



데이터 분석 환경 구축

"hello world"



긴 여정

긴 여정

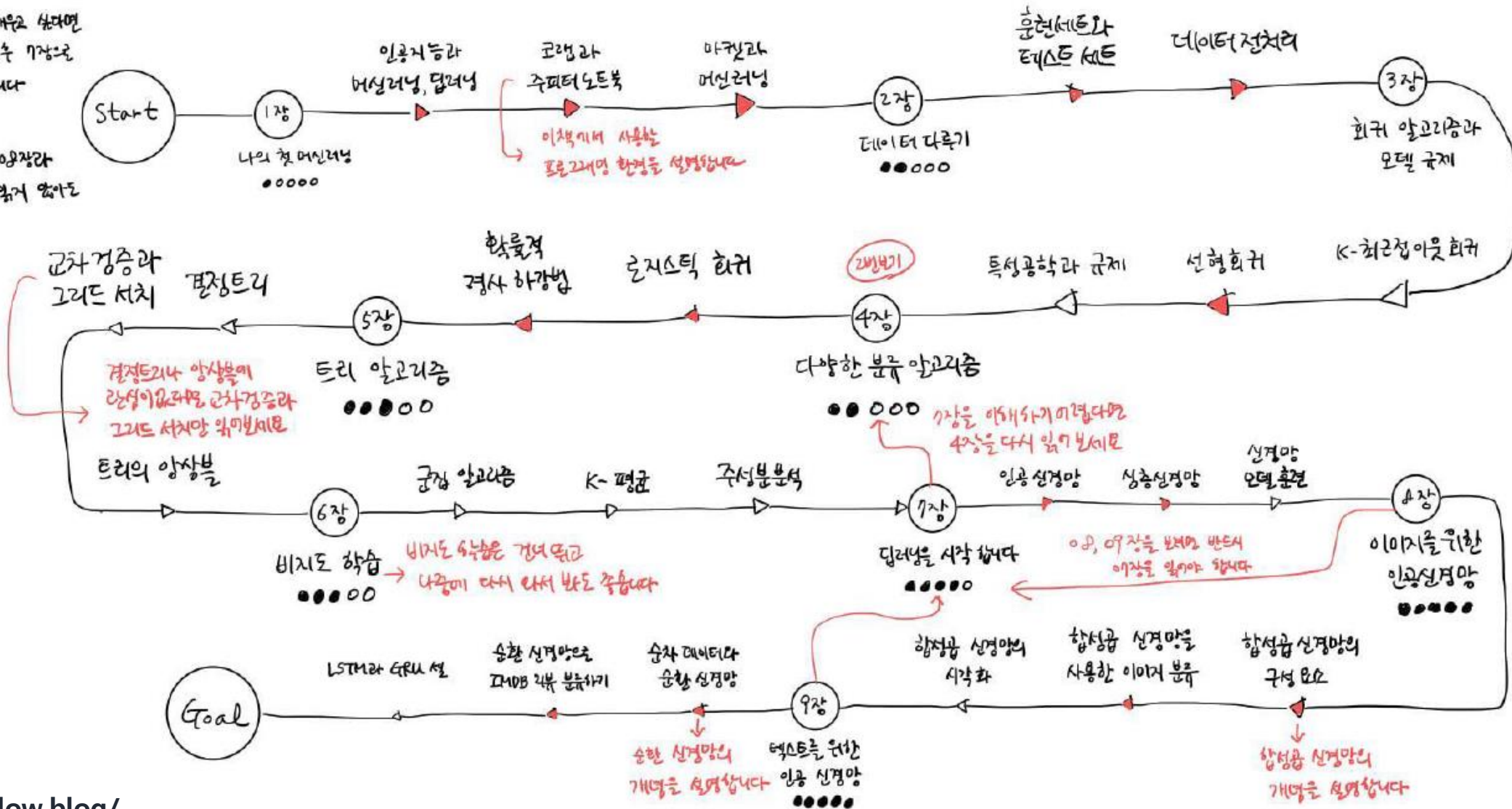
▷ : 건너뛰어도 좋은 것 ◀ : 꼭 읽고 넘어가야 할 것

머신러닝 이~ 06장

답러닝만 먼저 배우고 싶다면
이~04장을 읽은 후 7장으로
건너 뛰어도 좋습니다

답러닝 이~09장

이장을 읽은 후 10장과
11장을 순서대로 읽지 않아도
괜찮습니다



INDEX

데이터 과학, 데이터 분석

통계

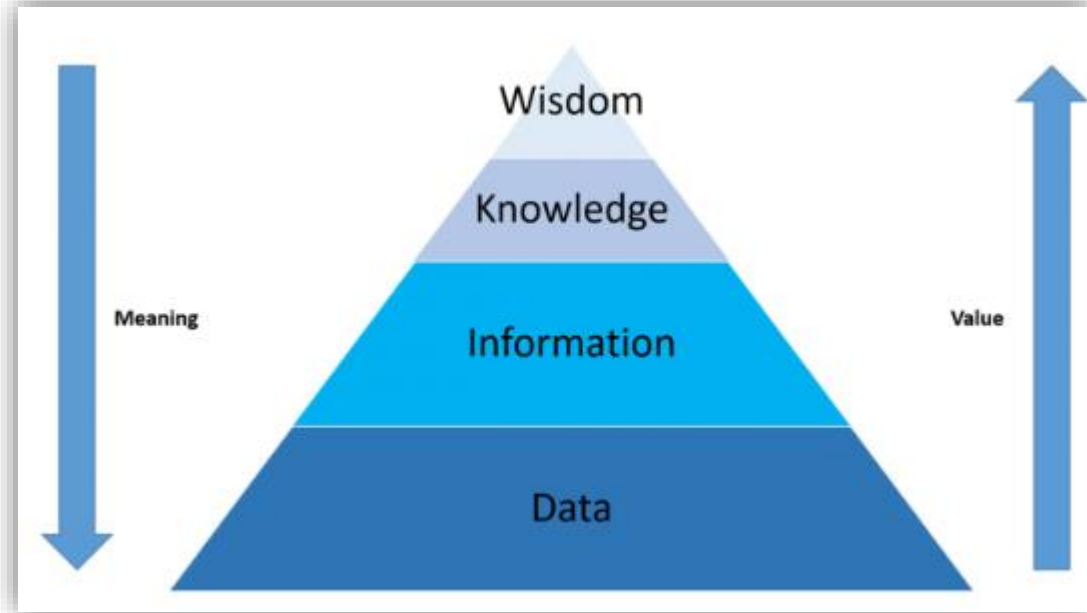
프로그래밍 언어

구글 코랩과 주피터 노트북

파이썬

데이터 분석

데이터, 정보, 지식



데이터 분석

데이터

