

제조데이터 분석 및 시각화 ('22.02.22-'22.02.25) 1일차

Prepared by DaeKyeong Kim
Ph.D.



학습일정 및 내용



일차	시 간	교육 내용	세부 학습 내용	비고
1일차 (OO/ OO)		Syllabus Python Basics		
	5교시		Syllabus와 작업 환경 이해	강의, 실습 및 코칭
	6교시		파이썬 기초	강의, 실습 및 코칭
	7교시		파이썬 기초	강의, 실습 및 코칭
	8교시		파일 읽고 쓰기	강의, 실습 및 코칭
2일차 (OO/ OO)	1교시	Data Structures Data Cleanup		
	2교시			
	3교시			
	4교시			
	5교시		수치배열 데이터 패키지 소개	강의, 실습 및 코칭
	6교시		과학기술계산패키지 소개	강의, 실습 및 코칭
	7교시		sympy	강의, 실습 및 코칭
	8교시		데이터분석 패키지 소개	강의, 실습 및 코칭

학습일정 및 내용



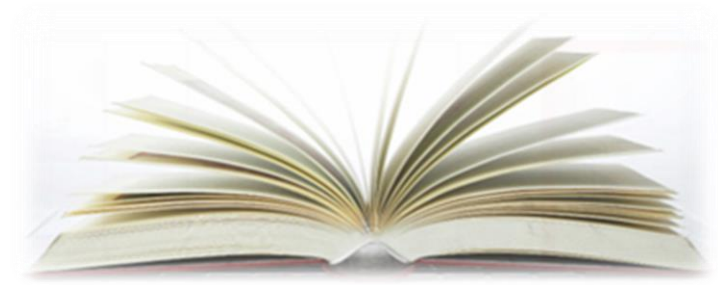
일차	시 간	교육 내용	세부 학습 내용	비고
3일차 (OO/ OO)	1교시	데이터 셋 적재하기와 Cleanup		
	2교시			
	3교시			
	4교시			
	5교시		데이터 셋 적재하기	강의, 실습 및 코칭
	6교시		Data Structures	강의, 실습 및 코칭
	7교시		Data Cleanup: Investigation, Matching, and Formatting	강의, 실습 및 코칭
	8교시		Data Cleanup: Investigation, Matching, and Formatting	강의, 실습 및 코칭
4일차 (OO/ OO)	1교시	Data Structures Data Cleanup		
	2교시			
	3교시			
	4교시			
	5교시		시각화 패키지 소개	
	6교시		시각화 패키지 소개	
	7교시		시각화 패키지 소개	
	8교시		Iris를 통한 시각화 연습	

학습일정 및 내용



일차	시 간	교육 내용	세부 학습 내용	비고
5일차 (OO/ OO)	1교시	제조데이터 분석 및 시각화 미니 프로젝트		
	2교시			
	3교시			
	4교시			
	5교시		미니 프로젝트	강의, 실습 및 코칭
	6교시		미니 프로젝트	강의, 실습 및 코칭
	7교시		미니 프로젝트	강의, 실습 및 코칭
	8교시		미니 프로젝트	강의, 실습 및 코칭

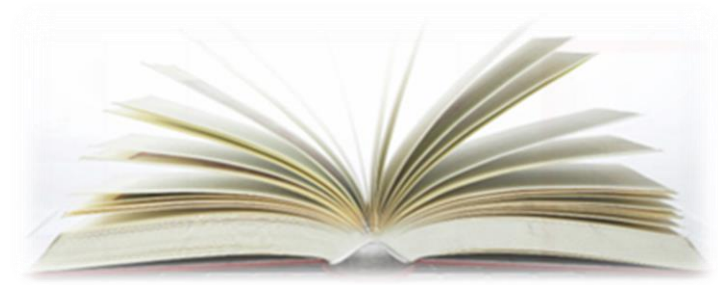
Contents



1.	Syllabus	6
2.	Python Basics	41

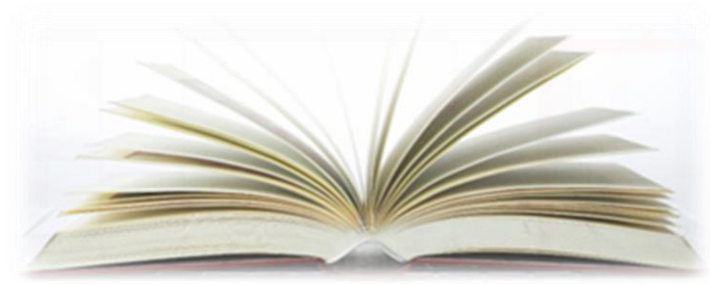
Section 1

Syllabus



1. 강사 소개
2. Pre-Requisites
3. Course Introduction
4. Course Structure

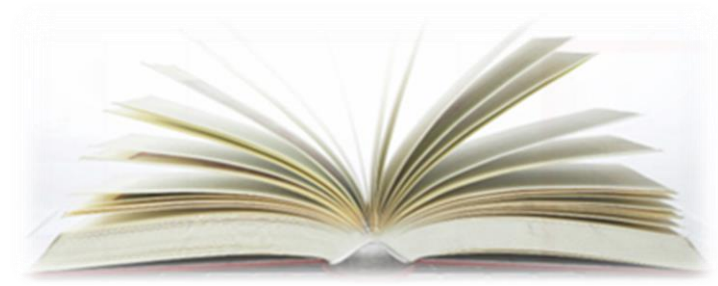
학습목표



❖ 이 워크샵에서는 강사 소개, Course Introduction과 Python Basics 에 대해 알 수 있다.

- Python 작업 환경
- Basic Data Types
- Data Structures
- 파이썬 프로그래밍 기초

Subsection 1



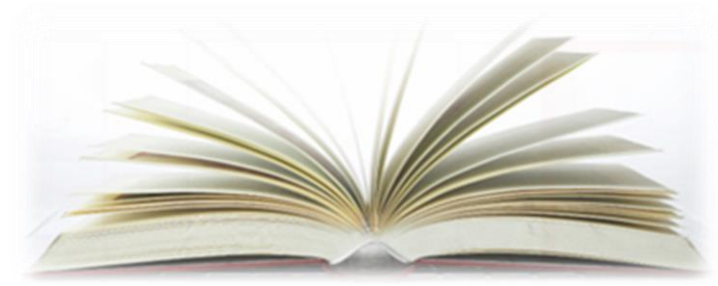
강사 소개

- ❖ Ph.D. | Solutions Architect | Software Architect | NCS 기업 컨설팅 컨설턴트 전문위원 | Trainer
- ❖ Ph.D. DaeKyeong Kim
- ❖ E-mail: onlooker2zip@naver.com
- ❖ Tel: 010-9591-1401

Trainer

- IoT | Big Data | Machine Learning | AI | Agile & TDD | Cloud Computing Services | Docker | Kubernetes

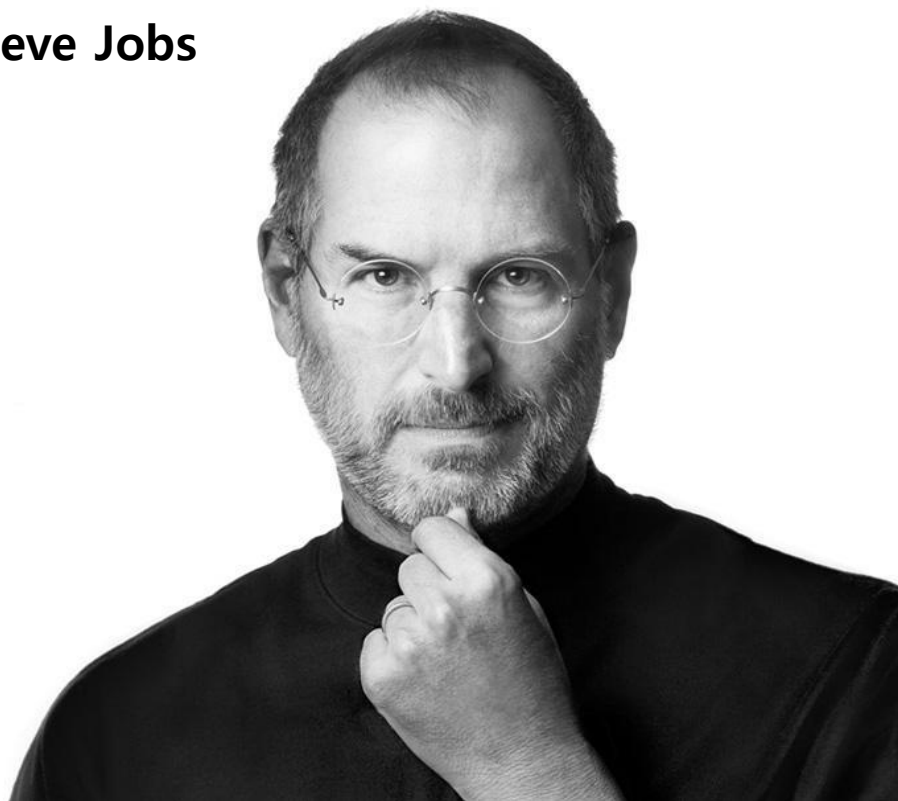
Subsection 2



Pre-Requisites

**Everybody in this country should learn how to program a computer
because it teaches you how to think.**

-Steve Jobs



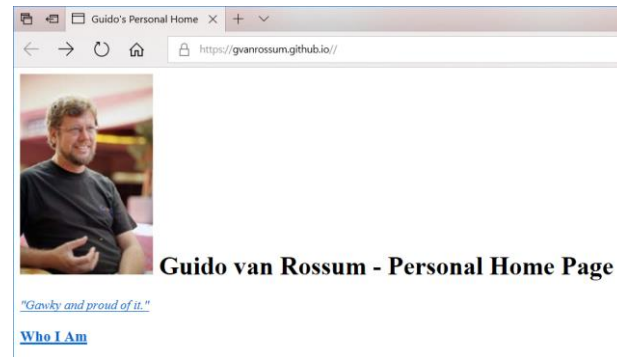
index | TIOBE - The Software Quality Company

Some say that the IT industry is changing continuously. Every day a new IT buzzword pops up somewhere. But if we take a closer look at the top 9 of the TIOBE index, it appears to be unchanged.

