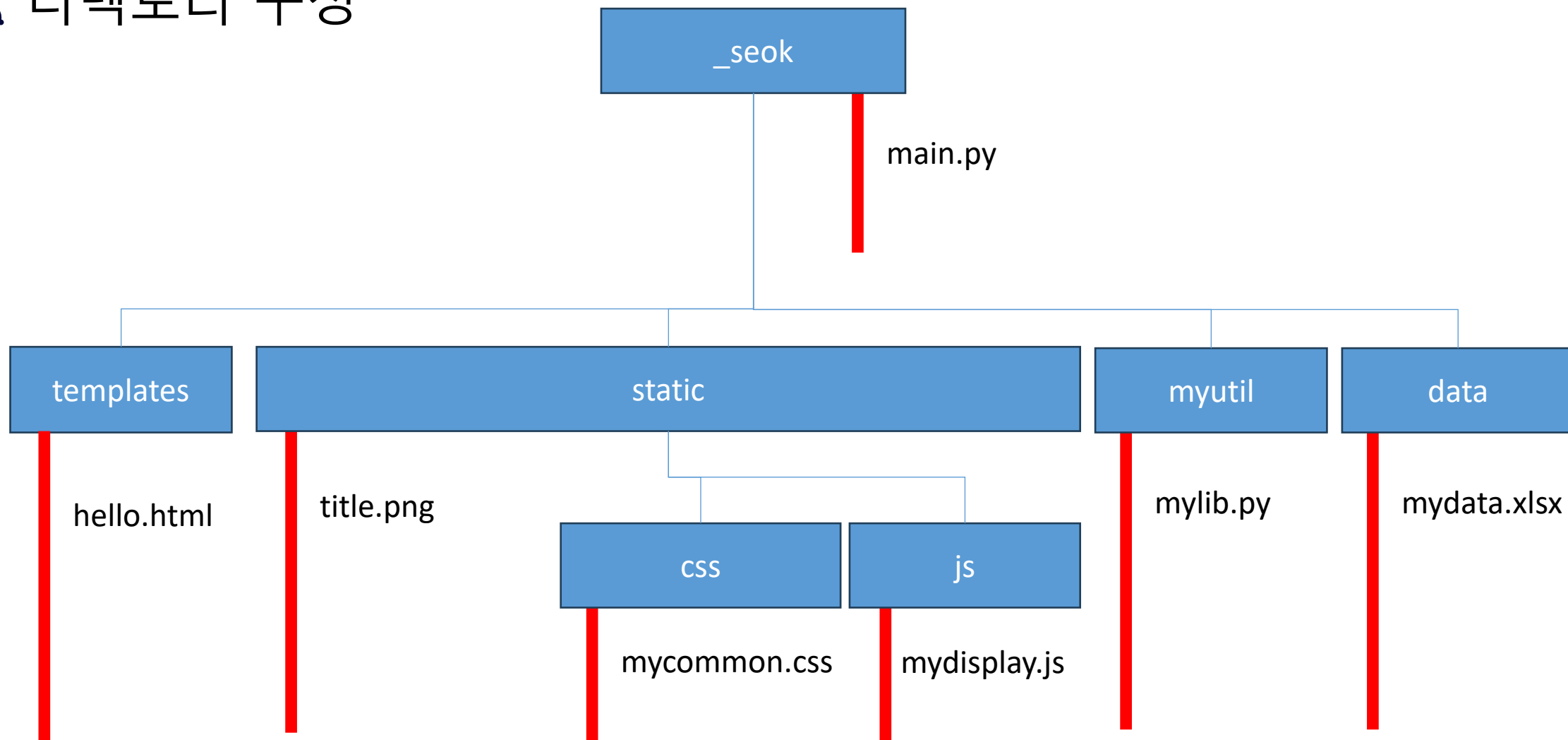
- Python 의 마이크로 웹 프레임워크
- 다양한 웹 엔진과 붙여서 사용할 수 있고, 가볍게 운영
- 비교적 코드가 단순하고 특히, 관련된 확장기능이 많아 API 서버를 만들기에 편리
- Django 와 같은 풀 스택 프레임워크에 비해, 개발자의 능력과 목적에 맞게 커스텀이 가능

