

빅데이터시스템

BIG DATA ANALYTICS
TECHNOLOGY

데이터분석을 위한 파이썬

목 차

01 파이썬 기초

변수와객체, 자료형과연산자, 제어문

02 파이썬 함수

사용자함수, 내장함수, 모듈과패키지

03 데이터분석용 라이브러리

numpy, pandas, matplotlib

04 Flask 웹서버 수정

디렉토리구성, html파일생성, js파일생성, 라이브러리파일수정,
Flask Main파일생성, 웹서버실행

*. 전수업리뷰

⌚ 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	

*. 전수업리뷰

⌚ 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

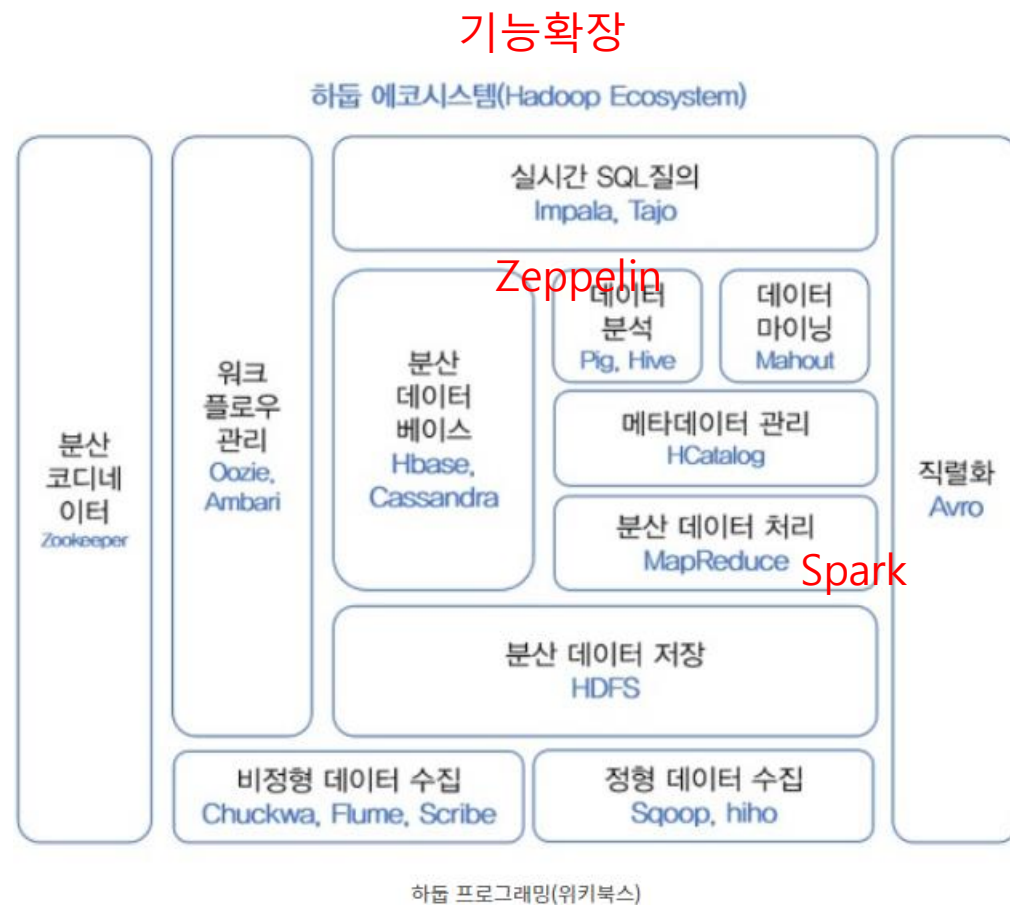
- 대량의 데이터를 저장 및 분석, 처리할 수 있는 대용량의 고속 저장 공간 보유
- 고성능 계산 능력과 실시간으로 발생하는 빅데이터를 처리 및 분석하여 일관성을 유지
- 빅 데이터에서 발생하는 개인 정보를 위한 정보 보안 관리체계 지원도 필요
- 빅데이터 플랫폼은 오픈 소스인 하둡을 근간으로 많이 사용

*. 전수업리뷰

⌚ 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



*. 전수업리뷰

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

*. 전수업리뷰

⌚ 빅데이터 분석 방법

■ 분석 목적에 따른 구분

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

*. 전수업리뷰

⌚ 빅데이터 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

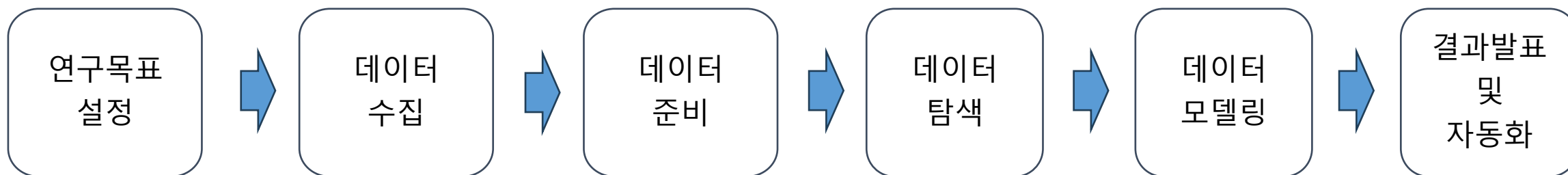
*. 전수업리뷰

⌚ 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



*. 전수업리뷰

⌚ 연구 목표 설정

■ 의의

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

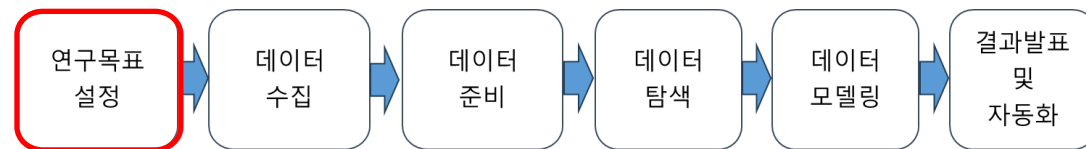
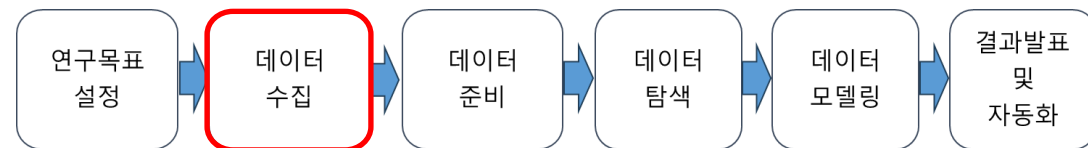


표 3-1 프로젝트 헌장 예시

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)											

*. 전수업리뷰



⌚ 데이터 수집

■ 데이터의 위치와 형태 확인 및 원시 데이터를 수집

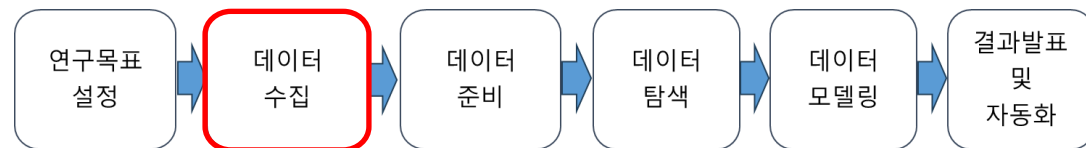
- 필요한 데이터를 수집할 때는 이미 가지고 있는 내부 데이터베이스나 데이터 저장소를 이용
- 외부에서 수집하는 경우 다양한 수집 기술을 활용할 수 있음
- 수집할 데이터의 유형과 종류를 파악한 뒤 그에 맞는 수집 기술을 선택해서 사용

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

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

*. 전수업리뷰

⌚ 데이터 수집



수집 기술	설명	수집 데이터
크롤링	• SNS, 뉴스, 웹 정보처럼 인터넷에서 제공하는 데이터를 수집할 수 있다.	웹 추출 데이터
FTP	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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

*. 전수업리뷰

⌚ 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.

*. 전수업리뷰

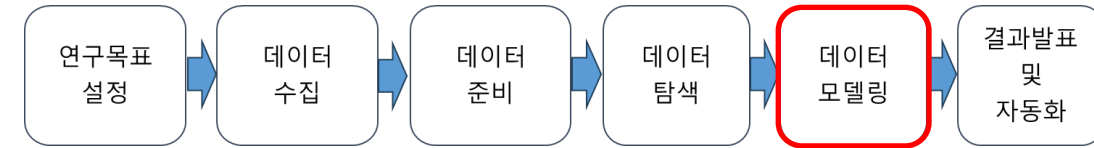


⌚ 데이터 탐색

■ 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

*. 전수업리뷰



⌚ 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

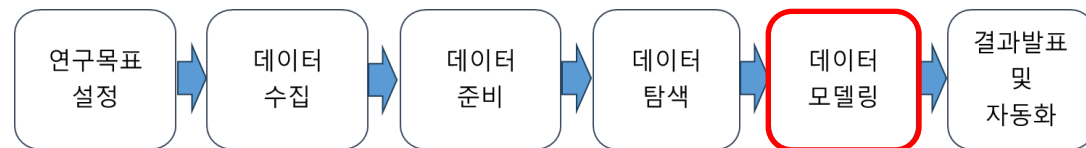
■ 데이터 분석 모델의 종류

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.

*. 전수업리뷰

⌚ 데이터 모델링

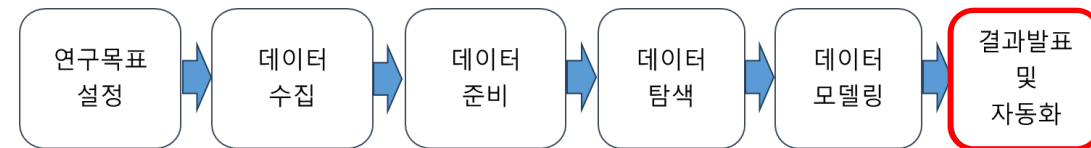
■ 데이터 분석 모델의 종류(계속)



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).