Jan 2021	Jan 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	17.38%	+1.61%
2	1	▼	Java	11.96%	-4.93%
3	3		Python	11.72%	+2.01%
4	4		C++	7.56%	+1.99%
5	5		C#	3.95%	-1.40%
6	6		Visual Basic	3.84%	-1.44%
7	7		JavaScript	2.20%	-0.25%
8	8		PHP	1.99%	-0.41%
9	18	▲	R	1.90%	+1.10%
10	23	▲	Groovy	1.84%	+1.23%
11	15	▲	Assembly language	1.64%	+0.76%

- ❖ 파이썬(Python)은 1991년 프로그래머인 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어로, 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑(dynamically typed) 대화형 언어이다.
- ❖ 파이썬이라는 이름은 귀도 반 로섬이 좋아하는 코미디 프로그램이던 "Monty Python's Flying Circus"에서 따온 것
- ❖ 파이썬은 초보자부터 전문가까지 폭넓은 사용자층을 보유하고 있다. 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가하고 있다. 또 파이썬은 순수한 프로그램 언어로서의 기능 외에도 다른 언어로 쓰인 모듈들을 연결하는 풀언어(glue language)로써 자주 이용된다. 실제 파이썬은 많은 상용 응용 프로그램에서 스크립트 언어로 채용되고 있다.

출처 - Wiki

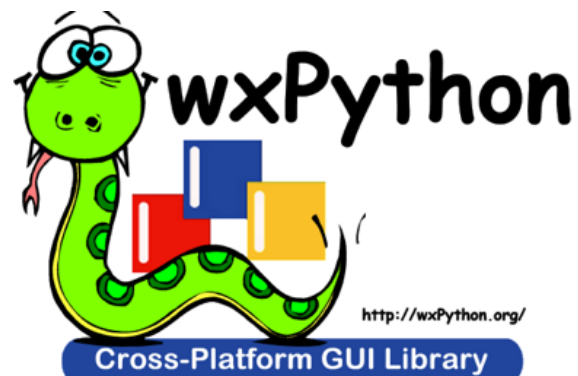


- ❖ 생산성이 뛰어나다.
Life is too short, You need python!!
- ❖ 초보자한테 좋은 언어 – 인터프리터 언어
- ❖ 파이썬은 문법이 쉬워서 코드를 보면 직관적으로 알 수 있는 부분이 많다.
- ❖ 파이썬은 다양한 플랫폼에서 사용
- ❖ 라이브러리가 풍부
- ❖ 애니메이션이나 그래픽을 쉽게 사용

❖ Web



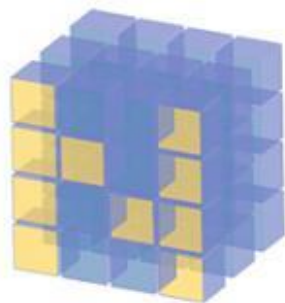
❖ GUI, Game



❖ Data Analytics



Pandas

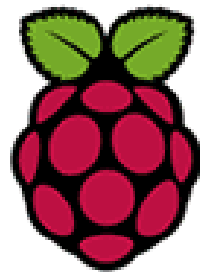


NumPy

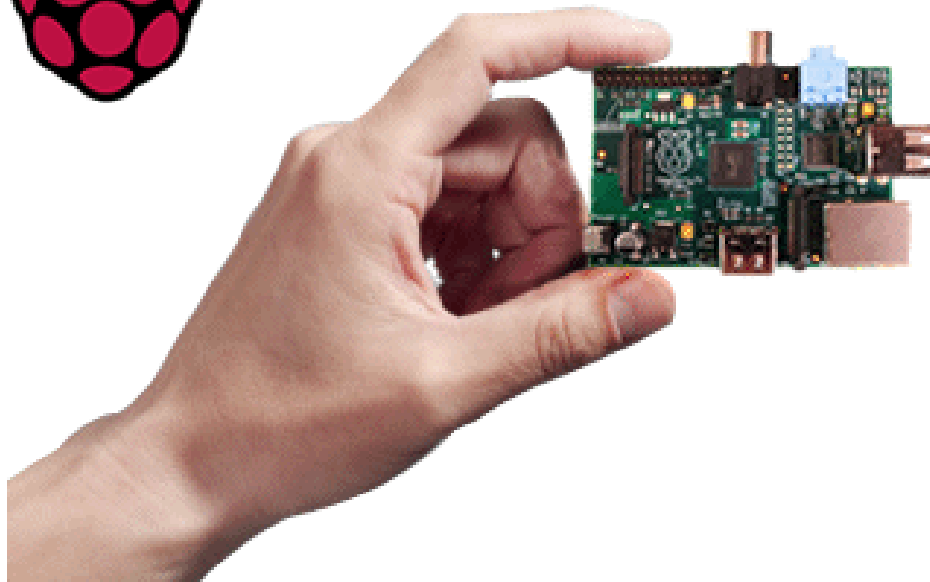
❖ Machine Learning



❖ Internet Of Things



Raspberry PiTM



...



Instagram

You**Tube**

Google

Google

이미지

어코스틱

텍스트

시계열

...

딥러닝 프레임워크

딥러닝 기초

머신러닝

확률통계 기반 데이터 분석

기본수학

분석용 데이터 구축

Data visualization

탐색적 데이터 분석

Python 라이브러리

Python

❖ Python - IDLE 설치

❖ 배포판 설치

❖ IDE 설치- PyCharm

← → anaconda.com/distribution/ T Cafe 티카래 국립안동대학교 5G로 진화하는 게... 공공데이터포털 TOEIC 안동대학교

ANACONDA

Products Wh

Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

Download

The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:

- Quickly download 1,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with Conda
- Develop and train machine learning and deep learning models with scikit-learn, TensorFlow, and Theano
- Analyze data with scalability and performance with Dask, NumPy, pandas, and Numba
- Visualize results with Matplotlib, Bokeh, Datashader, and Holoviews