#### ■ Flask 설치

```
python -m pip install --upgrade flask
```

# 4. Flask를 활용한 웹서버 구축

## ⌚ 디렉토리 구성





## 4. Flask를 활용한 웹서버 구축

### ⌚ 디렉토리 생성

- cmd 창을 열고 작업

```

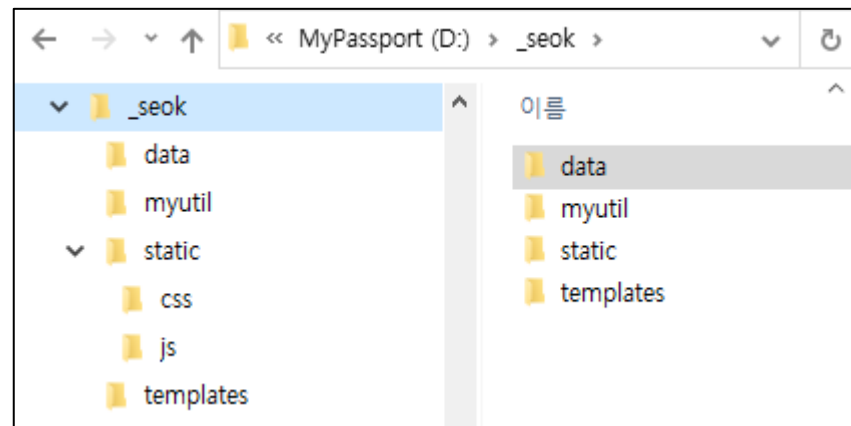
C:\> 명령 프롬프트
C:\Users\dossa> d:
D:\>
D:\>
D:\> mkdir _seok
D:\> cd _seok
D:\_seok> mkdir templates
D:\_seok> mkdir static
D:\_seok> mkdir myutil
D:\_seok> mkdir data
D:\_seok> cd static
D:\_seok\static> mkdir css
D:\_seok\static> mkdir js
    
```

d:  
mkdir \_seok  
cd \_seok

mkdir templates  
mkdir static  
mkdir myutil  
mkdir data

cd static  
mkdir css  
mkdir js

탐색기에서 확인



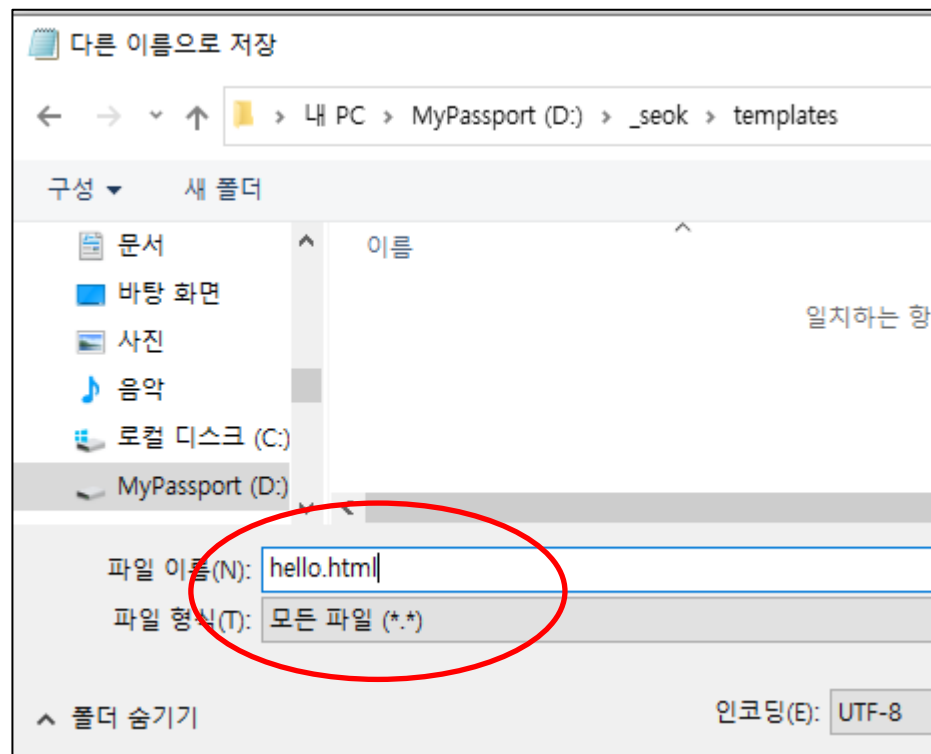
# 4. Flask를 활용한 웹서버 구축

## ⌚ html 파일 생성

### ■ 메모장 창을 열고 작업

```
<!DOCTYPE html>
<html>
<head>
  <title>챗봇 비서</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <script src='/static/js/mydisplay.js'></script>
  <link rel="stylesheet" href="/static/css/mycommon.css">
  <script>
    var a = return_screen_size();
    document.write(a);
  </script>
</head>
<body>
  <div></div>
  <div>엑셀 파일 읽기</div>
  <form action="/get_data" id="input-form" method="post">
    <input type="text" name="input_data" placeholder="hello...">
    <button type="submit">확인</button>
  </form>
</body>
</html>
```