*. 전수업리뷰



⌚ 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 현장에 명시된 목표를 달성했는지 산출물이 제대로 작성되었는지, 일정과 예산은 계획대로 진행되었는지 여부를 확인
- 모든 참여자를 대상으로 분석 결과를 발표
- 분석 과정을 재사용할 수 있도록 자동화

*. 전수업리뷰

Flask

■ Flask 란

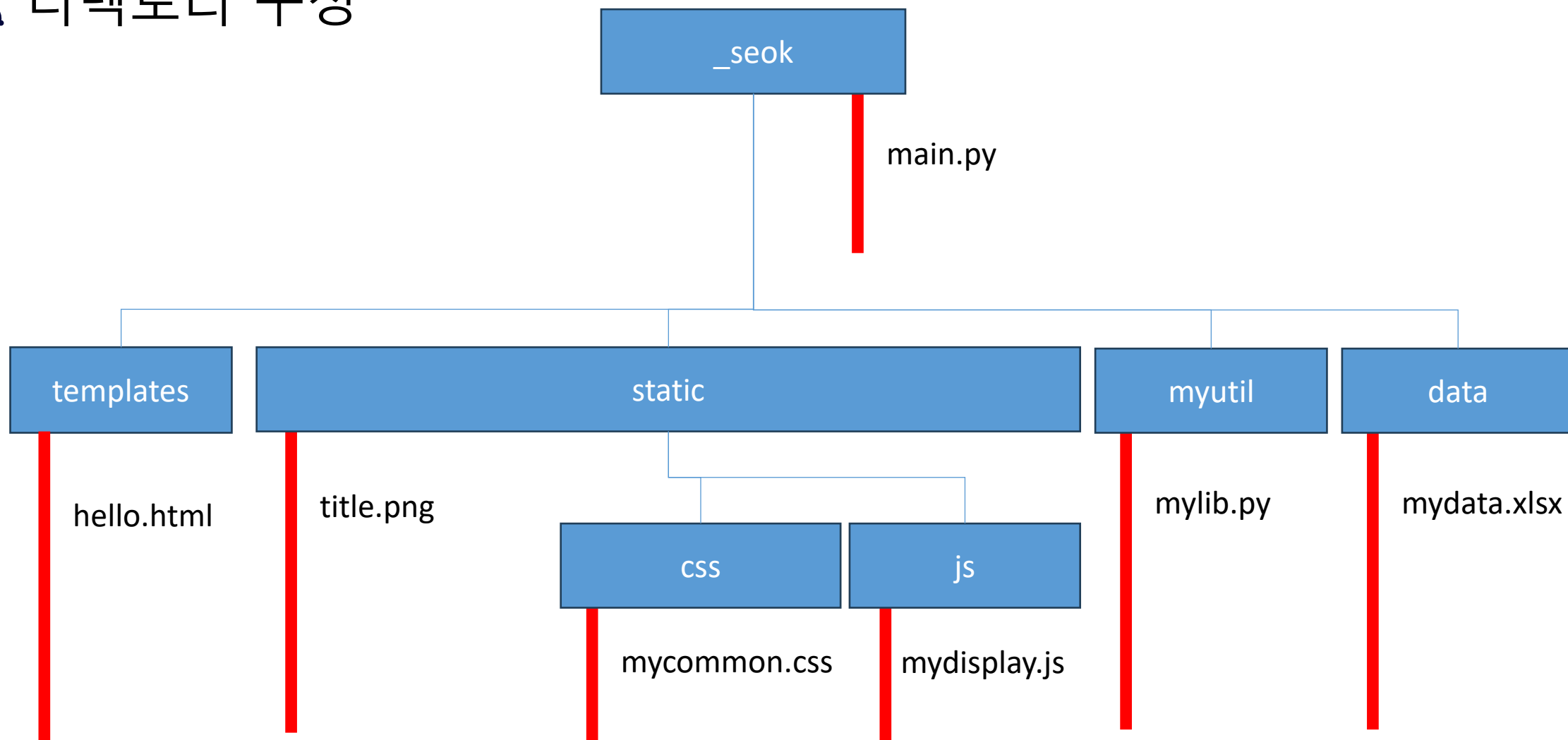
- Python 의 마이크로 웹 프레임워크
- 다양한 웹 엔진과 붙여서 사용할 수 있고, 가볍게 운영
- 비교적 코드가 단순하고 특히, 관련된 확장기능이 많아 API 서버를 만들기에 편리
- Django 와 같은 풀 스택 프레임워크에 비해, 개발자의 능력과 목적에 맞게 커스텀이 가능

■ Flask 설치

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

*. 전수업리뷰

⌚ 디렉토리 구성



*. 전수업리뷰

⌚ 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

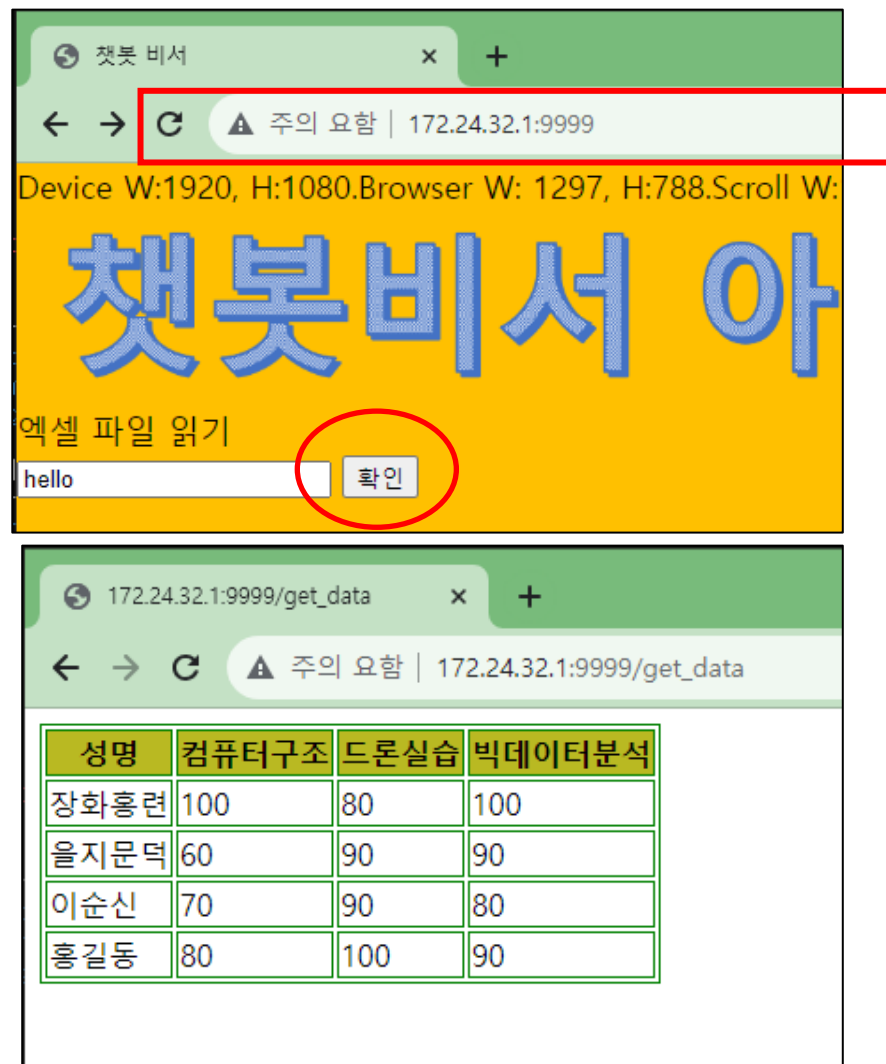
D:\_seek>python main.py
* Serving Flask app 'main'
* Debug mode: off
WARNING: This is a development server. Do not use without proper precautions.
* Running on http://172.24.32.1:9999
Press CTRL+C to quit
    