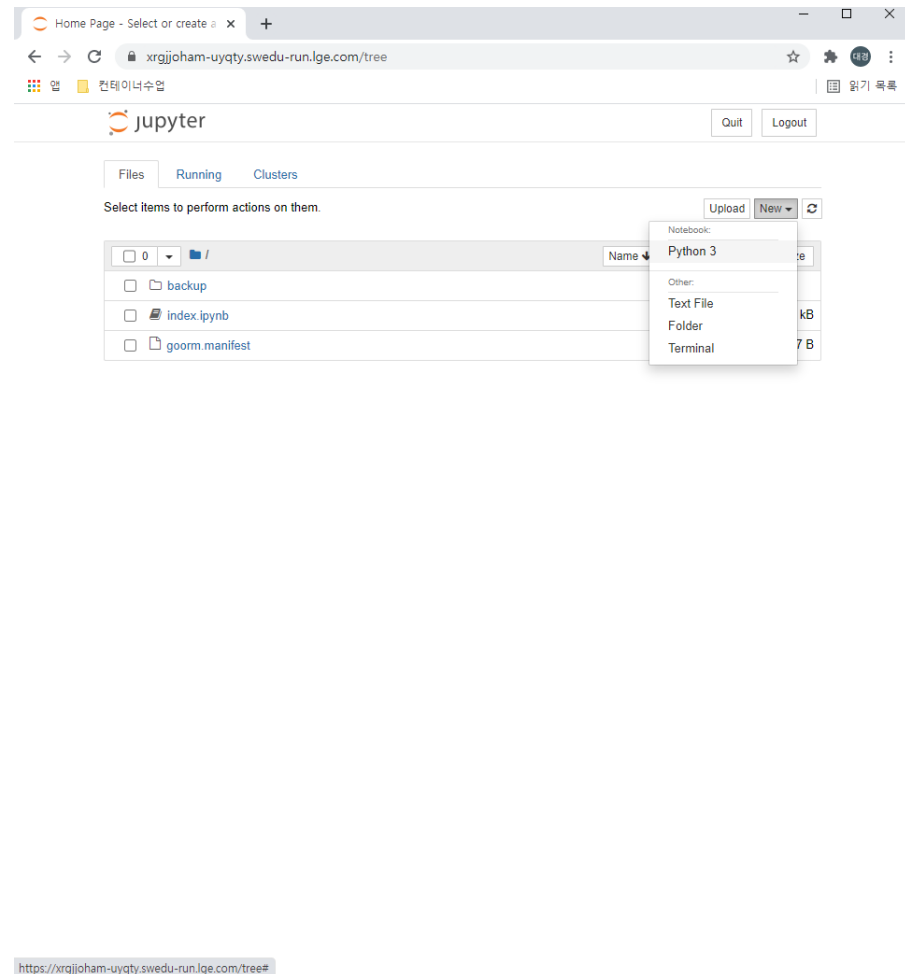


Windows | macOS | Linux



- ❖ Google Colab은 AI개발자들을 위해 구글에서 제공하는 무료 클라우드 서비스
- ❖ 풀 네임은 Google Colaboratory
- ❖ Google Drive + Jupyter Notebook
 - Google Drive처럼 협업 가능(동시에 수정 가능)
- ❖ <https://colab.research.google.com/>로 접속시 사용 가능
- ❖ 컴퓨터 사양(19년 12월 기준)
 - Ubuntu 17.10
 - CPU 제논 2.3GHz
 - 메모리 13G
 - GPU : K80 또는 T4 :
 - TPU도 사용 가능
- ❖ GPU 사용시 최대 12시간
- ❖ Github의 소스 코드를 Colab에서 사용 가능

❖ 새 Python 파일 생성



❖ 첫 번째 코랩 노트북

```
3+5
```

```
8
```

```
print("Hello 파이썬!!!")
```

```
Hello 파이썬!!!
```

❖ 작업 환경 정보 조회와 코랩 리눅스 명령

```
# OS
!cat /etc/issue.net
```

```
Ubuntu 18.04.2 LTS
```

```
# CPU 상황
!head /proc/cpuinfo
```

```
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 85
model name     : Intel(R) Xeon(R) Platinum 8124M CPU @ 3.00GHz
stepping       : 4
microcode      : 0x1000157
cpu MHz        : 2999.998
cache size     : 25344 KB
physical id    : 0
```

```
# 메모리 상황
!head -n 3 /proc/meminfo
```

```
MemTotal:       7814692 kB
MemFree:        533060 kB
MemAvailable:   6261620 kB
```

```
# 디스크 상황
!df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
none	4,8G	60M	4,5G	2%	/
tmpfs	64M	0	64M	0%	/dev
tmpfs	3,8G	0	3,8G	0%	/sys/fs/cgroup
shm	64M	0	64M	0%	/dev/shm
/dev/nvme0n1p1	117G	69G	49G	59%	/goorm
/dev/nbd1p1	4,8G	60M	4,5G	2%	/workspace
tmpfs	3,8G	0	3,8G	0%	/proc/acpi
tmpfs	3,8G	0	3,8G	0%	/proc/scsi
tmpfs	3,8G	0	3,8G	0%	/sys/firmware