파일 - 다른이름으로저장



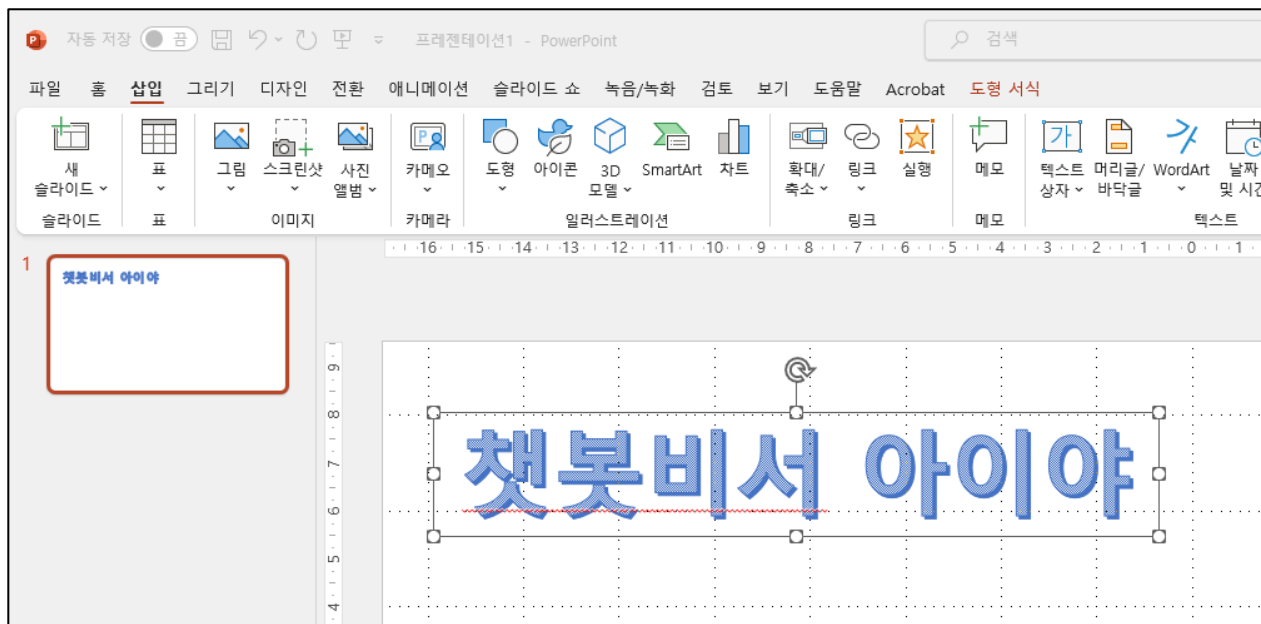
\_seok\templates 밑에 hello.html 로 저장

# 4. Flask를 활용한 웹서버 구축

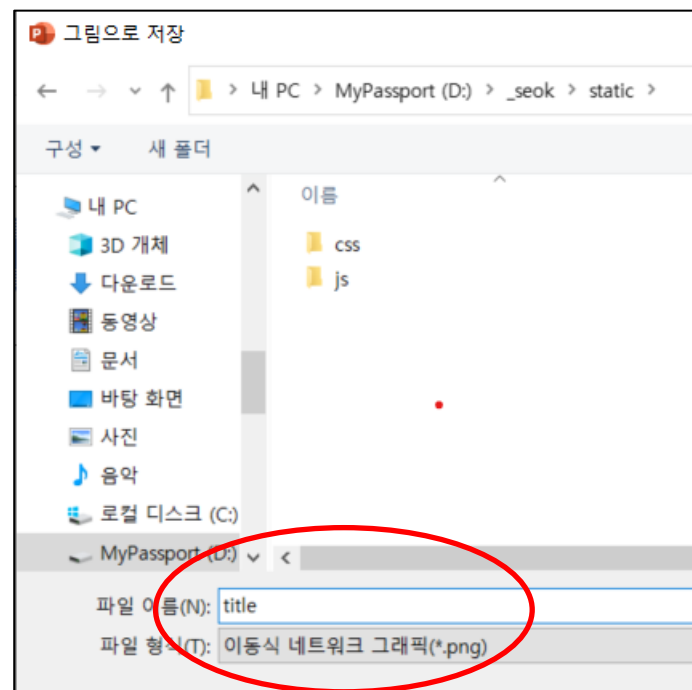
## ⌚ 이미지파일 생성

- 파워포인트 창을 열고 작업

삽입 – WordArt – 글자 입력



글자 박스 클릭 – 오른쪽 마우스 – 그림으로 저장



\_seok\static 밑에 title.png 로 저장

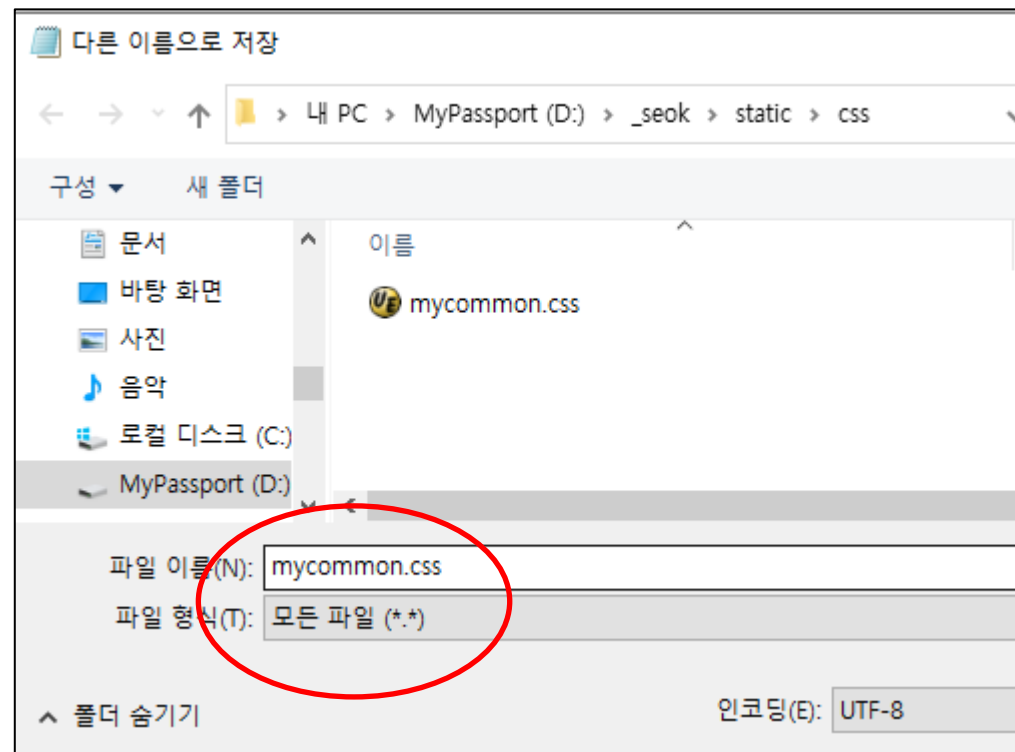
## 4. Flask를 활용한 웹서버 구축

### ⌚ CSS 파일 생성

- 메모장 창을 열고 작업

```
/*=====*/
/* 기본 CSS 설정 */
/*=====*/
body {
    font-family: sans-serif;
    margin: 0;
    padding: 0;
    background-color: #ffc000;
    font-size: 18px;
}
```