```

d:

cd _seek

python main.py

여기 주소 입력



*. 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

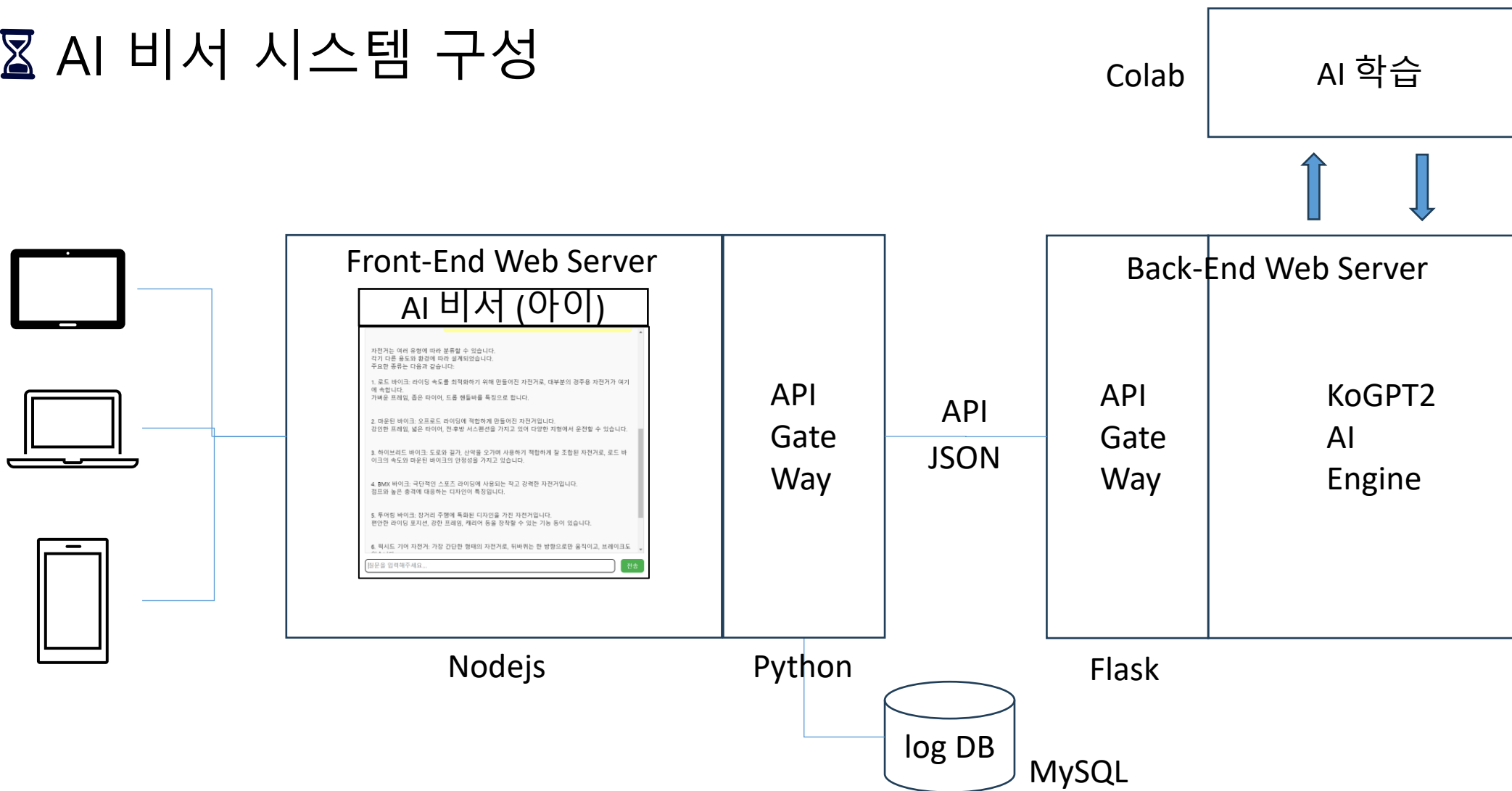
- ✓ 빅데이터 산업의 이해
- ✓ 파이썬 프로그래밍
- ✓ 크롤링
- ✓ 통계분석
- ✓ 텍스트빈도분석
- ✓ 지리정보분석
- ✓ 회귀분석/분류분석
- ✓ 텍스트마이닝

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



*. 전달 사항

⌚ 과제-1 : 나만의 AI 비서 만들기 준비하기

- 챗봇 형태의 AI 비서에게 원하는 것을 질문하면, 응답을 할 수 있도록 학습을 시키자

구분	내용
AI 비서 이름	예) 아이야~
AI 비서 능력	예) 학습을 시키고자 하는 내용 - 중국 음식의 종류를 알고, 음식에 대한 설명을 잘하는 비서 - 우주 행성에 대해 물어 보면, 언제든 줄줄 얘기해 주는 비서 - 우리나라 역사에 대해 모두 설명해 주는 비서 - 지구상에 존재하는 나라와 도시에 대해 모두 알고 있는 비서 - 음식을 만드는 레시피를 상세히 알려 주는 비서
학습시킬 자료를 수집할 곳	예) 크롤링, 백과사전, 구글 검색, ChatGPT에 질의, 책자 읽어 오기
데이터 셋	- 학습시킬 데이터셋은 질문, 응답 형식으로 Excel 에 작성해야 함

prompt	completion
짜장면은 어떤 음식인가요?	중국의 전통적인 면 요리입니다. 두부, 돼지고기, 양파, 대파, 마늘 등을 사용하여 만듭니다.
마라탕은 어떤 음식인가요?	중국의 스타일로 만든 매운탕입니다. 고추, 마라(마라는 중국어로 고추와 겨자를 섞은 것) 등을 사용하여 만듭니다.
홍콩 스타일 딤섬은 어떤 음식인가요?	중국의 전통적인 음식입니다. 소고기, 새우, 돼지고기, 닭고기 등 다양한 재료를 사용하여 만듭니다.
양장피는 어떤 음식인가요?	중국의 전통적인 요리 중 하나입니다. 쇠고기, 돼지고기, 닭고기 등 다양한 재료를 사용하여 만듭니다.
마파두부는 어떤 음식인가요?	중국의 전통적인 요리 중 하나입니다. 두부와 쇠고기, 돼지고기 등 다양한 재료를 사용하여 만듭니다.
칠리 크랩은 어떤 음식인가요?	중국의 전통적인 요리 중 하나입니다. 게살과 칠리 소스를 사용하여 만듭니다.
양꼬치는 어떤 음식인가요?	중국의 전통적인 구이 요리입니다. 양고기와 다양한 채소를 꼬치에 꽂아 구웁니다.
새우 튀김은 어떤 음식인가요?	중국의 전통적인 요리 중 하나입니다. 새우와 부드러운 반죽으로 만들어져 튀겨 집니다.
마판당면은 어떤 음식인가요?	중국의 전통적인 면 요리 중 하나입니다. 당면과 쇠고기, 돼지고기 등 다양한 재료를 사용하여 만듭니다.
탕수육은 어떤 음식인가요?	중국의 전통적인 요리 중 하나
짜장면의 역사는?	짜장면의 역사는 꽤 오래되었으며, 19세기 후반에 만들어졌다고 알려져 있습니다. 짜장면은 두부, 돼지고기, 양파, 대파, 마늘 등을 사용하여 만들어집니다. 짜장면은 중국에서 매우 인기 있는 음식 중 하나이며, 전 세계적으로도 인기가 있습니다.

1. 파이썬 기초

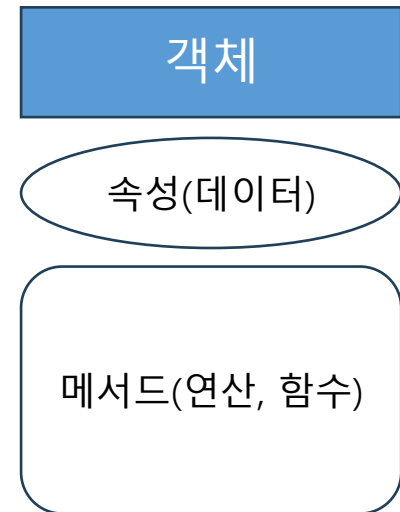
⌚ 변수와 객체

■ 변수

- 값을 저장하는 메모리 공간
- 파이썬에서는 변수를 미리 선언하지 않음
- 변수에 저장해서 사용하는 값의 자료형으로 변수의 자료형이 결정

■ 객체

- 변수 형태의 속성과 함수 형태의 메서드를 가진 것
- 각 객체는 자기의 속성(내부 데이터)과 메서드(내부 연산)를 가짐
- 타 프로그래밍 언어와 달리 파이썬에서는 모든 변수와 자료형이 객체로 되어 있음



1. 파이썬 기초

⌚ 자료형과 연산자

■ 숫자형

```
>>> a = 123          정수
>>> b = 12.34        실수
>>> c = 1 + 2j        복소수(a + bi)
>>> c.real            실수
1.0
>>> c.imag            허수
2.0
>>> c.conjugate()     켄레복소수
(1 - 2j)
>>> abs(c)            절대값
2.23606797749979     8진수
>>> d = 0o12
>>> d
10
>>> e = 0x12A        16진수
>>> e
298
```

■ 논리형

```
>>> l = True
>>> l
True
```

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

C:\> cd \\_seok
C:\> mkdir test
C:\> cd test
C:\> python
Python 3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023)
Type "help", "copyright", "credits" or "license()" for more
>>>
```

■ 연산자

```
>>> a = 3
>>> b = 4
>>> a + b            더하기
7
>>> a - b            빼기
-1
>>> a * b            곱하기
12
>>> a / b            나누기
0.75
>>> a ** b           제곱
81
>>> a % b            나머지
3
>>> a // b           몫
0
```

1. 파이썬 기초

⌚ 자료형과 연산자

■ 문자열 자료형

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
N	o	w		i	s		b	e	t	t	e	r		t	h	a	n		n	e	v	e	r

■ 문자열

```
>>> s1 = 'Hello Python'
>>> s1
'Hello Python'
>>> s2 = "Hello Python"
>>> s2
'Hello Python'
>>> s3 = '''Hello Python'''
>>> s3
'Hello Python'
>>> s4 = """"Hello Python""""
>>> s4
'Hello Python'
```

■ 연산자

```
>>> head = "Python"
>>> tail = " is fun"
>>> head + tail
'Python is fun'
>>> head * 2
'PythonPython'
>>> print("=" * 5)
=====
>>> a = "Now is better than never"
>>> a[0]
'N'
>>> a[-1]
'r'
```

```
>>> b = a[0] + a[1] + a[2]
>>> b
'Now'
>>> a[0:3]
'Now'
>>> a[:3]
'Now'
>>> a[19:]
'never'
>>> a[:]
'Now is better than never'
```

1. 파이썬 기초

⌚ 자료형과 연산자

■ 문자열 자료형

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[22]	[23]
N	o	w		i	s		b	e	t	t	e	r		t	h	a	n		n	e	v	e	r

■ 연산자(계속)

```
>>> c = "python"
>>> c.count('y')
1
>>> c.find('y')
1
>>> c.find('x')
-1
>>> c.index('y')
1
```

```
>>> b = ","
>>> d = b.join('abcd')
>>> d
'a, b, c, d'
>>> c.upper()
'PYTHON'
>>> c.lower()
'python'
>>> d = " py "
>>> d.lstrip()
'py '
>>> d.rstrip()
' py'
```

```
>>> d.strip()
'py'

>>> a = "Python is difficult."
>>> a.replace("difficult", "funny")
'Python is funny.'
>>> a.split()
['Python', 'is', 'difficult.']
>>> b = "a, b, c, d"
>>> b
'a, b, c, d'
>>> b.split(',')
['a', 'b', 'c', 'd']
```