```
!python --version
```

```
Python 3.7.4
```

❖ 수업자료다운로드

```
!rm -rf bigStudy  
!git clone 'https://github.com/looker2zip/bigStudy.git'
```



The image shows the JupyterLab interface. At the top, there's a 'jupyter' logo and 'Quit' and 'Logout' buttons. Below that, there are tabs for 'Files', 'Running', and 'Clusters'. A message says 'Select items to perform actions on them.' with 'Upload', 'New', and a refresh icon. The file explorer shows a directory structure with a dropdown menu set to '/'. The files listed are 'backup', 'bigStudy', and '1일차.ipynb'. A red arrow points to '1일차.ipynb'. The table has columns for 'Name', 'Last Modified', and 'File size'.

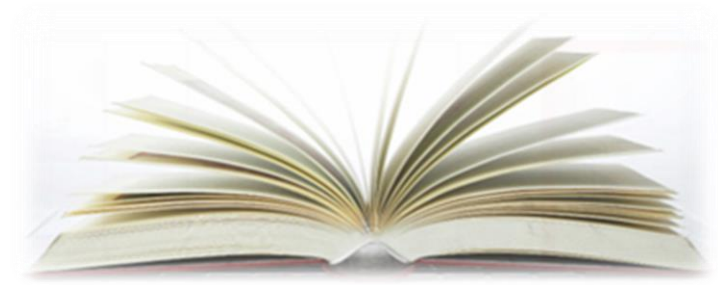
	Name	Last Modified	File size
<input type="checkbox"/>	0		
<input type="checkbox"/>	/		
<input type="checkbox"/>	backup	7일 전	
<input type="checkbox"/>	bigStudy	9시간 전	
<input type="checkbox"/>	1일차.ipynb	9시간 전	135 kB

❖ 윈도우에서 수업자료다운로드

<https://github.com/looker2zip/bigStudy>

The screenshot displays the GitHub repository page for `looker2zip/bigStudy`. The repository is public and has 1 branch and 0 tags. The 'Code' button is highlighted, and the 'Clone' dropdown menu is open, showing options to clone using HTTPS, GitHub CLI, or SVN, and to open with GitHub Desktop or download a ZIP file. The repository content shows a 'dataset' folder and a 'README.md' file. The README.md file content is visible, showing the title 'bigStudy'.

Subsection 3



Course Introduction

4차 산업혁명이란?

- ❖ 기술 연구 및 투자의 장기간 침체가 있었던 인공지능(AI)은 최근 딥러닝 기반 기술의 발달 및 기존 기술과의 결합 등을 통해 산업 전반에 적용 가능한 수준으로 발전하고 있음

인공지능, 빅데이터, 초연결 등으로 촉발되는 **지능화 혁명**, 그리고 그 이상



“빅데이터가 기업들의 성공의 배경이란 점은 더 이상 비밀이 아니지만
갈수록 빅데이터를 효율적으로 사용하는 것이 어려운 과제가 되고 있다

...

그러나 그 해결책 역시 인공지능”

인포비전(InfoVision)의 혁신 책임자(CTIO) ‘치트레이 마니(Chithrai Mani)’, 2020.10

“데이터는 실행가능한 통찰력으로 변환되지 않으면, 아무 소용이 없다”

마크 테드슨(Mark Tedson), YourStory Media, 2019. 8

- ❖ 전 세계에서 연결된 장치 즉, 사물인터넷(IoT)의 수는 2025년까지 750억 개가 넘을 것으로 예상. 이러한 기술로 수집되고 저장되는 수많은 데이터는 전 세계 조직과 사회에 혁신적인 이점을 가져다 줄 수 있지만 이 데이터를 해석할 수 있는 경우에만 가능. 여기서 등장하는 것이 **데이터 과학**
- ❖ **데이터 과학**은 과학적 방법, 프로세스, 알고리즘, 시스템을 사용하여 데이터에서 가치를 추출하는 통합 학문 분야
- ❖ **데이터 과학자**는 통계, 컴퓨터 과학, 비즈니스 지식을 포함한 다양한 기술을 결합하여 웹, 스마트폰, 고객, 센서 및 기타 소스에서 수집한 데이터를 분석
- ❖ 데이터 과학자의 업무에는 데이터 분석을 위한 전략 개발, 분석할 데이터 준비, 데이터 탐색, 분석 및 시각화, Python 및 R과 같은 프로그래밍 언어를 사용한 데이터로 모델 구축, 여러 모델을 어플리케이션으로 배포하는 작업이 포함
- ❖ 가장 효과적인 데이터 과학 업무는 팀 단위로 수행. 이 팀에는 데이터 과학자 외에도 문제를 정의하는 비즈니스 분석가, 데이터와 데이터 액세스 방법을 준비하는 데이터 엔지니어, 기본 프로세스와 인프라를 감독하는 IT 설계자, 데이터를 배포하는 어플리케이션 개발자, 분석 모델이나 출력을 어플리케이션 및 제품에 배포하는 어플리케이션 개발자를 포함

- ❖ 업무 이해 혹은 문제 정의(problem definition)
- ❖ 데이터 이해(data definition)
- ❖ 실험 계획(design of experiment) 혹은 표본화(sampling)
- ❖ 데이터 추출 혹은 취득(data acquisition)
- ❖ 데이터 가공(data processing, data wrangling)
 - PDCA(Plan-Do-Check-Action) 주기에 따라 반복
- ❖ 탐색적 분석과 데이터 시각화(exploratory data analysis, data visualization)
- ❖ 확증적 데이터 분석(Confirmatory Data Analysis, CDA) 혹은 통계적 모델링 혹은 모형화(statistical modeling)
 - 지도 학습 모델
 - 자율 학습 모델
- ❖ 효과 검증
 - A/B 테스트 등
- ❖ 서비스 구현
 - KPI(Key Performance Indicator) 모니터링 가능

- ❖ 빅 데이터 시대의 여러 가지 출처에서 수집한 가공하지 않은 데이터는 처음에는 사용할 수 없다. 하지만 충분히 깨끗해져 분석 알고리즘으로 입력될 수 있는 모든 데이터는 데이터 랭글링 과정을 거쳐야 함
- ❖ 데이터 시각화 또는 효율적인 ETL 경로를 만들길 원하는 사람들은 데이터 정리 과정을 수행해야 함
- ❖ **Data wrangling** From Wikipedia, the free encyclopedia
- ❖ 데이터 랭글링 (데이터 병합이라고도 함)은 분석과 같은 다양한 다운 스트림 목적에 더 적합하고 가치있는 데이터를 만들기 위해 하나의 '원시' 데이터 형식에서 다른 형식으로 데이터를 변환하고 매핑하는 프로세스
- ❖ 데이터 랭글링의 목표는 품질과 유용한 데이터를 보장하는 것
- ❖ 데이터를 분석하기 위해 대부분의 시간을 데이터 정리 및 조직, 라벨링에 소요하고 있다. 데이터 랭글링 과정 전체는 다양한 데이터 유형과 출처, API, 적용, 데이터 정리 등 전부를 통합하는 작업 필요

Python Basics

- Python 작업 환경
- Basic Data Types
- Data Structures
- 파이썬 프로그래밍 기초
- ...

Python 라이브러리

Gathering Data/Data Cleanup

Python 시각화

Python Basics

Python 라이브러리

- Numpy
- matplotlib
- seaborn
- pandas
- Scipy
- statsmodels
- scikit-learn
- ...

Gathering Data/Data Cleanup

Python 시각화

Python Basics

Python 라이브러리

Gathering Data/Data Cleanup

- Query and replace
- Finding Outliers and Bad Data
- Formatting Data
- ...

Python 시각화

Python Basics

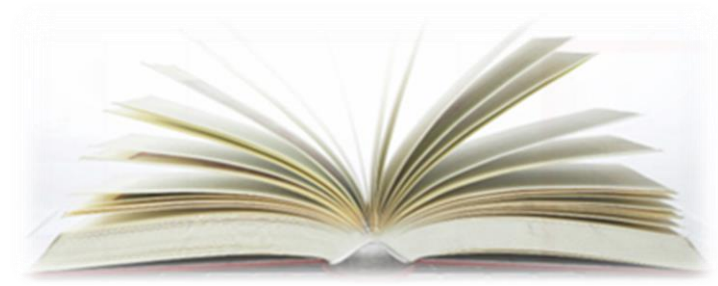
Python 라이브러리

Gathering Data/Data Cleanup

Python 시각화

- matplotlib
- Seaborn
- ...

Subsection 4



Course Structure

❖ 필요에 따라 다음 내용을 적절하게 조합해 수업을 진행한다.

- Lecture
- Demos
- Quizzes
- Practice Questions
- Q&A

Schedule

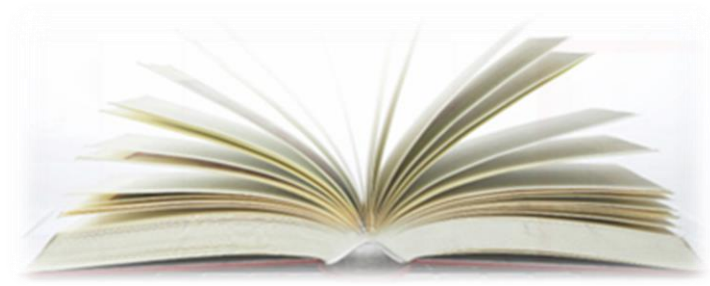
Unit

2

교과시간표

12.28 (화)	과목명	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련		특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	8	8
	내용	인공지능 서비스 목표 항목 정의하기 목표 항목 도출 관리 항목 세부 사항 정의	인공지능 서비스 목표 항목 관리 항목별 현재 수준 정의 서비스 제약 조건 도출 서비스 목표 수준 설정	분석 목표 정의서 프로젝트 현장 (Project Charter)		훈련, 평가, 검증 데이터 세트	교차 유효성(cross validation) 검사	MAE MSE RMS MAPE	오류율, 정확도, 민감도, 특이도	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 통계기반		
	강사명	김대경	김대경	김대경		김대경	김대경	김대경	김대경	김대경		
01.11 (화)	과목명	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련		특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위에 대한 핵심 개념 이해 및 PoC 훈련	8	16
	내용	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 통계기반	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 머신러닝기반	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 머신러닝기반		AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 머신러닝기반	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 딥러닝기반	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 딥러닝기반	AI-딥러닝 분석모델 선정과 핵심 개념-통계 및 목표 설정 딥러닝기반	능력단위 선정		
	강사명	김대경	김대경	김대경		김대경	김대경	김대경	김대경	김대경		
01.21 (금)	과목명	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위 선정	특허정보 구조, 특허제도, 데이터 사이언스 등 빅데이터 분석 직무 처리를 위한 능력단위 선정	선정한 능력단위 처리를 위한 빅데이터/인공지능-딥러닝 환경 구축		선정한 능력단위 처리를 위한 빅데이터/인공지능-딥러닝 환경 구축	선정한 능력단위 처리를 위한 빅데이터/인공지능-딥러닝 환경 구축	선정한 능력단위 처리를 위한 빅데이터/인공지능-딥러닝 환경 구축	시험한 빅데이터/인공지능-딥러닝 환경 결과와 발표자료 작성	발표		
	내용	능력단위 확정	능력단위에 따른 능력단위 요소 정리	원10에 DataAnalysis 환경 구성 개관		Windows9에서 Anaconda Prompt를 이용해 가상환경 만들기	Windows10에서 Anaconda Prompt를 이용해 가상환경 만들기	Google Colab	분석 결과 및 자료 취합 후 발표자료 작성	발표	8	24
	강사명	윤세찬	윤세찬	김대경		윤세찬	윤세찬	김대경	김대경	김대경		
01.24 (월)	과목명	선정한 능력단위 처리를 위한 빅데이터/인공지능-딥러닝 환경 구축	빅데이터/인공지능-딥러닝 환경 시험	빅데이터/인공지능-딥러닝 환경 시험		시험한 빅데이터/인공지능-딥러닝 환경 결과 취합 후 발표자료 작성	시험한 빅데이터/인공지능-딥러닝 환경 결과 취합 후 발표자료 작성	발표				
	내용	통계기반 분석 모델 수행	머신러닝기반 분석 모델 수행	딥러닝기반 분석 모델 수행		분석 결과 및 자료 취합 후 발표자료 작성	분석 결과 및 자료 취합 후 발표자료 작성	발표			6	30
	강사명	윤세찬	윤세찬	윤세찬		윤세찬	윤세찬	윤세찬				
	강사소	사무실/강의실	사무실/강의실	사무실/강의실		사무실/강의실	사무실/강의실	사무실/강의실	사무실/강의실	사무실/강의실		

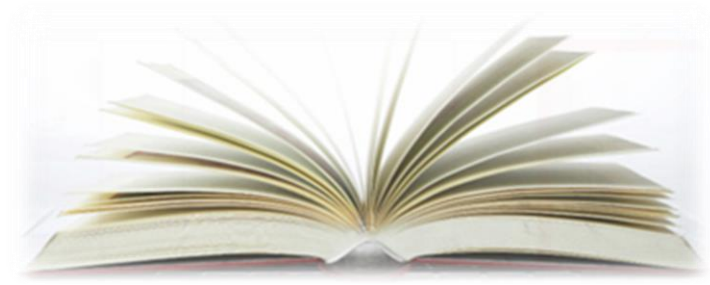
Section 2



Python Basics

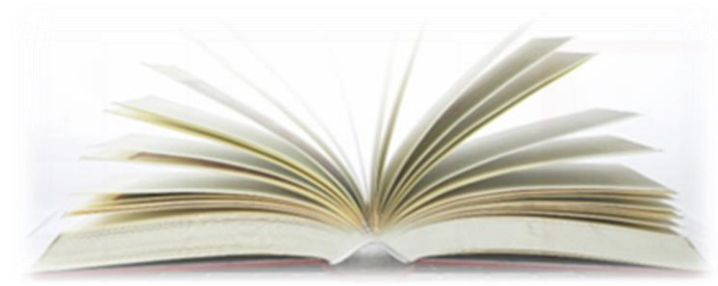
1. 파이썬 기초
2. 파이썬 라이브러리
3. 사용자 입력과 출력
4. 데이터 저장

학습목표



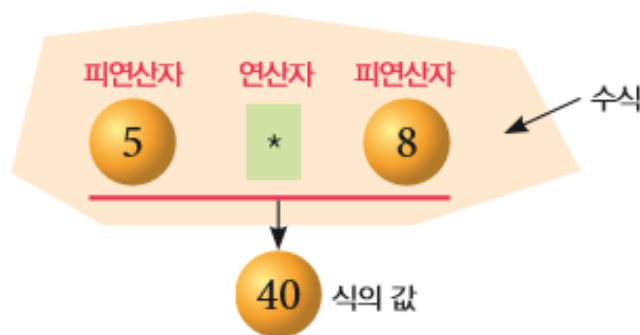
- ❖ 이 워크샵에서는 Python 기초 문법과 Coding 작성법을 알 수 있다. 파이썬 라이브러리를 알 수 있다.
- Python 작업 환경
- Basic Data Types
- Data Structures
- 파이썬 프로그래밍 기초

Subsection 1



파이썬 기초

- ❖ 수식(expression)=피연산자들과 연산자의 조합
 - 연산자(operator)는 연산을 나타내는 기호
 - 피연산자(operand)는 연산의 대상이 되는 것



- ❖ "8 PLUS 2 EQUALS 10." (8 더하기 2는 10.)
- ❖ "10 MINUS 2 EQUALS 8." (10 빼기 2는 8.)
- ❖ "2 TIMES(MULTIPLIED) 8 EQUALS 16." (2 곱하기 8은 16)
- ❖ "10 DIVIDED BY 2 EQUALS 5." (10 나누기 2는 5.)
- ❖ "7 DIVIDED BY 3 IS 2 WITH A REMAINDER OF 1." (7 나누기 3은 2와 나머지 1)

❖ 예

 $8+2$

10

 $10-2$

8

 $2*8$

16

 $10/5$ # 몫, 나누기

2,0

 $10//5$ # 몫, 나누기

2

 $7/3$ # 몫, 나누기

2,3333333333333335

 $7//3$ # 몫, 나누기

2

 $7\%3$ # 나머지

1

연산 순서와 괄호

- ❖ 곱셈과 나눗셈을 덧셈이나 뺄셈보다 먼저 하지만, 괄호를 사용해 연산 순서를 제어할 수 있다.