파일 - 다른이름으로 저장



\_seok\static\css 밑에 mycommon.css 로 저장

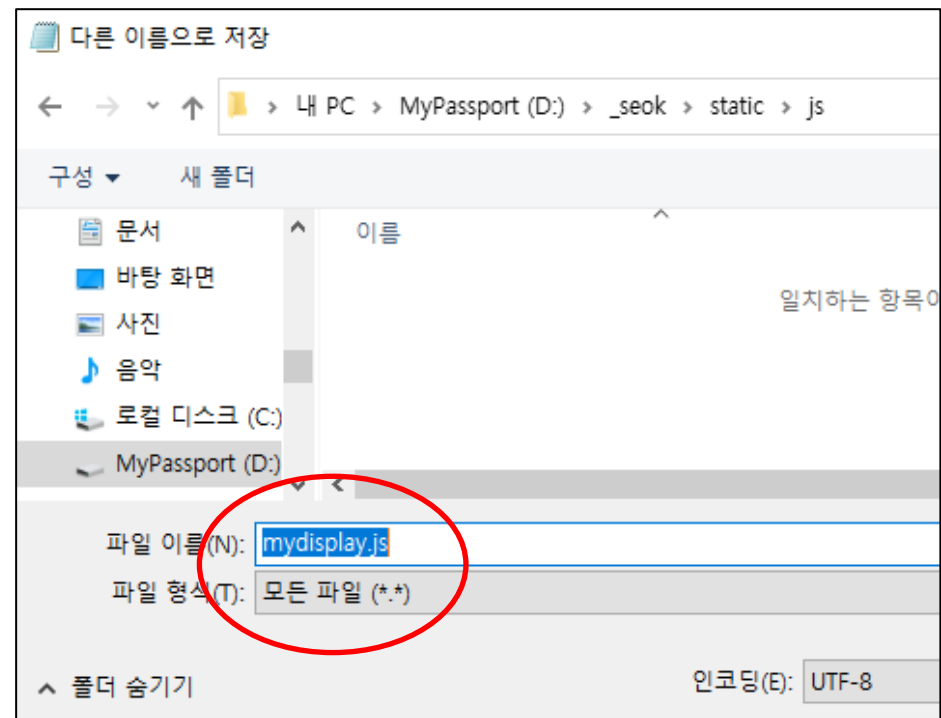
## 4. Flask를 활용한 웹서버 구축

### ⌚ js 파일 생성

- 메모장 창을 열고 작업

```
//=====
// 화면 사이즈 측정
// DelftStack에서 가져옴
//=====
function return_screen_size() {
    // Get the size of the device screen
    var screenWidth = screen.width;
    var screenHeight = screen.height;
    // Get the browser window size
    var windowWidth = window.innerWidth;
    var windowHeight = window.innerHeight;
    // Get the size of the entire webpage
    const scrollWidth = document.documentElement.scrollWidth;
    const scrollHeight = document.documentElement.scrollHeight;
    var str = "Device W:" + screenWidth + ", H:" + screenHeight + ".";
    str += "Browser W: " + windowWidth + ", H:" + windowHeight + ".";
    str += "Scroll W:" + scrollWidth + ", H:" + scrollHeight + ".";
    return str;
}
```