1. 파이썬 기초

⌚ 자료형과 연산자

■ 리스트 자료형 연산자

```
>>> a = [1, 2, 3]
>>> b = ['Life', 'is', 'too', 'short']
>>> c = [1, 2, 'Life', 'is']
>>> d = [1, 2, [3, 4], ['Life', 'is']]
>>> d[0]
1
>>> d[2]
[3, 4]
>>> d[3][-1]
'is'
>>> d[0:3]
[1, 2, [3, 4]]
>>> a + b
[1, 2, 3, 'Life', 'is', 'too', 'short']
>>> b[0] + " hi~ ^^;"
'Life hi~ ^^;'
```

```
>>> a * 3
[1, 2, 3, 1, 2, 3, 1, 2, 3]
>>> a[2] = 99
>>> a
[1, 2, 99]
>>> a[1:2] = ['a', 'b', 'c']
>>> a
[1, 'a', 'b', 'c', 99]
>>> a[-1] = ['d', 'e', 'f']
>>> a
[1, 'a', 'b', 'c', ['d', 'e', 'f']]
>>> del a[-1]
>>> a
[1, 'a', 'b', 'c']
>>> a.append(5)
>>> a
[1, 'a', 'b', 'c', 5]
```

```
>>> a = [3, 4, 1, 9]
>>> a.reverse()
>>> a
[9, 1, 4, 3]
>>> a.insert(0, 99)
>>> a
[99, 9, 1, 4, 3]
>>> a.remove(99)
>>> a
[9, 1, 4, 3]
>>> a = [2, 1, 0, 2, 3, 2, 4, 2]
>>> a.count(2)
4
```

1. 파이썬 기초

⌚ 자료형과 연산자

■ 튜플 자료형 연산자

```
>>> t1 = (1, )
>>> t2 = (1, 2, 3)
>>> t3 = 1, 2, 3
>>> t4 = (1, 2, (3, 4), ('Life', 'is'))
>>> t4[0]
1
>>> t4[3][-1]
'is'
>>> t4[0:3]
(1, 2, (3, 4))
>>> t1 + t2
(1, 1, 2, 3)
>>> t1 + "hi~"
```

Traceback (most recent call last):

File "<pyshell#157>", line 1, in <module>

```
>>> t2 * 3
(1, 2, 3, 1, 2, 3, 1, 2, 3)
>>> t2[2] = 99
```

Traceback (most recent call last):

File "<pyshell#159>", line 1, in <module>

t2[2] = 99

TypeError: 'tuple' object does not support item assignment

1. 파이썬 기초

⌚ 자료형과 연산자

■ 딕셔너리 자료형 연산자

```
>>> dic = {'name':'Hong', 'phone':'01012345678', 'birth':'0814'}
>>> dic[1] = 'a'
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 1: 'a'}
>>> dic['pet'] = 'dog'
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 1: 'a', 'pet': 'dog'}
>>> del dic[1]
>>> dic
{'name': 'Hong', 'phone': '01012345678', 'birth': '0814', 'pet': 'dog'}
>>> dic['name']
'Hong'
>>> dic.keys()
dict_keys(['name', 'phone', 'birth', 'pet'])
>>> list(dic.keys())
['name', 'phone', 'birth', 'pet']
```

```
>>> dic.values()
dict_values(['Hong', '01012345678', '0814', 'dog'])
>>> list(dic.values())
['Hong', '01012345678', '0814', 'dog']
>>> dic.items()
dict_items([('name', 'Hong'), ('phone', '01012345678'), ('birth', '0814'), ('pet', 'dog')])
>>> dic.clear()
>>> dic
{}

```

1. 파이썬 기초

⌚ 자료형과 연산자

■ 집합 자료형 연산자

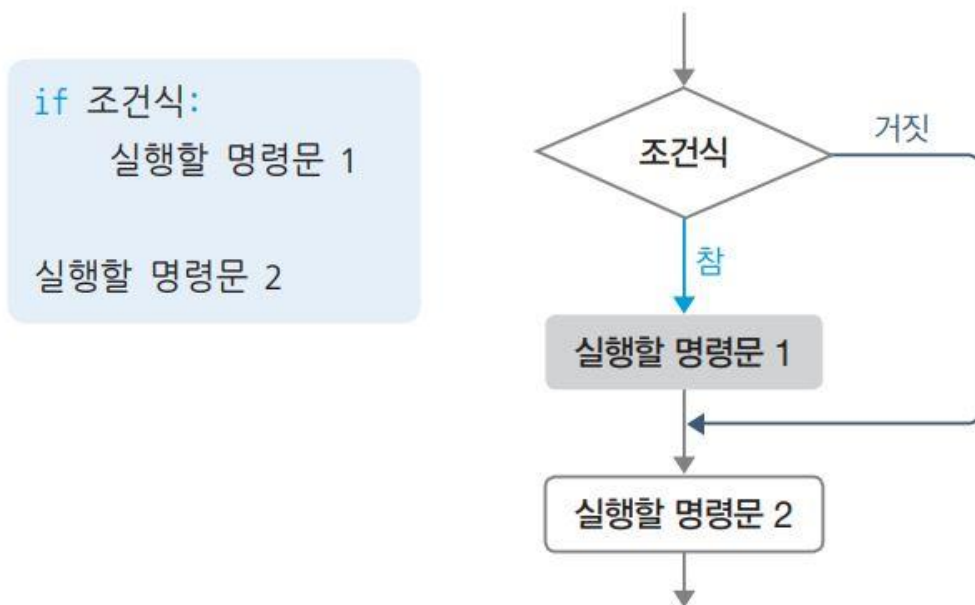
```
>>> s1 = {1, 2, 'a', 5}
>>> s2 = set([1, 2, 3, 4, 5, 6])
>>> s2
{1, 2, 3, 4, 5, 6}
>>> s3 = set([4, 5, 6, 7, 8, 9])
>>> s3
{4, 5, 6, 7, 8, 9}
>>> s2 & s3
{4, 5, 6}
>>> s2.intersection(s3)
{4, 5, 6}
>>> s2 | s3
{1, 2, 3, 4, 5, 6, 7, 8, 9}
>>> s2.union(s3)
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

```
>>> s2 - s3
{1, 2, 3}
>>> s2.difference(s3)
{1, 2, 3}
>>> s2.add(7)
>>> s2
{1, 2, 3, 4, 5, 6, 7}
>>> s2.update([6, 7, 8, 9, 10])
>>> s2
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
>>> s2.remove(7)
>>> s2
{1, 2, 3, 4, 5, 6, 8, 9, 10}
```

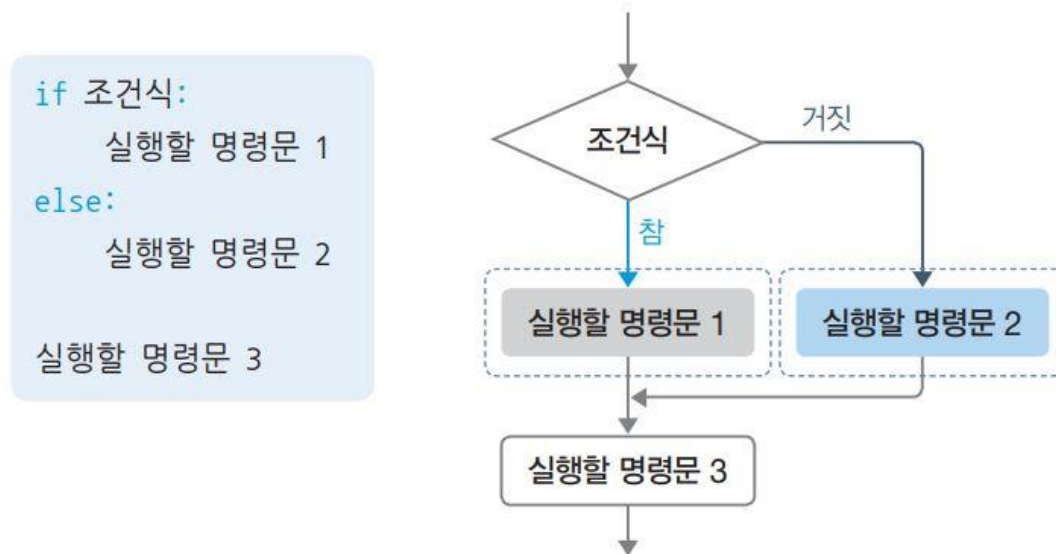
1. 파이썬 기초

⌚ 제어문

▪ if문



▪ if-else문

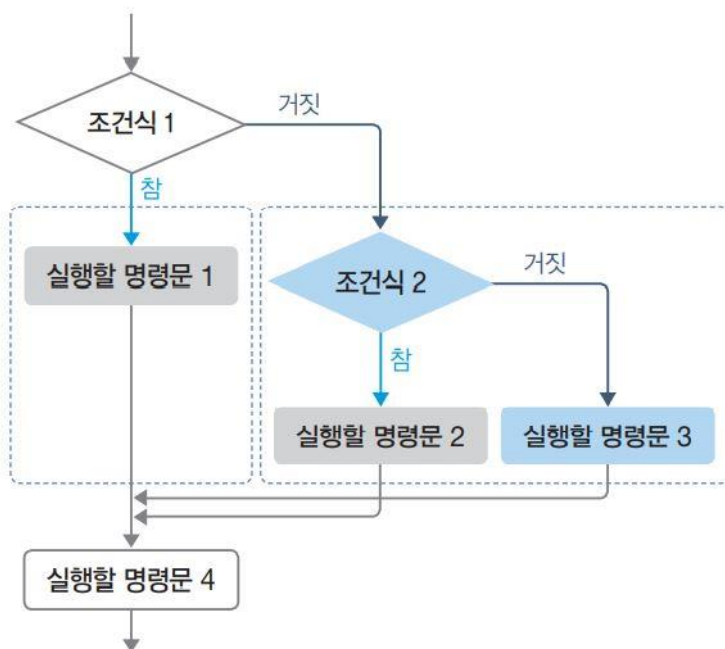


1. 파이썬 기초

⌚ 제어문

■ if-elif-else문