```
In [15]: 3 + 2 * 4
```

```
Out [15]: 11
```

```
In [16]: 3 + (2 * 4)
```

```
Out [16]: 11
```

```
In [17]: (3 + 2) * 4
```

```
Out [17]: 20
```

❖ 퀴즈 $100 \div [3 \times \{10 - (3 \times 2)\} + 8]$

```
In [18]: 100 / (3 * (10 - (3 * 2)) + 8)
```

```
Out [18]: 5.0
```

❖ 줄 제일 앞에 #을 쓰면 해당 행은 주석으로 취급

```
In [19]: # 10+ 10
```


- ❖ 파이썬에서는 문자나 문자열을 표현할 때 큰따옴표, 작은따옴표 어떤 것을 사용해도 무관하다.

```
In [20]: "A"
```

```
Out [20]: 'A'
```

```
In [21]: 'A'
```

```
Out [21]: 'A'
```

```
In [22]: type("A")
```

```
Out [22]: str
```

```
In [23]: type('A')
```

```
Out [23]: str
```

❖ Integers examples:

- 10
- 1
- 0
- -1
- -10

❖ Floats examples:

- 2
- 2.0

❖ 정수부가 0인 실수는 0을 생략할 수 있다.

- `.1 #.1 = 0.1`

```
In [24]: type(10)
```

```
Out [24]: int
```

```
In [26]: type(2.0)
```

```
Out [26]: float
```

```
In [27]: .1 #.1
```

```
Out [27]: 0.1
```

❖ 대소문자 구별 주의

```
In [28]: type(True)
```

```
Out [28]: bool
```

```
In [30]: type(TRUE)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-30-0bcb49f3077c> in <module>
----> 1 type(TRUE)

NameError: name 'TRUE' is not defined
```

```
In [31]: type(False)
```

```
Out [31]: bool
```

```
In [32]: type(FALSE)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-32-f80fe4603aff> in <module>
----> 1 type(FALSE)

NameError: name 'FALSE' is not defined
```

```
In [33]: 1 > 0.89
```

```
Out [33]: True
```

```
In [34]: 3 < 2
```

```
Out [34]: False
```

- ❖ 변수명-변수는 컴퓨터의 메모리 안에 만들어지는 공간으로 우리는 여기에 숫자나 문자를 저장할 수 있다.
 - ✓ 문자, 숫자, 밑줄(_)로 구성됩니다.
 - ✓ 숫자는 처음에 나올 수 없습니다.
 - ✓ 대소문자를 구분합니다.
 - ✓ 예약어 사용 불가.
- ❖ 지역 변수 – 함수 안에서 선언되는 변수
- ❖ 전역 변수 – 함수 외부에서 선언되는 변수
- ❖ 인스턴스 변수 – 클래스 안에 선언된 변수, 앞에 self.가 붙는다.

```
In [35]: x = 10  
x + 1
```

```
Out [35]: 11
```

```
In [37]: count=1  
message="Hello!"  
count+message
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-37-a04cbcb708ef> in <module>  
      1 count=1  
      2 message="Hello!"  
>>> 3 count+message  
  
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

- ❖ 리스트는 여러 요소들을 갖는 집합(컬렉션)으로 새로운 요소를 추가하거나 갱신, 삭제하는 일이 가능하다. 파이썬의 리스트는 동적배열(Dynamic Array)로서 자유롭게 확장할 수 있는 구조를 갖는다.
- ❖ 리스트의 요소들은 Square bracket([])으로 둘러싸여 컬렉션을 표현하는데, 각 요소들은 서로 다른 타입이 될 수 있으며, 콤마(,)로 구분한다. 요소가 없는 빈 리스트는 "[]"와 같이 표현한다.

```
In [12]: L = [1,2,3,4,5]
```

```
In [13]: L[0] = 100
```

```
In [14]: print("L[0] == ", L[0], ", L[2] == ", L[2], ", L[4] == ", L[4])
```

```
L[0] == 100 , L[2] == 3 , L[4] == 5
```

```
In [15]: print("L[-1] == ", L[-1], ", L[-3] == ", L[-3], ", L[-5] == ", L[-5])
```

```
L[-1] == 5 , L[-3] == 3 , L[-5] == 100
```

- ❖ Tuple은 리스트와 같지만 다음 내용이 다르다.
- ❖ 리스트 내 원소는 변경 가능하지만, 튜플은 변경할 수 없다.

```
In [16]: T = (1,2,3,4,5)
```

```
In [17]: print("T[0] == ", T[0], ", T[2] == ", T[2], ", T[4] == ", T[4])
```

```
T[0] == 1 , T[2] == 3 , T[4] == 5
```

```
In [18]: print("T[-1] == ", T[-1], ", T[-3] == ", T[-3], ", T[-5] == ", T[-5])
```

```
T[-1] == 5 , T[-3] == 3 , T[-5] == 1
```

```
In [19]: T[0] = 100
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-19-4fc510be16ee> in <module>  
----> 1 T[0] = 100  
  
TypeError: 'tuple' object does not support item assignment
```

❖ Dictionary는 맵과 비슷

```
In [22]: score={"KOR":95, "ENG":90, "MATH":100}
```

```
In [23]: print("score['KOR'] == ", score['KOR'])
```

```
score['KOR'] == 95
```

```
In [24]: print(score)
```

```
{'KOR': 95, 'ENG': 90, 'MATH': 100}
```

```
In [25]: print("score key == ", score.keys())  
print("score value == ", score.values())  
print("score items == ", score.items())
```

```
score key == dict_keys(['KOR', 'ENG', 'MATH'])  
score value == dict_values([95, 90, 100])  
score items == dict_items([('KOR', 95), ('ENG', 90), ('MATH', 100)])
```

❖ 문자열

- 케이스 변경
- 문자열 끝에서 공백 제거
- 문자열 분할

❖ 정수 및 소수

- 더하기 및 빼기
- 간단한 수학

❖ 리스트

- 리스트에서 더하거나 빼기
- 리스트의 마지막 항목 제거
- 리스트 재정렬
- 리스트 정렬

❖ 딕셔너리

- 키 / 값 쌍 추가
- 해당 키에 새 값 설정
- 키로 값 조회


```
type('20011')  
type(20011)
```

```
In [27]: type('20011')
```

```
Out [27]: str
```

```
In [28]: type(20011)
```

```
Out [28]: int
```

dir 과 help

❖ 함수나 모듈에 대한 정보

- `dir(클래스)` 를 입력할 경우 해당 클래스에 내장되어있는 함수들의 이름 리스트를 알아낼 수 있다.
- `help(클래스)`나 `help(함수)`를 할 경우 해당 클래스나 함수의 정보를 얻을 수 있지만 불필요한 정보까지 얻을 수 있다.

```
In [32]: dir(list)
```

```
['__reduce_ex__',  
 '__repr__',  
 '__reversed__',  
 '__rmul__',  
 '__setattr__',  
 '__setitem__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'append',  
 'clear',  
 'copy',  
 'count',  
 'extend',  
 'index',  
 'insert',  
 'pop',  
 'remove',  
 'reverse',  
 'sort']
```

```
In [34]: help(list.append)
```

Help on method_descriptor:

```
append(self, object, /)  
    Append object to the end of the list.
```

```
In [35]: import random
         dir(random)
```

```
Out [35]: ['BPF',
          'LOG4',
          'NV_MAGICCONST',
          'RECIP_BPF',
          'Random',
          'SG_MAGICCONST',
          'SystemRandom',
          'TWOPI',
          '_BuiltinMethodType',
          '_MethodType',
          '_Sequence',
          '_Set',
          '__all__',
          '__builtins__',
          '__cached__',
          '__doc__',
          '__file__',
          '__loader__',
          '__name__',
          '__package__']
```

```
In [36]: help(random.randint)
```

Help on method randint in module random:

randint(a, b) method of random.Random instance
Return random integer in range [a, b], including both end points.

if 문

if <조건식> :
 <구문>

```
In [37]: x = 5  
         if x == 5:  
             print("x is equal to 5,")
```

x is equal to 5,

```
In [38]: x = 3  
         if x == 5:  
             print('x is equal to 5,')  
         else:  
             print('x is not equal to 5,')
```

x is not equal to 5,

loop와 range()

for 변수 이름 **in** 리스트 :

<구문>

❖ Range

- range() 함수는 보통 아래와 같이 1~3개의 파라미터를 갖는데, 파라미터는 파라미터 갯수에 따라 아래와 같이 다른 의미를 갖는다.