파일 - 다른이름으로저장

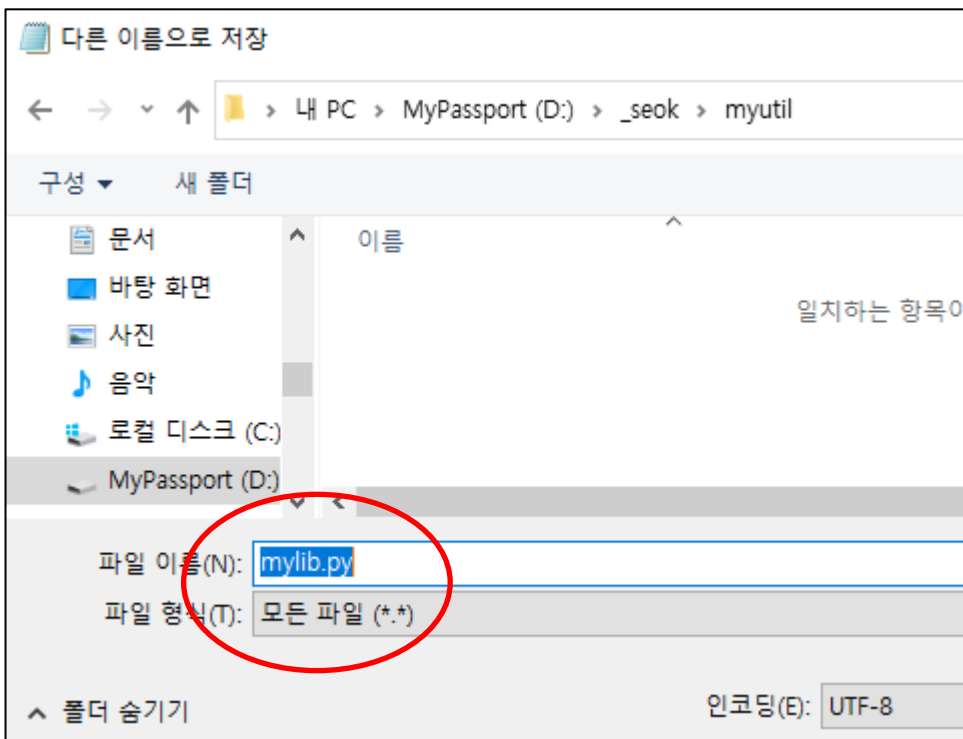


\_seok\static\js 밑에 mydisplay.js 로 저장

## 4. Flask를 활용한 웹서버 구축

### ⌚ 라이브러리 파일 생성

- 메모장 창을 열고 작업



\_seok\myutil 밑에 mylib.py로 저장

4칸 8칸 12칸

```
#=====
# 공통 라이브러리
# 지정한 Excel 파일의 데이터 읽어 오기
#=====
import pandas as pd
def Read_xlsx_Data(_file):
    _df = pd.read_excel(_file)
    _df = _df.fillna("")
    _fields = _df.columns.tolist()
    #print(_file)

    _array = []
    _array.append([])
    for _f in _fields:
        _array[0].append(_f)

    for _f in _fields:
        _index = 1
        for _a in _df[_f]:
            if _f in _fields[0]:
                _array.append([])
                _array[_index].append(_a)
                _index += 1

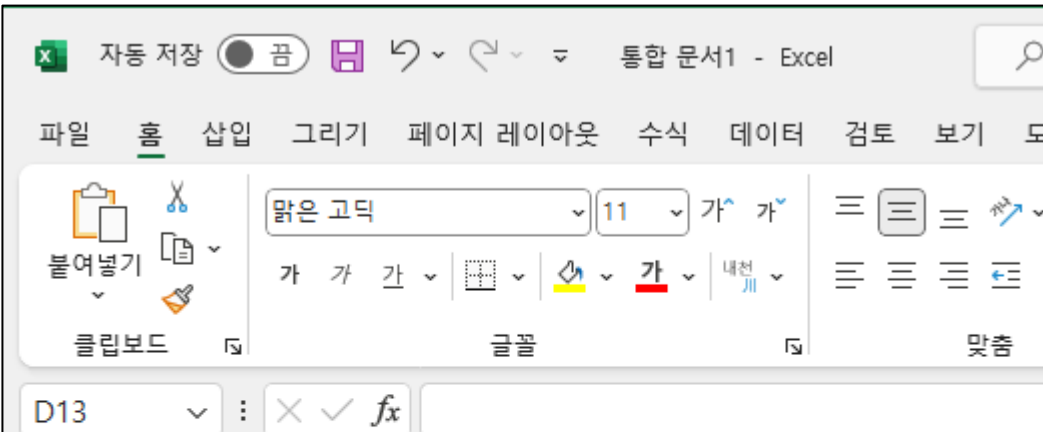
    return _array
```