```
if 조건식 1:
    실행할 명령문 1
elif 조건식 2:
    실행할 명령문 2
else:
    실행할 명령문 3
실행할 명령문 4
```



■ 조건 연산자

```
>>> x = 3
>>> y = 2
>>> x == y
False
>>> x != y
True
>>> x >= y
True
>>> x in [1, 2, 3]
True
>>> x not in [1, 2, 3]
False
>>> money = 1300
>>> if money >= 1200:
        print("OK")
```

OK

1. 파이썬 기초

⌚ 제어문

■ for문

```
>>> list = ['one', 'two', 'three']
>>> for i in list:
    x = i + '!'
    print(x)
```

```
one!
two!
three!
```

■ while문

```
>>> i = 0
>>> while i < 5:
    i += 1
    print('*' * i)
```

```
*
**
***
****
*****
```

2. 파이썬 함수

사용자함수

■ 함수 정의

```
>>> def sum1(a, b):
    x = a + b
    return x
```

```
>>> def sum2(*args):
    x = 0
    for i in args:
        x += i
    return x
```

■ 함수 호출

```
>>> a = 5
>>> b = 3
>>> sum1(a, b)
8
>>> sum1(3, 5)
8
>>> sum2(1, 2, 3, 4, 5)
15
>>> sum2(2, 3.5, 10)
15.5
```


2. 파이썬 함수

⌚ 내장함수

■ 함수 종류

```
>>> abs(-3.5)
3.5
>>> chr(97)
'a'
>>> divmod(7, 3)
(2, 1)
>>> oct(8)
'0o10'
>>> hex(16)
'0x10'
>>> a = 3
>>> id(a)
1728080976
>>> int('3')
3
```

```
>>> str(3)
'3'
>>> list("Python")
['P', 'y', 't', 'h', 'o', 'n']
>>> list((1, 2, 3))
[1, 2, 3]
>>> tuple("Python")
('P', 'y', 't', 'h', 'o', 'n')
>>> tuple([1, 2, 3])
(1, 2, 3)
>>> type("abc")
<class 'str'>
>>> type(a)
<class 'int'>
```

```
>>> sum = lambda a,b: a+b
>>> sum
<function <lambda> at 0x000002C826BABA0>
>>> sum(3, 5)
8
>>> max([1, 4, 2, 8, 6])
8
>>> max("Python")
'y'
>>> min([1, 4, 2, 8, 6])
1
>>> pow(2, 4)
16
>>> c = input()
21
>>> c
'21'
>>> list(range (5, 10))
[5, 6, 7, 8, 9]
>>> list(range (5, 10, 2))
[5, 7, 9]
>>> len('Python')
6
```

2. 파이썬 함수

⌚ 모듈과 패키지

■ 선언과 불러오기

```
>>> Request('http://www.naver.com')
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    Request('http://www.hanb.co.kr')
NameError: name 'Request' is not defined
>>> import urllib.request
>>> urllib.request.Request('http://www.naver.com')
<urllib.request.Request object at 0x000001E5E0AE8390>
>>> import pandas
>>> pandas.DataFrame()
Empty DataFrame
Columns: []
Index: []
>>> from datetime import datetime
>>> datetime.now()
datetime.datetime(2023, 10, 11, 20, 26, 2, 559119)
```

2. 파이썬 함수

⌚ 모듈과 패키지

■ 파일 쓰기

```
>>> f = open("test1.txt", 'w')
>>> f
<_io.TextIOWrapper name='test1.txt' mode='w'
encoding='cp949'>
>>> f.close()
>>> f = open("test2.txt", 'w')
>>> for i in range(1, 6):
    data = ("%d번째 줄입니다. \n" % i)
    f.write(data)
<Enter>
>>> f.close()

^Z
dir test*
```

■ 파일 읽기

```
>>> f = open("test2.txt", 'r')
>>> line = f.readline()
>>> print(line)
1번째 줄입니다.
>>> while True:
    line = f.readline()
    if not line: break
    print(line)
<Enter>
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
.....
>>> f.close()
```

3. 데이터분석용 라이브러리

⌚ numpy

```
>>> import numpy as np
>>> np.__version__
'1.23.5'
>>> ar1 = np.array([1, 2, 3, 4, 5])
>>> ar1
array([1, 2, 3, 4, 5])
>>> type(ar1)
<class 'numpy.ndarray'>
>>> ar2 = np.array([[10, 20, 30], [40, 50, 60]])
>>> ar2
array([[10, 20, 30],
       [40, 50, 60]])
>>> ar3 = np.arange(1, 11, 2)
>>> ar3
array([1, 3, 5, 7, 9])
>>> ar4 = np.array([1, 2, 3, 4, 5, 6]).reshape((3, 2))
```

```
>>> ar4
array([[1, 2],
       [3, 4],
       [5, 6]])
>>> ar5 = np.zeros((2, 3))
>>> ar5
array([[0., 0., 0.],
       [0., 0., 0.]])
>>> ar6 = ar2[0:2, 0:2]
>>> ar6
array([[10, 20],
       [40, 50]])
>>> ar7 = ar2[0, :]
>>> ar7
array([10, 20, 30])
>>> ar8 = ar1 + 10
>>> ar8
array([11, 12, 13, 14, 15])
```

```
>>> ar1 + ar8
array([12, 14, 16, 18, 20])
>>> ar8 - ar1
array([10, 10, 10, 10, 10])
>>> ar1 * 2
array([ 2, 4, 6, 8, 10])
>>> ar1 / 2
array([0.5, 1. , 1.5, 2. , 2.5])
>>> ar9 = np.dot(ar2, ar4)
>>> ar9
array([[220, 280],
       [490, 640]])
```

```
[[10, 20, 30],
 [40, 50, 60]]
```

```
[[1, 2],
 [3, 4],
 [5, 6]]
```

3. 데이터분석용 라이브러리

pandas

■ Series 자료형

```
>>> import pandas as pd
>>> pd.__version__
'1.0.3'
>>> data1 = [10, 20, 30, 40, 50]
>>> data1
[10, 20, 30, 40, 50]
>>> data2 = ['1반', '2반', '3반', '4반', '5반']
>>> data2
['1반', '2반', '3반', '4반', '5반']
>>> sr1 = pd.Series(data1)
>>> sr1
0    10
1    20
2    30
3    40
4    50
dtype: int64
```

```
>>> sr2 = pd.Series(data2)
>>> sr2
0    1반
1    2반
2    3반
3    4반
4    5반
dtype: object
>>> sr3 = pd.Series([101, 102, 103, 104, 105])
>>> sr3
0    101
1    102
2    103
3    104
4    105
dtype: int64
>>> sr4 = pd.Series(['월', '화', '수', '목', '금'])
>>> sr4
0    월
1    화
2    수
3    목
4    금
dtype: object
```

3. 데이터분석용 라이브러리

pandas

■ Series 자료형

```
>>> sr5 = pd.Series(data1, index = [1000,
1001, 1002, 1003, 1004])
```

```
>>> sr5
1000  10
1001  20
1002  30
1003  40
1004  50
dtype: int64
```

```
>>> sr6 = pd.Series(data1, index = data2)
```

```
>>> sr6
1반   10
2반   20
3반   30
4반   40
5반   50
dtype: int64
```

```
>>> sr7 = pd.Series(data2, index = data1)
```

```
>>> sr7
10  1반
20  2반
30  3반
40  4반
50  5반
dtype: object
```

```
>>> sr8 = pd.Series(data2, index = sr4)
```

```
>>> sr8
월   1반
화   2반
수   3반
목   4반
금   5반
dtype: object
```

```
>>> sr8[2]
```

```
'3반'
```

```
>>> sr8['수']
```

```
'3반'
```

```
>>> sr8[-1]
```

```
'5반'
```


3. 데이터분석용 라이브러리

pandas

■ Series 자료형

```
>>> sr8[0:4]
월 1반
화 2반
수 3반
목 4반
dtype: object>>> sr8.index
Index(['월', '화', '수', '목', '금'], dtype =
'object')
>>> sr8.values
array(['1반', '2반', '3반', '4반', '5반'], dtype =
object)
```

```
>>> sr1 + sr3
0 111
1 122
2 133
3 144
4 155
dtype: int64
>>> sr4 + sr2
0 월1반
1 화2반
2 수3반
3 목4반
4 금5반
dtype: object
```

3. 데이터분석용 라이브러리

pandas

■ DataFrame 자료형

```
>>> data_dic = {
    'year': [2021, 2022, 2023],
    'sales': [350, 480, 1099]
}
>>> data_dic
{'year': [2021, 2022, 2023], 'sales': [350, 380, 1099]}
>>> df1 = pd.DataFrame(data_dic)
>>> df1
   year  sales
0  2021    350
1  2022    380
2  2023   1099
>>> df2 = pd.DataFrame([[89.2, 92.5, 90.8], [92.8, 89.9, 95.2]],
index = ['중간고사', '기말고사'], columns = data2[0:3])
```

```
>>> df2
      1반    2반    3반
중간고사 89.2   92.5   90.8
기말고사 92.8   89.9   95.2
>>> data_list = [['20231101', 'Hong', '90', '95'],
['20231102', 'Kim', '93', '94'], ['20231103', 'Lee', '87', '97']]
>>> df3 = pd.DataFrame(data_list)
>>> df3
      0      1      2      3
0  20231101  Hong    90    95
1  20231102   Kim    93    94
2  20231103   Lee    87    97
>>> df3.columns = ['학번', '이름', '중간고사', '기말고사']
>>> df3
      학번      이름  중간고사  기말고사
0  20231101  Hong      90      95
1  20231102   Kim      93      94
2  20231103   Lee      87      97
```

3. 데이터분석용 라이브러리

⌚ pandas

■ DataFrame 자료형

```
>>> df3.head(2)
```

	학번	이름	중간고사	기말고사
0	20231101	Hong	90	95
1	20231102	Kim	93	94