예제	파라미터 의미	리턴값	뜻
range(3)	Stop	0, 1, 2	0부터 3미만의 숫자를 포함하는 range 객체
range(3,6)	Start, Stop	3, 4, 5	3부터 6미만의 숫자를 포함하는 range 객체
range(2,11,2)	Start, Stop, Step	2, 4, 6, 8, 10	2부터 11미만의 2의 배수를 포함하는 range 객체

```
In [39]: for i in [1,2,3]:  
         print(i)
```

```
1  
2  
3
```

```
In [40]: range(0,3)
```

```
Out [40]: range(0, 3)
```

```
In [ ]: for i in range(0, 3):  
         print(i)
```

```
In [42]: for i in range(0, 3):  
         print("Hello Python")
```

```
Hello Python  
Hello Python  
Hello Python
```

❖ 계산 로직을 저장해 놓은 것

```
In [43]: (x + 2) * 4
```

```
Out [43]: 20
```

```
In [44]: def sample_function(data):  
         return ((data + 2) * 4)
```

```
In [45]: sample_function(3)
```

```
Out [45]: 20
```

❖ 간단한 함수와 람다(lambda) 함수

```
In [46]: def Times(a,b):  
         return a*b
```

```
In [47]: Times(10,10)
```

```
Out [47]: 100
```

```
In [48]: sum=lambda a,b:a+b
```

```
In [49]: sum(3,4)
```

```
Out [49]: 7
```

Class 클래스명:

def 메서드 A(self, 인수 A):

처리 A

def 메서드 B(self, 인수 B):

처리 B

```
In [53]: class Sample_Class:
          def __init__(self, data1, data2):
              self.data1 = data1
              self.data2 = data2

          def method2(self):
              return(self.data1 + self.data2)
```

```
In [54]: sample_instance = Sample_Class(data1 = 2, data2 = 3)
```

```
In [55]: sample_instance.data1
```

```
Out [55]: 2
```

```
In [56]: sample_instance.data2
```

```
Out [56]: 3
```

```
In [57]: sample_instance.method2()
```

```
Out [57]: 5
```


- ❖ 현재 파이썬 3.0 버전에서는 대략 200개가 넘는 모듈을 지원
 - ✓ 문자열(string), 날짜(date), 시간(time), 십진법(decimal), 랜덤(random)
 - ✓ 파일(file), os, sqlite3, sys, xml, email, http 등등
- ❖ math 모듈은 삼각함수, 제곱근, 로그함수 등 수학과 관련된 기능이 들어 있는 내장 모듈.

math.ceil(<i>x</i>)	math.fabs(<i>x</i>)	math.factorial(<i>x</i>)	math.floor(<i>x</i>)
math.fmod(<i>x</i> , <i>y</i>)	math.frexp(<i>x</i>)	math.fsum(<i>iterable</i>)	math.isinf(<i>x</i>)
math.isnan(<i>x</i>)	math.ldexp(<i>x</i> , <i>i</i>)	math.modf(<i>x</i>)	math.trunc(<i>x</i>)
math.exp(<i>x</i>)	math.expm1(<i>x</i>)	math.log(<i>x</i> [, <i>base</i>])	math.log1p(<i>x</i>)
math.log10(<i>x</i>)	math.pow(<i>x</i> , <i>y</i>)	math.sqrt(<i>x</i>)	math.acos(<i>x</i>)
math.asin(<i>x</i>)	math.atan(<i>x</i>)	math.atan2(<i>y</i> , <i>x</i>)	math.cos(<i>x</i>)
math.hypot(<i>x</i> , <i>y</i>)	math.sin(<i>x</i>)	math.tan(<i>x</i>)	math.degrees(<i>x</i>)
math.radians(<i>x</i>)	math.acosh(<i>x</i>)	math.asinh(<i>x</i>)	math.atanh(<i>x</i>)
math.cosh(<i>x</i>)	math.sinh(<i>x</i>)	math.tanh(<i>x</i>)	math.erf(<i>x</i>)
math.erfc(<i>x</i>)	math.gamma(<i>x</i>)	math.lgamma(<i>x</i>)	math.pi & math.e



```
import math
dir(math)
```



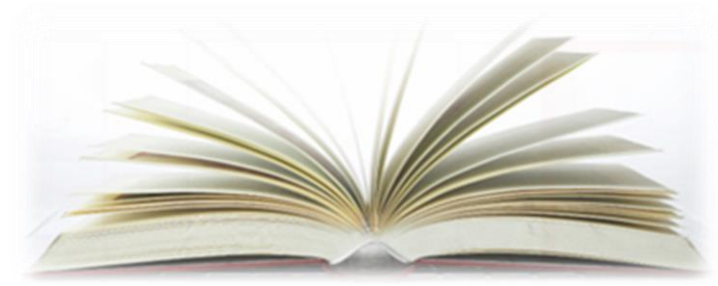
```
['__doc__',
 '__loader__',
 '__name__',
```

```
[65] math.pow(2, 10)
```



```
1024.0
```

Subsection 2



파이썬 라이브러리

❖ 주피터 노트북

- ✓ 프로그램 코드를 브라우저에서 실행해주는 대화식 환경.
- ✓ 탐색적 데이터 분석에 적합.

❖ Numpy 넘파이

- ✓ 파이썬으로 과학 계산을 하려면 꼭 필요한 패키지.
- ✓ 다차원 배열을 위한 기능과 선형 대수 연산과 푸리에 변환 같은 고수준 수학 함수와 유사 난수 생성기를 포함
- ✓ 다음 명령으로 설치한다.

`pip install numpy`

- ✓ 임포트할 때는 보통 np라는 별명으로 임포트한다.

`import numpy as np`

❖ Scipy 싸이파이

- ✓ 과학 계산을 함수를 모아놓은 파이썬 패키지.
- ✓ 고성능 선형 대수, 함수 최적화, 신호 처리, 특수한 수학 함수와 통계 분포 등을 포함한 많은 기능 제공
- ✓ 다음 명령으로 설치한다.

`pip install scipy`

- ✓ 임포트할 때는 보통 `sp`라는 별명으로 임포트한다.

`import scipy as sp`

❖ matplotlib 맷플랏립

- ✓ 대표적인 과학 계산을 그래프 라이브러리.
- ✓ 선 그래프, 히스토그램, 산점도 등을 지원하며 고품질 그래프를 그려준다.

❖ matplotlib 맷플랏립

- ✓ 대표적인 과학 계산을 그래프 라이브러리.
- ✓ 선 그래프, 히스토그램, 산점도 등을 지원하며 고품질 그래프를 그려준다.
- ✓ 다음 명령으로 설치한다.

`pip install matplotlib`

- ✓ 임포트할 때는 보통 `pylab` 서브패키지를 `plt`라는 별명으로 임포트한다.

`import matplotlib.pyplot as plt`

❖ pandas 판다스

- ✓ 데이터 처리와 분석을 위한 파이썬 라이브러리.
- ✓ R의 data.frame을 본떠서 설계한 DataFrame이라는 데이터 구조를 기반으로 만들어졌다.
- ✓ 엑셀의 스프레드시트와 비슷한 테이블 형태.
- ✓ SQL처럼 테이블에 쿼리나 조인을 수행할 수 있다.
- ✓ SQL, 엑셀 파일, CSV 파일 같은 다양한 파일과 DB에서 읽어들이 수 있다.
- ✓ 다음 명령으로 설치한다.

`pip install pandas`

- ✓ импорт할 때는 보통 pd라는 별명으로 импорт한다.