파일 - 다른이름으로저장

## 4. Flask를 활용한 웹서버 구축

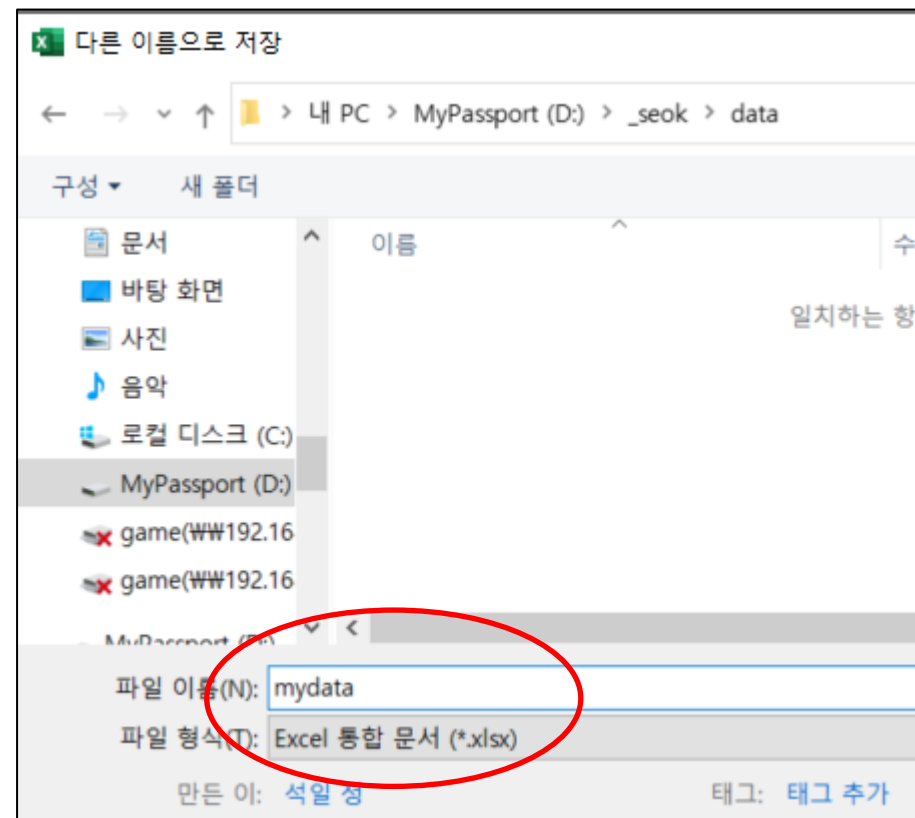
### ⌚ 엑셀 파일 생성

- Excel 창을 열고 작업



	A	B	C	D	E
1	성명	컴퓨터구조	드론실습	빅데이터분석	
2	홍길동	80	100	90	
3	이순신	70	90	80	
4	을지문덕	60	90	90	
5	장화홍련	100	80	100	
6					

파일 - 다른이름으로저장



\_seok\data밑에 mydata.xlsx 로 저장



# 4. Flask를 활용한 웹서버 구축

## ⌚ Flask Main 파일 생성

### ■ 메모장 창을 열고 작업

4칸 8칸

```
#=====
# Flask 웹서버 메인 프로그램
#=====
import socket
import pandas as pd
from flask import Flask, render_template, request
from myutil.mylib import Read_xlsx_Data

app = Flask(__name__)

@app.route('/', methods=['POST', 'GET'])
def home():
    return render_template('hello.html')

@app.route('/get_data', methods=['POST'])
def get_data():
    try:
        input_data = request.form["input_data"]
        print("***input_data : ", input_data)
```

8칸

```
        print("***input_data : ", input_data)
        _file = './data/mydata.xlsx'
        _list = Read_xlsx_Data(_file)
        df = pd.DataFrame(_list[1:], columns=_list[0])
        result = viewPage(df)
        return result
    except Exception as ee:
        print("***error : ", ee)
```

```
def viewPage(df):