```
>>> df3.tail(2)
```

	학번	이름	중간고사	기말고사
1	20231102	Kim	93	94
2	20231103	Lee	87	97

```
>>> df3['이름']
```

```
0 Hong
1 Kim
2 Lee
```

```
Name: 이름, dtype: object
```

```
C:\_seok\test>type score.csv
,0,1,2,3
0,20231101,Hong,90,95
1,20231102,Kim,93,94
2,20231103,Lee,87,97
C:\_seok\test>
```

```
>>> df3.to_csv('./score.csv', header = 'False')
```

```
>>> df4 = pd.read_csv('./score.csv', encoding='utf-8', index_col=0)
```

```
>>> df4
```

	학번	이름	중간고사	기말고사
0	20231101	Hong	90	95
1	20231102	Kim	93	94
2	20231103	Lee	87	97

3. 데이터분석용 라이브러리

⌚ matplotlib

■ 라인플롯 차트 그리기

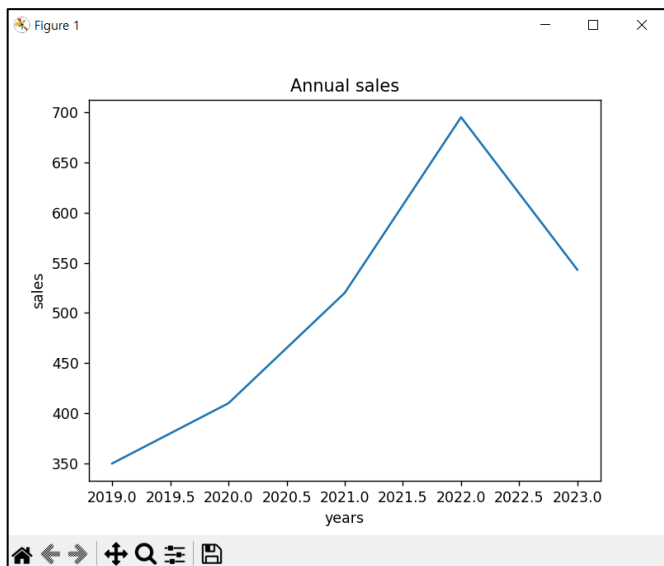
`python -m pip install matplotlib`

```
>>> import matplotlib
```

```
>>> matplotlib.__version__
```

'3.8.0'

```
>>> import matplotlib.pyplot as plt
```



1. 데이터 준비

```
>>> x = [2019, 2020, 2021, 2022, 2023]
```

```
>>> y = [350, 410, 520, 695, 543]
```

2. x축과 y축 데이터를 지정하여 라인플롯 생성

```
>>> plt.plot(x, y, color='blue')
```

[<matplotlib.lines.Line2D object at 0x0000015DB82D58C8>]

3. 차트 제목 설정

```
>>> plt.title('Annual sales')
```

Text(0.5, 1.0, 'Annual sales')

4. x축 레이블 설정

```
>>> plt.xlabel('years')
```

Text(0.5, 0, 'years')

5. y축 레이블 설정

```
>>> plt.ylabel('sales')
```

Text(0, 0.5, 'sales')

6. 라인플롯 표시

```
>>> plt.show()
```

3. 데이터분석용 라이브러리

⌚ matplotlib

■ 바차트 그리기

1. 데이터 준비

```
>>> y1 = [350, 410, 520, 695]
>>> y2 = [200, 250, 385, 350]
>>> x = range(len(y1))
```

2. x축과 y축 데이터를 지정하여 라인플롯 생성

```
>>> plt.bar(x, y1, width = 0.7, color = "blue")
<BarContainer object of 4 artists>
>>> plt.bar(x, y2, width = 0.7, color = "red", bottom = y1)
<BarContainer object of 4 artists>
```

3. 차트 제목 설정

```
>>> plt.title('Quarterly sales')
Text(0.5, 1.0, 'Quarterly sales')
```

4. x축 레이블 설정

```
>>> plt.xlabel('Quarters')
Text(0.5, 0, 'Quarters')
```

5. y축 레이블 설정

```
>>> plt.ylabel('sales')
Text(0, 0.5, 'sales')
```

6. 눈금 이름 리스트 생성

```
>>> xLabel = ['first', 'second', 'third', 'fourth']
```

7. 바 차트의 x축 눈금 이름 설정

```
>>> plt.xticks(x, xLabel, fontsize = 10)
([<matplotlib.axis.XTick object at
0x0000015DB5722B48>, <matplotlib.axis.XTick
object at 0x0000015DB5722B08>, <matplotlib.
axis.XTick object at 0x0000015DB82E2688>,
<matplotlib.axis.XTick object at
0x0000015DB60C5188>], [Text(0, 0, 'first'),
Text(0, 0, 'second'), Text(0, 0, 'third'), Text(0,
0, 'fourth')])
```

3. 데이터분석용 라이브러리

⌚ matplotlib

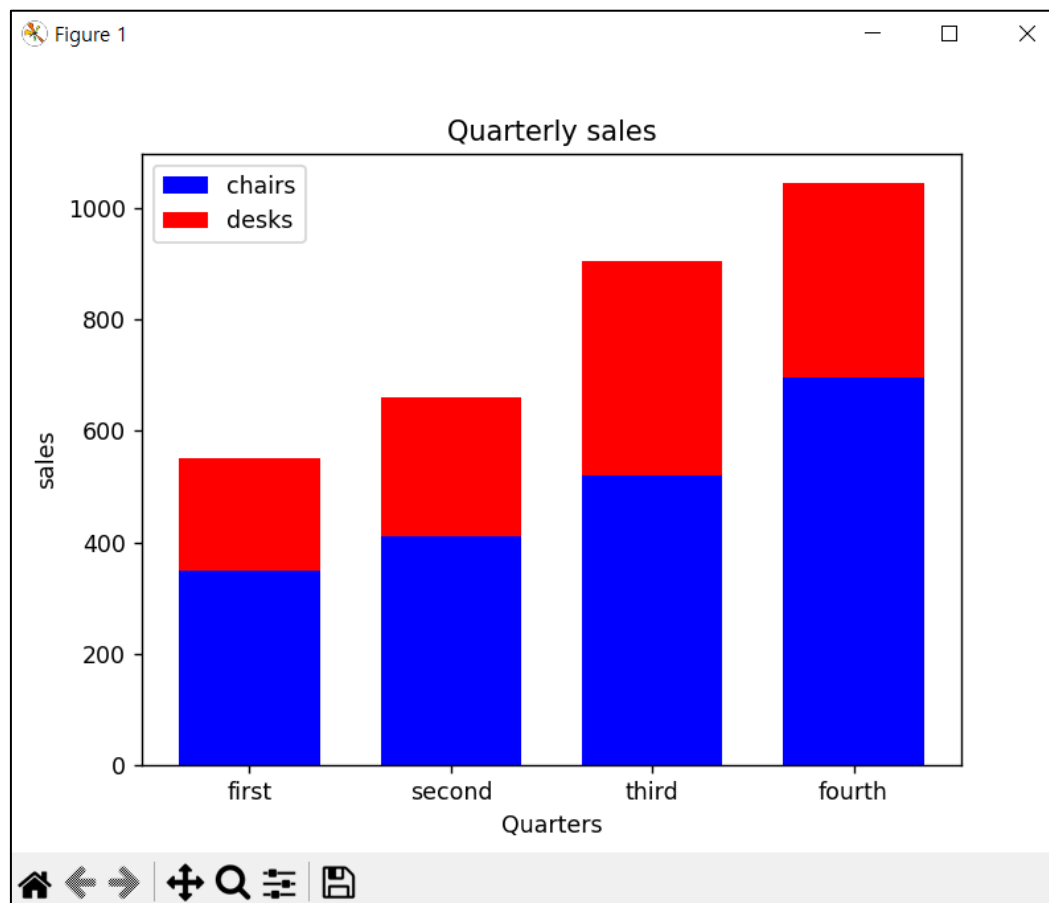
■ 바차트 그리기

8. 범례 설정

```
>>> plt.legend(['chairs', 'desks'])  
<matplotlib.legend.Legend object at  
0x0000020F2BBA0908>
```

9. 바 차트 표시

```
>>> plt.show()
```



* 연습문제-1

⌚ 구구단

```

C:\_seok\test>python gugudan.py
2 x 1 = 2   3 x 1 = 3   4 x 1 = 4   5 x 1 = 5
2 x 2 = 4   3 x 2 = 6   4 x 2 = 8   5 x 2 = 10
2 x 3 = 6   3 x 3 = 9   4 x 3 = 12  5 x 3 = 15
2 x 4 = 8   3 x 4 = 12  4 x 4 = 16  5 x 4 = 20
2 x 5 = 10  3 x 5 = 15  4 x 5 = 20  5 x 5 = 25
2 x 6 = 12  3 x 6 = 18  4 x 6 = 24  5 x 6 = 30
2 x 7 = 14  3 x 7 = 21  4 x 7 = 28  5 x 7 = 35
2 x 8 = 16  3 x 8 = 24  4 x 8 = 32  5 x 8 = 40
2 x 9 = 18  3 x 9 = 27  4 x 9 = 36  5 x 9 = 45