`import pandas as pd`

❖seaborn 패키지

- ✓ seaborn("시본"이라고 읽는다) 패키지는 맷플롯립 패키지에서 지원하지 않는 고급 통계 차트를 그리는 통계용 시각화 기능을 제공한다.
- ✓ 다음 명령으로 설치한다.

pip install seaborn

- ✓ импорт할 때는 보통 sns라는 별명으로 импорт한다.

import seaborn as sns

❖ statsmodels 패키지

- ✓ statsmodels("스탯츠모델즈"라고 읽는다) 패키지는 추정 및 검정, 회귀분석, 시계열분석 등의 기능을 제공하는 파이썬 패키지다. 기존에 R에서 가능했던 다양한 회귀분석과 시계열분석 방법론을 그대로 파이썬에서 이용할 수 있다. 다음은 statsmodels 패키지가 제공하는 기능의 일부다.
- ✓ 예제 데이터셋
- ✓ 검정 및 모수추정
- ✓ 회귀분석
 - 선형회귀
 - 강건회귀
 - 일반화 선형모형
 - 혼합효과모형
 - 이산종속변수
- ✓ 시계열 분석
 - SARIMAX 모형
 - 상태공간 모형
 - 벡터 AR 모형
- ✓ 생존분석
- ✓ 요인분석

- ✓ 다음 명령으로 설치한다.
`pip install statsmodels`
- ✓ 임포트할 때는 보통 api 서브패키지를 sm이라는 별명으로 임포트한다.
`import statsmodels.api as sm`

❖ scikit-learn 패키지

- ✓ scikit-learn("사이킷런"이라고 읽는다) 패키지는 머신러닝 교육을 위한 최고의 파이썬 패키지다. scikit-learn 패키지의 장점은 다양한 머신러닝 모형을 하나의 패키지에서 모두 제공하고 있다는 점이다. 다음은 scikit-learn 패키지에서 제공하는 머신러닝 모형의 목록의 일부다.
- ✓ 데이터셋
 - 회귀분석, 분류, 클러스터링용 가상 데이터셋 생성
 - 각종 벤치마크 데이터셋
- ✓ 전처리
 - 스케일링
 - 누락데이터 처리
 - 텍스트 토큰화
- ✓ 지도학습
 - 회귀분석
 - LDA/QDA
 - 서포트벡터머신
 - 퍼셉트론, SGD
 - KNN
 - 가우스프로세스
 - 나이브베이즈
 - 의사결정나무
 - 랜덤포레스트, 부스팅

✓ 비지도 학습

- 가우스 혼합모형
- 클러스터링
- PCA

✓ 성능 최적화

- 교차검증
- 특징선택
- 하이퍼파라미터 최적화

✓ 설치와 임포트에 사용하는 이름은 sklearn이다. 다음 명령으로 설치한다.

`pip install sklearn`

✓ 임포트할 때는 보통 sk이라는 별명으로 임포트한다.

`import sklearn as sk`

❖ missingno 패키지

- ✓ pandas 데이터프레임 데이터에서 누락된 데이터를 찾고 시각화하는 기능을 제공한다.
- ✓ 다음 명령으로 설치한다.
- ✓ `pip install missingno`

❖ patsy 패키지

- ✓ pandas 데이터프레임 데이터에서 선택, 변형하는 기능을 제공한다.
- ✓ statsmodels가 의존하는 패키지이므로 statsmodels 패키지를 설치하면 별도로 설치할 필요가 없다.

- ❖ 텍스트 전처리용 패키지
 - ✓ nltk 패키지
 - ✓ spacy 패키지
 - ✓ konlpy 패키지
 - ✓ soynlp 패키지
 - ✓ gensim 패키지
- ❖ 이미지 전처리용 패키지
 - ✓ opencv 패키지
- ❖ 사운드 전처리용 패키지
 - ✓ librosa 패키지
- ❖ 지리정보 전처리용 패키지
 - ✓ geopandas 패키지

```
import sys
print("python 버전 : {}".format(sys.version))

import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))

import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))

import numpy as np
print("numpy 버전 : {}".format(np.__version__))

import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))

import IPython
print("IPython 버전 : {}".format(IPython.__version__))

import sklearn
print("sklearn : {}".format(sklearn.__version__))
```

5]: # Unit 2 - 머신러닝에 필요한 라이브러리들 설치/확인

```
import sys

print("python 버전 : {}".format(sys.version))

import pandas as pd
print("pandas 버전 : {}".format(pd.__version__))

import matplotlib
print("matplotlib 버전 : {}".format(matplotlib.__version__))

import numpy as np
print("numpy 버전 : {}".format(np.__version__))

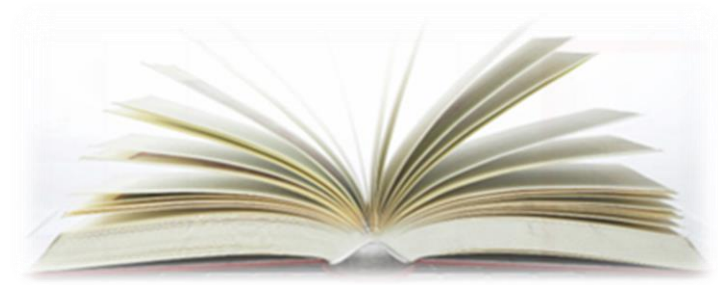
import scipy as sp
print("scipy 버전 : {}".format(sp.__version__))

import IPython
print("IPython 버전 : {}".format(IPython.__version__))

import sklearn
print("sklearn : {}".format(sklearn.__version__))
```

```
python 버전 : 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64
pandas 버전 : 1.0.0
matplotlib 버전 : 3.1.3
numpy 버전 : 1.18.1
scipy 버전 : 1.4.1
IPython 버전 : 7.12.0
sklearn : 0.22.1
```

Subsection 3



사용자 입력과 출력

❖ input은 입력되는 모든 것을 문자열로 취급한다.

```
a = input()
```

안녕하세요? Hello!

```
a
```

'안녕하세요? Hello!'

- ❖ 사용자에게 입력받을 때 "숫자를 입력하세요"라든지 "이름을 입력하세요"라는 안내 문구 또는 질문이 나오도록 하고 싶을 때가 있다. 그럴 때는 `input()`의 괄호 안에 질문을 입력하여 프롬프트를 띄워주면 된다.

```
number = input("숫자를 입력하세요: ")
```

숫자를 입력하세요: 5

```
print(number)
```

5

❖ print의 사용예

```
a = 123
print(a)
b = "Python"
print(b)
c = [1, 2, 3]
print(c)
```

```
123
Python
[1, 2, 3]
```

❖ 큰따옴표(")로 둘러싸인 문자열은 + 연산과 동일하다

```
print("life" "is" "too short") # ①
print("life"+"is"+"too short") # ②
```

```
lifeistoo short
lifeistoo short
```

문자열 띄어쓰기는 콤마로 한다

```
print("life", "is", "too short")
```

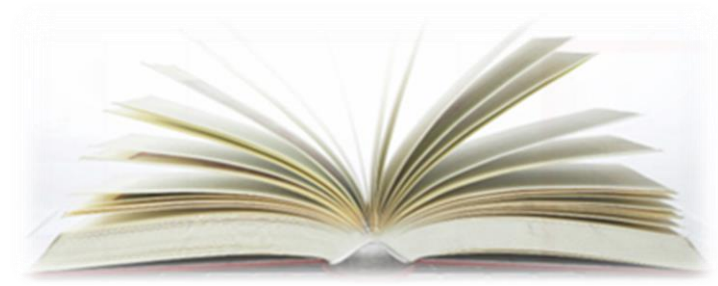
life is too short

한 줄에 곱갯값 출력하기

```
for i in range(10):  
    print(i, end=' ')
```

0 1 2 3 4 5 6 7 8 9

Subsection 4



데이터 저장

- ❖ 다음 소스 코드를 에디터로 작성해서 저장한 후 실행해 보자. 프로그램을 실행한 디렉터리에 새로운 파일이 하나 생성된 것을 확인할 수 있을 것이다.

```
f = open("FileTest.txt", 'w')  
f.close()
```

```
In [1]: f = open("FileTest.txt", 'w')  
        f.close()
```

jupyter

Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0	Name ↓	Last Modified	File size
<input type="checkbox"/>	backup	6일 전	
<input type="checkbox"/>	1일차.ipynb	Running 몇 초 전	36.9 kB
<input type="checkbox"/>	index.ipynb	6일 전	51.1 kB
<input type="checkbox"/>	FileTest.txt	2분 전	0 B
<input type="checkbox"/>	goorm.manifest	11분 전	637 B



- ❖ 파일을 생성하기 위해 우리는 파이썬 내장 함수 `open`을 사용했다. `open` 함수는 다음과 같이 "파일 이름"과 "파일 열기 모드"를 입력값으로 받고 결과값으로 파일 객체를 돌려준다.
- ❖ 파일 객체 = `open(파일 이름, 파일 열기 모드)`
- ❖ 파일 열기 모드에는 다음과 같은 것이 있다.

파일열기모드	설명
r	읽기모드 - 파일을 읽기만 할 때 사용
w	쓰기모드 - 파일에 내용을 쓸 때 사용
a	추가모드 - 파일의 마지막에 새로운 내용을 추가 시킬 때 사용

❖ 위 예에서는 파일을 쓰기 모드로 열기만 했지 정작 아무것도 쓰지는 않았다. 이번에는 에디터를 열고 프로그램의 출력 값을 파일에 직접 써 보자.

```
In [2]: f = open("FileTest.txt", 'w')
        for i in range(1, 11):
            data = "%d번째 줄입니다.\n" % i
            f.write(data)
        f.close()
```

readline() 함수 사용하기

❖ `f.open("FileTest.txt", 'r')`로 파일을 읽기 모드로 연 후 `readline()`을 사용해서 파일을 읽어 출력

```
In [3]: f = open("FileTest.txt", 'r')
        line = f.readline()
        print(line)
        f.close()
```

1번째 줄입니다.

```
In [5]: f = open("FileTest.txt", 'r')
        while True:
            line = f.readline()
            if not line: break
            print(line)
        f.close()
```

1번째 줄입니다.