    # 역순으로 재정렬 (최신데이터를 맨 위로 올림)
    df = df.sort_index(ascending=False)
    sResult = "<!DOCTYPE html> <html> <head>"
    sResult += "<meta charset='utf-8'> </head> <body>"

    fields = df.columns.tolist()
    sResult += "<table border=1 role='table' bordercolor='green'>"
    sResult += "<thead role='rowgroup'>"
    sResult += "<tr role='row' bgcolor='#b9b922'>"
    for _f in fields:
        sResult += "<th role='columnheader'>" + _f + "</th>"

    sResult += "</tr> </thead>"
    sResult += "<tbody role='rowgroup'>"
    for index, row in df.iterrows():
        sResult += "<tr role='row'>"
```

# 4. Flask를 활용한 웹서버 구축

## ⌚ Flask Main 파일 생성

- 메모장 창을 열고 작업

4칸 8칸 12칸

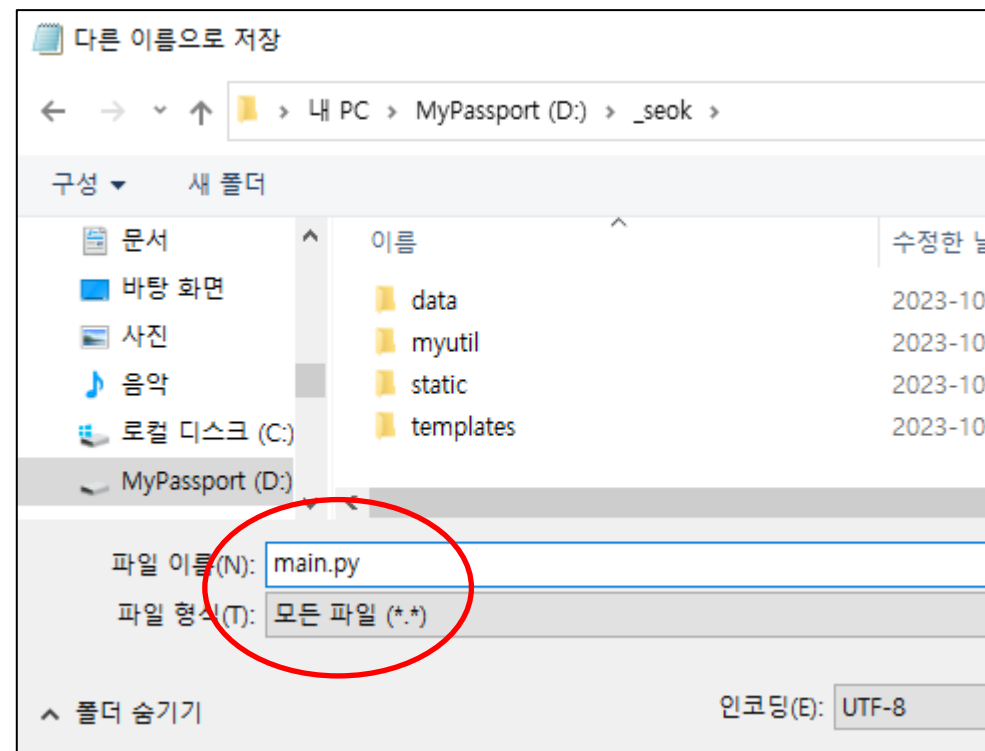
```
for _f in fields:
    if str(row[_f]) == '':
        sResult += "<td role='cell'>-</td>"
    else:
        sResult += "<td role='cell'>" + str(row[_f]) + "</td>"

    sResult += "</tr>"
sResult += "</tbody></table>"
sResult += "</body></html>"

return(sResult)

if __name__ == '__main__':
    _myip = socket.gethostname(socket.gethostname())
    app.run(host=_myip, port=9999, debug=False)
```