6 x 1 = 6   7 x 1 = 7   8 x 1 = 8   9 x 1 = 9
6 x 2 = 12  7 x 2 = 14  8 x 2 = 16  9 x 2 = 18
6 x 3 = 18  7 x 3 = 21  8 x 3 = 24  9 x 3 = 27
6 x 4 = 24  7 x 4 = 28  8 x 4 = 32  9 x 4 = 36
6 x 5 = 30  7 x 5 = 35  8 x 5 = 40  9 x 5 = 45
6 x 6 = 36  7 x 6 = 42  8 x 6 = 48  9 x 6 = 54
6 x 7 = 42  7 x 7 = 49  8 x 7 = 56  9 x 7 = 63
6 x 8 = 48  7 x 8 = 56  8 x 8 = 64  9 x 8 = 72
6 x 9 = 54  7 x 9 = 63  8 x 9 = 72  9 x 9 = 81

```


* 연습문제-2

⌚ 데이터를 csv 로 저장

아래의 자료를 sales.csv 에 저장

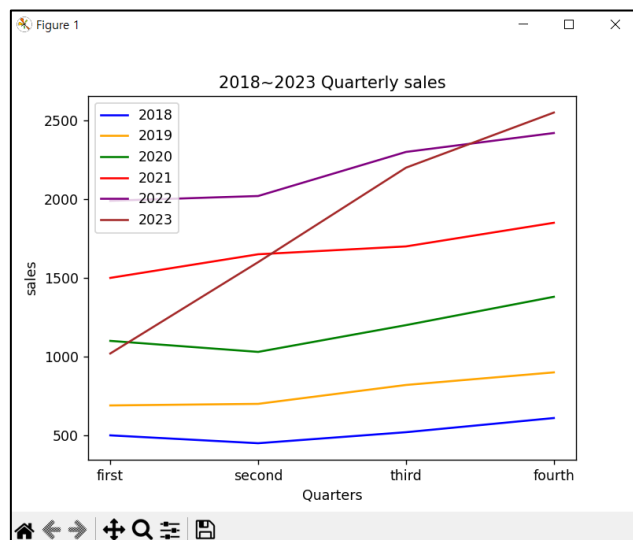
	first	second	third	fourth
2018	500	450	520	610
2019	690	700	820	900
2020	1100	1030	1200	1380
2021	1500	1650	1700	1850
2022	1990	2020	2300	2420
2023	1020	1600	2200	2550

* 연습문제-3

⌚ 라인플롯 그리기

아래의 자료를 그래프로 그리기

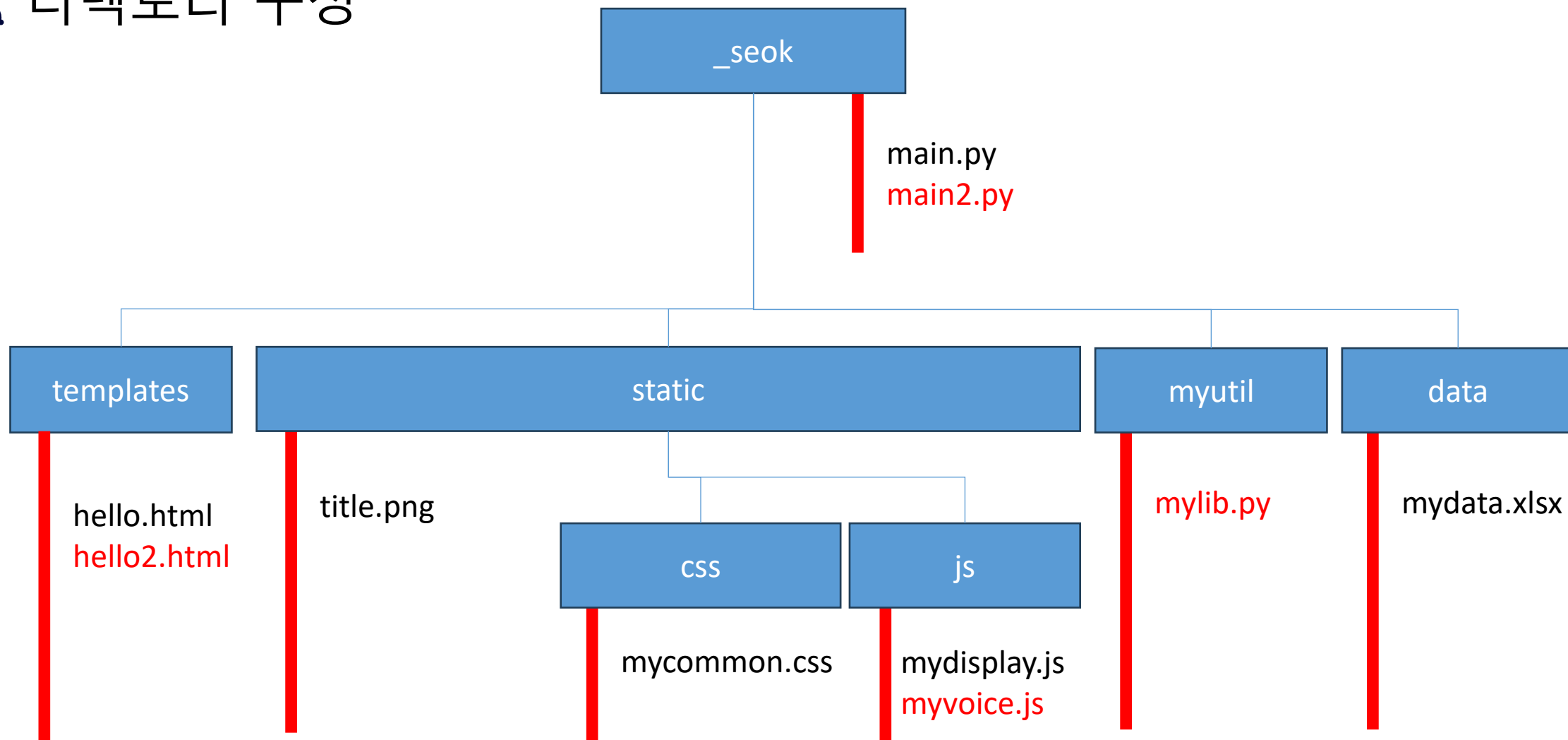
	first	second	third	fourth
2018	500	450	520	610
2019	690	700	820	900
2020	1100	1030	1200	1380
2021	1500	1650	1700	1850
2022	1990	2020	2300	2420
2023	1020	1600	2200	2550



4. Flask 웹서버 수정

⌚ 디렉토리 구성

※스마트폰으로 접속할 수 있도록 소켓 수정
음성인식 모듈 추가
라이브러리 추가
음성인식용 자바스크립트 추가



4. Flask 웹서버 수정

⌚ 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>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.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">
    <button id="start-btn" type="button"><span class="blind">음성</span></button>
    <input type="text" name="input_data" id="input_data" placeholder="hello...">
    <button type="submit">확인</button>
  </form>
  <script src='/static/js/myvoice.js'></script>
</body>
</html>
```

파일 – 다른이름으로저장

_seok\templates\hello2.html 로 저장

4. Flask 웹서버 수정

⌚ js 파일 생성

■ 메모장 창을 열고 작업

```
//=====
// 음성 인식 및 변환
//=====
var SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
var recognition = new SpeechRecognition();
recognition.lang = 'ko-KR'; // 한국어로 설정
recognition.interimResults = true; // 중간 결과를 반환
recognition.continuous = true; // 연속적인 결과를 반환

var recognitionTimeout; // 음성이 없는 경우를 확인하기 위한 타이머 변수

recognition.onstart = function() {
  console.log('Voice recognition started. Try speaking into the microphone.');
```

\$('#start-btn').addClass('action'); // 음성 인식이 시작되면 class="action" 추가

```
}

recognition.onend = function() {
  $('#start-btn').removeClass('action');
  $('#user-input').val(''); // 음성 인식 종료 후 인풋 필드 리셋
}
```

4. Flask 웹서버 수정

⌚ js 파일 생성

■ 메모장 창을 열고 작업(계속)

```
recognition.onend = function() {
    $('#start-btn').removeClass('action'); // 음성 인식이 종료되면 class="action" 제거
}

recognition.onresult = function(event) {
    var transcript = event.results[event.results.length - 1][0].transcript;
    document.getElementById('input_data').value = transcript;
    clearTimeout(recognitionTimeout); // 결과가 도착하면 타이머를 제거

    recognitionTimeout = setTimeout(() => { // 3초 후에 음성 인식을 중단
        recognition.stop();
    }, 3000);
}

document.querySelector('#start-btn').addEventListener('click', function(e) {
    recognition.start();
});

// 전송 버튼이 클릭되면 음성 인식이 중단되도록 합니다.
document.querySelector('#input-form button[type=submit]').addEventListener('click', function(e) {
    recognition.stop();
});
```

파일 – 다른이름으로저장

_seok\static\js 밑에 myvoice.js 로 저장

4. Flask 웹서버 수정

⌚ 라이브러리 파일 수정

- 메모장 창을 열고 작업

4칸 8칸 12칸

```
#####
# 공통 라이브러리
# 지정한 Excel 파일의 데이터 읽어 오기
#####
import pandas as pd
def mylib_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
```

4. Flask 웹서버 수정

⌚ 라이브러리 파일 수정

■ 메모장 창을 열고 작업

```

4칸   8칸   12칸
#=====  

# 공통 라이브러리  

# 데이터프레임을 웹에 보이기  

#=====  

def mylib_ViewPage(df, input_data):  

    # 역순으로 재정렬 (최신데이터를 맨 위로 올림)  

    df = df.sort_index(ascending=False)  

    sResult = "<!DOCTYPE html><html><head>"  

    sResult += "<meta charset='utf-8'>"  

    sResult += "<meta name='viewport' content='width=device-width, initial-scale=1'>"  

    sResult += "</head><body>"  

    sResult += "<div><a href='/'>Home</a></div>"  

    sResult += "<div>" + input_data + "</div>"  

    fields = df.columns.tolist()  

    sResult += "<table border=1 role='table' bordercolor='green'>"  

    sResult += "<thead role='rowgroup'>"  

    sResult += "<tr role='row' bgcolor='#b9b922'>"

```


4. Flask 웹서버 수정

⌚ 라이브러리 파일 수정

■ 메모장 창을 열고 작업

```

4칸   8칸   12칸
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'>"
    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)

```

파일 - 저장

_seok\myutil 밑에 mylib.py 로 저장

4. Flask 웹서버 수정

⌚ 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 mylib_Read_xlsx_Data, mylib_ViewPage

app = Flask(__name__)

@app.route('/', methods=['POST', 'GET'])
def home():
    return render_template('hello2.html')