2번째 줄입니다.

3번째 줄입니다.

4번째 줄입니다.

5번째 줄입니다.

6번째 줄입니다.

7번째 줄입니다.

8번째 줄입니다.

9번째 줄입니다.

10번째 줄입니다.

readlines 함수 사용하기

- ❖ **readlines** 함수는 파일의 모든 줄을 읽어서 각각의 줄을 요소로 갖는 리스트로 돌려준다. 따라서 위 예에서 **lines**는 리스트 ["1 번째 줄입니다.", "2 번째 줄입니다.", ..., "10 번째 줄입니다."]가 된다.

```
In [6]: f = open("FileTest.txt", 'r')
        lines = f.readlines()
        for line in lines:
            print(line)
        f.close()
```

1번째 줄입니다.

2번째 줄입니다.

3번째 줄입니다.

4번째 줄입니다.

5번째 줄입니다.

6번째 줄입니다.

7번째 줄입니다.

8번째 줄입니다.

9번째 줄입니다.

10번째 줄입니다.

read 함수 사용하기

❖ `f.read()`는 파일의 내용 전체를 문자열로 돌려준다.

```
In [7]: f = open("FileTest.txt", 'r')  
data = f.read()  
print(data)  
f.close()
```

```
1번째 줄입니다.  
2번째 줄입니다.  
3번째 줄입니다.  
4번째 줄입니다.  
5번째 줄입니다.  
6번째 줄입니다.  
7번째 줄입니다.  
8번째 줄입니다.  
9번째 줄입니다.  
10번째 줄입니다.
```

추가 모드('a')

- ❖ 쓰기 모드('w')로 파일을 열 때 이미 존재하는 파일을 열면 그 파일의 내용이 모두 사라지게 된다. 하지만 원래 있던 값을 유지하면서 단지 새로운 값만 추가해야 할 경우도 있다. 이런 경우에는 파일을 추가 모드('a')로 열면 된다.
- ❖ 파일.txt 파일을 추가 모드('a')로 열고 write를 사용해서 결괏값을 기존 파일에 추가해 적는 예이다. 여기에서 추가 모드로 파일을 열었기 때문에 새파일.txt 파일이 원래 가지고 있던 내용 바로 다음부터 결괏값을 적기 시작한다.

```
In [8]: f = open("FileTest.txt", 'a')
        for i in range(11, 20):
            data = "%d번째 줄입니다.\n" % i
            f.write(data)
        f.close()
```

```
In [9]: f = open("FileTest.txt", 'r')
        data = f.read()
        print(data)
        f.close()
```

```
1번째 줄입니다.
2번째 줄입니다.
3번째 줄입니다.
4번째 줄입니다.
5번째 줄입니다.
6번째 줄입니다.
```

- ❖ 파일을 열면 위와 같이 항상 close해 주는 것이 좋다. 하지만 이렇게 파일을 열고 닫는 것을 자동으로 처리할 수 있다면 편리하지 않을까? 파이썬의 with문이 바로 이런 역할을 해준다.

```
In [10]: f = open("NOwith.txt", 'w')  
f.write("Life is too short, you need python")  
f.close()
```

```
In [11]: with open("with.txt", "w") as f:  
f.write("Life is too short, you need python")
```

[Quit](#) [Logout](#)[Files](#) [Running](#) [Clusters](#)

Select items to perform actions on them.

[Upload](#) [New](#) [Refresh](#)

<input type="checkbox"/> 0		Name	Last Modified	File size
<input type="checkbox"/>	📁	backup	6일 전	
<input type="checkbox"/>	📄	1일차.ipynb	Running 1분 전	40.9 kB
<input type="checkbox"/>	📄	index.ipynb	6일 전	51.1 kB
<input type="checkbox"/>	📄	FileTest.txt	3분 전	428 B
<input type="checkbox"/>	📄	goorm.manifest	36분 전	637 B
<input type="checkbox"/>	📄	NOwith.txt	몇 초 전	34 B
<input type="checkbox"/>	📄	with.txt	몇 초 전	34 B



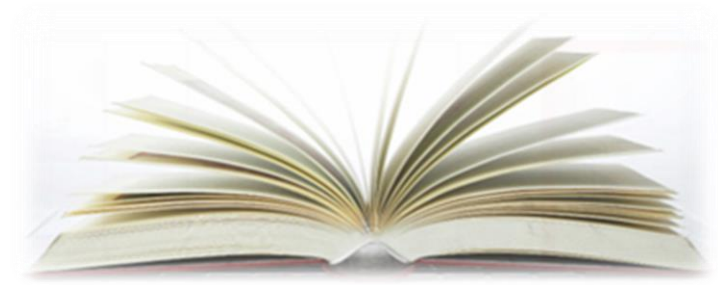
❖ Quiz 예

첫번째 숫자를 입력하세요:5
두번째 숫자를 입력하세요:15
두 수의 합은 20 입니다

```
input1 = int(input("첫번째 숫자를 입력하세요:"))  
input2 = int(input("두번째 숫자를 입력하세요:"))  
  
total = input1 + input2  
print("두 수의 합은 %s 입니다" % total)
```

첫번째 숫자를 입력하세요:5
두번째 숫자를 입력하세요:15
두 수의 합은 20 입니다

Unit A



참고자료

- ❖ <http://www.ncs.go.kr>
- ❖ NELLDAL/JOHN LEWIS 지음, 조영석/김대경/박찬영/송창근 역, 단계별로 배우는 컴퓨터과학, 홍릉과학출판사, 2018
- ❖ 혼자 공부하는 머신러닝+딥러닝 박해선 지음 | 한빛미디어 | 2020년 12월
- ❖ 머신러닝 실무 프로젝트 ,아리가 미치아키, 나카야마 신타, 니시바야시 다카시 지음 | 심효섭 옮김 | 한빛미디어 | 2018년 06월
- ❖ 파이썬을 활용한 머신러닝 쿡북 크리스 알본 지음 | 박해선 옮김 | 한빛미디어 | 2019년 09월
- ❖ 처음 배우는 머신러닝 김의중 지음 | 위키북스 | 2016년 07월
- ❖ 파이썬으로 배우는 머신러닝의 교과서 : 이토 마코토 지음 | 박광수(아크몬드) 옮김 | 한빛미디어 | 2018년 11월
- ❖ 기타 서적 및 웹 사이트 자료 다수 참조

❖ 리눅스 다운로드

```
!rm -rf bigStudy
```

```
!git clone 'https://github.com/looker2zip/bigStudy.git'
```

❖ 윈도우 다운로드

<https://github.com/looker2zip/bigStudy>

The screenshot shows the GitHub repository page for 'looker2zip/bigStudy'. The repository is public and has 4 commits, 0 stars, and 0 forks. The commit history is visible, showing the initial commit of README.md 12 days ago and a subsequent commit of dataset 7 days ago. The repository description is 'No description, website, or topics provided.' The page also includes navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights.

Commit Hash	Commit Message	Commit Date
9c012f5	Initial commit	12 days ago
9c012f5	Add files	7 days ago

```
%config InlineBackend.figure_format = 'retina'  
!apt -qq -y install fonts-nanum
```

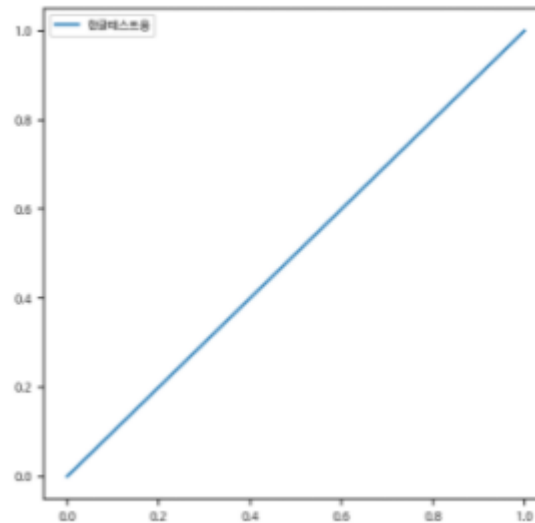
❖ 런타임 다시 시작

```
import matplotlib as mpl  
import matplotlib.pyplot as plt  
import matplotlib.font_manager as fm  
fontpath = '/usr/share/fonts/truetype/nanum/NanumBarunGothic.ttf'  
font = fm.FontProperties(fname=fontpath, size=9)  
plt.rc('font', family='NanumBarunGothic')  
mpl.font_manager._rebuild()
```

```
plt.figure(figsize=(5,5))  
plt.plot([0,1], [0,1], label='한글테스트용')  
plt.legend()  
plt.show()
```



```
plt.figure(figsize=(5,5))  
plt.plot([0,1], [0,1], label='한글테스트용')  
plt.legend()  
plt.show()
```





감사합니다.

- ❖ Mobile: 010-9591-1401
- ❖ E-mail: onlooker2zip@naver.com