파일 - 다른이름으로저장



\_seok 밑에 main.py 로 저장

# 4. Flask를 활용한 웹서버 구축

## ⌚ 웹서버 실행

- cmd 창을 열고 작업

```

C:\> 명령 프롬프트 - python main.py
Microsoft Windows [Version 10.0.19045.3448]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dossa>d:

D:\>cd _seek

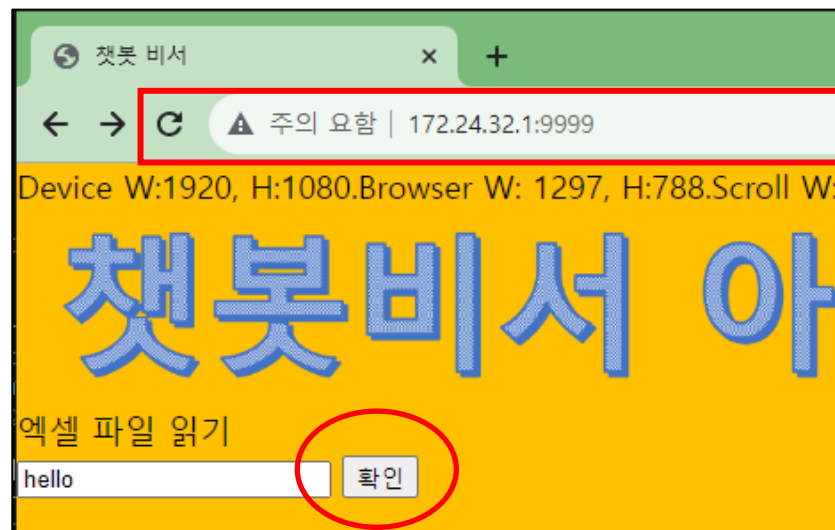
D:\_seek>python main.py
* Serving Flask app 'main'
* Debug mode: off
WARNING: This is a development server. Do not u
* Running on http://172.24.32.1:9999
Press CTRL+C to quit
    
```

d:

cd \_seek

python main.py

여기 주소 입력



성명	컴퓨터구조	드론실습	빅데이터분석
장화홍련	100	80	100
을지문덕	60	90	90
이순신	70	90	80
홍길동	80	100	90

# 참고 자료

- 자바와 파이썬으로 만드는 빅데이터시스템(제이펍, 황세규)
- 위키독스(<https://wikidocs.net/22654>)
- 네이버블로그(<https://blog.naver.com/classmethodkr/222822485338>)
- 데이터분석과 인공지능 활용 (NOSVOS, 데이터분석과인공지능활용편찬위원회 편)

## 참고 사이트

유튜버 : 빅공잼 : <https://www.youtube.com/watch?v=bnYxO2XRCQ0>

네이버 블로그 : 빅공잼

<https://biggongjam.notion.site/3-Hadoop-cd6944182da74edf8d2339b654e0bfb9>

<https://biggongjam.notion.site/4-Spark-2c341ddc8715411484cb2f0254b60126>

Q n A