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

```

4칸 8칸
        _file = './data/mydata.xlsx'
        _list = mylib_Read_xlsx_Data(_file)
        df = pd.DataFrame(_list[1:], columns=_list[0])
        result = mylib_ViewPage(df, input_data)

        return result

    except Exception as ee:
        print('***error : ', ee)

if __name__ == '__main__':
    #_myip = socket.gethostname(socket.gethostname())
    app.run(host='172.16.11.220', port=9999, debug=False)
    
```

IP 주소는 cmd 창에서 ipconfig 로
무선 IP 주소를 가져와서 등록

파일 - 다른이름으로 저장

_seok 밑에 main2.py 로 저장

4. Flask 웹서버 수정

⌚ 웹서버 실행

- cmd 창을 열고 작업

```
C:\> 명령 프롬프트 - python main2.py

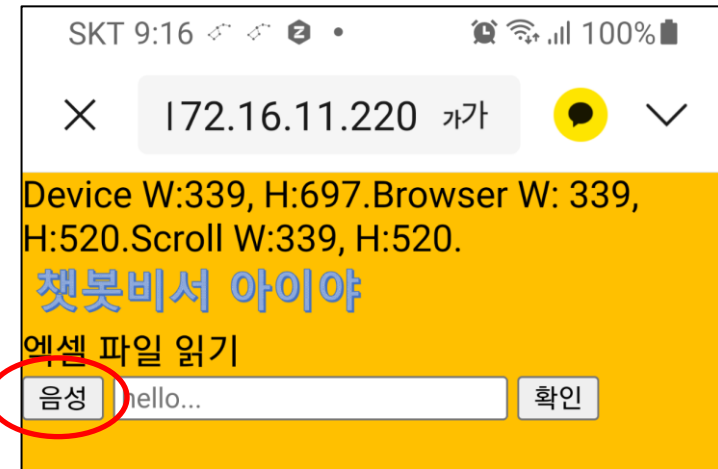
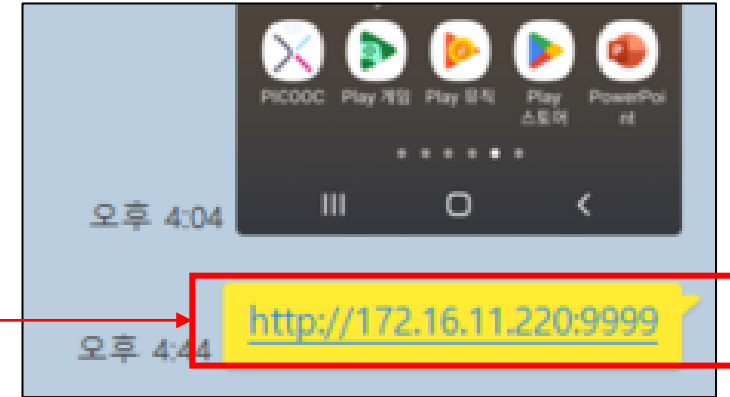
D:\seok>python main2.py
* Serving Flask app 'main2'
* Debug mode: off
WARNING: This is a development server. Do not use it in production.
* Running on http://172.16.11.220:9999
Press CTRL+C to quit
```

d:

cd _seok

python main2.py

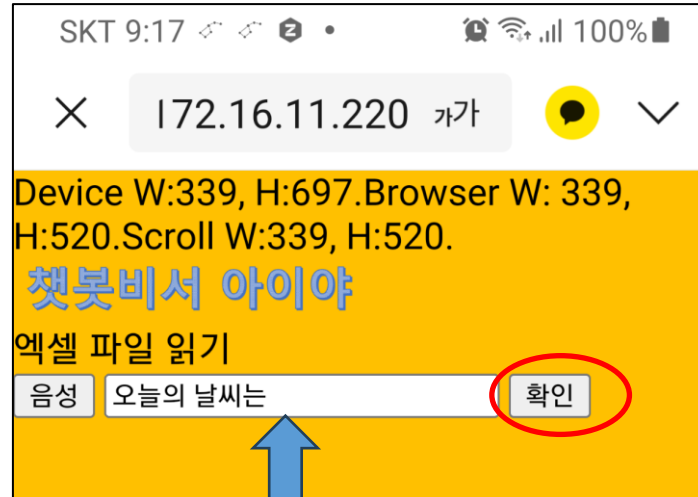
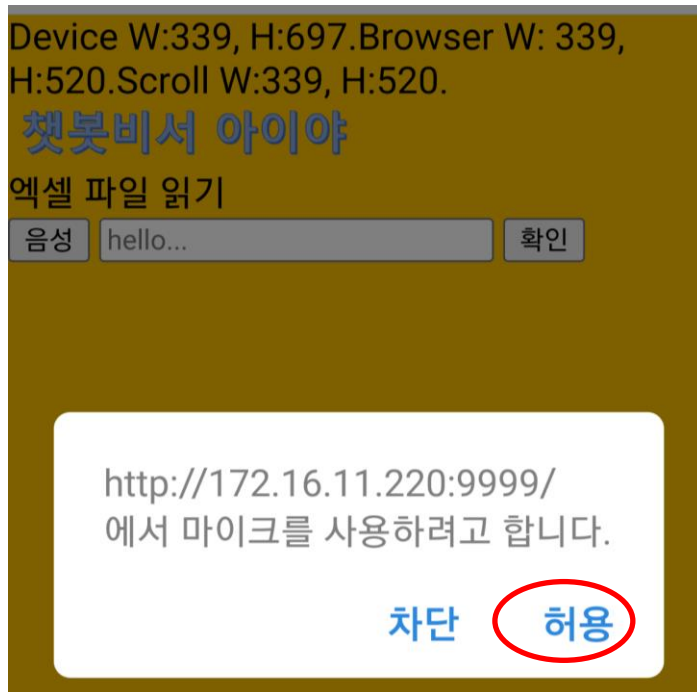
<동일한 Wifi 존>
카카오톡에 주소 입력 후 클릭



4. Flask 웹서버 수정

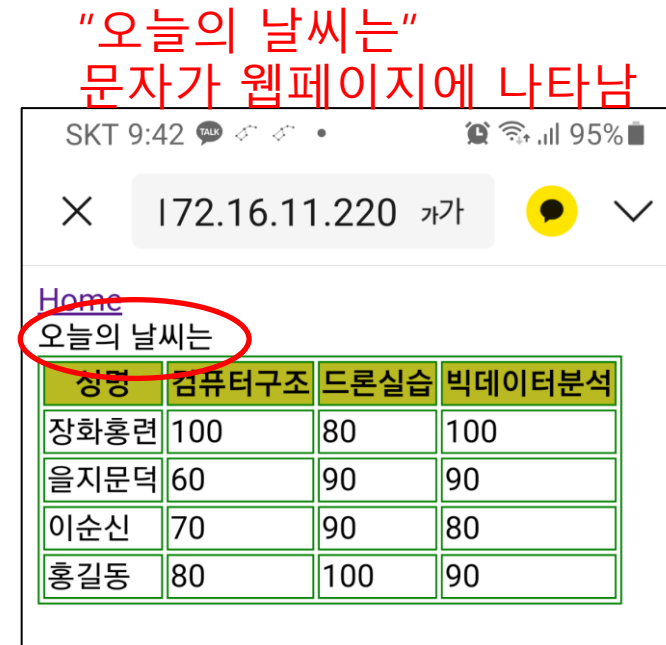
⌚ 웹서버 실행

- 스마트폰에서 접속(동일한 Wifi 존)



뽁~ 소리가 나면,
음성으로 말함
"오늘의 날씨는"

"오늘의 날씨는"
문자가 입력이 됨



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

구구단

```
#=====
# exam-1 : gugudan
#=====
for y in range(2, 10, 4):
    for x in range(1, 10):
        print("%d x %d = %2d  %d x %d = %2d  %d x %d = %2d  %d x %d = %2d"
              % (y, x, y * x, y+1, x, (y+1) * x, y+2, x, (y+2) * x, y+3, x, (y+3) * x))
    print("\n")
```

```
#=====
# exam-1 : gugudan2
# python 3.6 이상
#=====
for y in range(1, 10):
    for x in range(2, 6):
        print(f"{ x } x { y } = { x * y :2d} ", end="")
    print()
print()

for y in range(1, 10):
    for x in range(6, 10):
        print(f"{ x } x { y } = { x * y :2d}", end=" ")
    print()
```

데이터를 csv 로 저장

```
#=====
# exam-2 : write csv
#=====
import pandas as pd

y1 = [500,450,520,610]
y2 = [690,700,820,900]
y3 = [1100,1030,1200,1380]
y4 = [1500,1650,1700,1850]
y5 = [1990,2020,2300,2420]
y6 = [1020,1600,2200,2550]

df = pd.DataFrame([y1, y2, y3, y4, y5, y6],
                  index = [2018, 2019, 2020, 2021, 2022, 2023],
                  columns = ['first', 'second', 'third', 'forth'])
print(df)
df.to_csv('./sales.csv', encoding='utf-8', header = 'False')
```


라인플롯 그리기

```
#=====
# exam-3 : show lineplot chart
#=====
import pandas as pd
import matplotlib.pyplot as plt

y1 = [500,450,520,610]
y2 = [690,700,820,900]
y3 = [1100,1030,1200,1380]
y4 = [1500,1650,1700,1850]
y5 = [1990,2020,2300,2420]
y6 = [1020,1600,2200,2550]
x = range(len(y1))
xLabel = ['first', 'second', 'third', 'fourth']

plt.plot(x, y1, color='blue')
plt.plot(x, y2, color='orange')
plt.plot(x, y3, color='green')
```

```
plt.plot(x, y4, color='red')
plt.plot(x, y5, color='purple')
plt.plot(x, y6, color='brown')
plt.title('2018~2023 Quarterly sales')
plt.xlabel('Quarters')
plt.ylabel('sales')
plt.xticks(x, xLabel, fontsize=10)
plt.legend(['2018', '2019', '2020', '2021', '2022',
'2023'], loc='upper left')

plt.show()
```