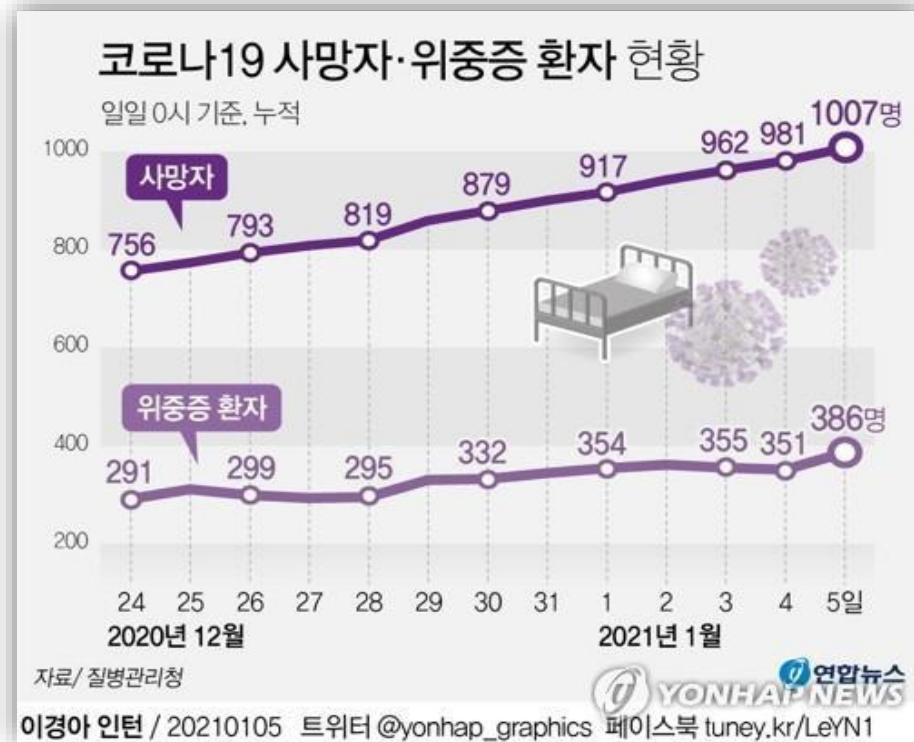
- 관찰이나 측정을 통해 수집된 사실이나 값들
- 사실들 그 자체에 대한 일차적인 표현
- 단순한 숫자, 문자, 기호 등으로 구성
- ex, "10, 20, 30"

구분	국내발생	해외유입
서울	11,531	3
부산	3,011	4
대구	2,497	0
인천	3,745	5
광주	1,422	3
대전	1,423	1
울산	795	0
세종	332	0
경기	17,976	51
강원	1,087	4
충북	1,156	0
충남	1,587	2
전북	1,507	3
전남	1,241	7
경북	1,926	2
경남	2,192	6
제주	476	2

데이터 분석

정보

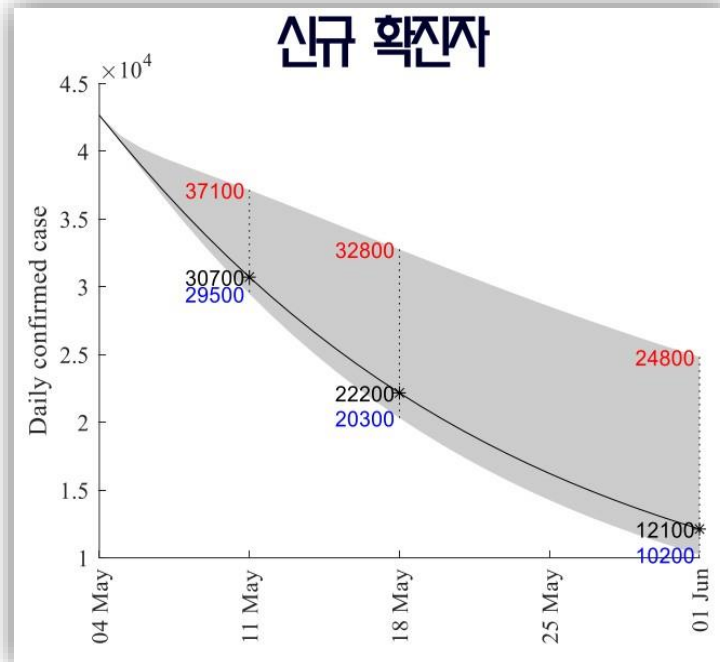
- 관련성과 목적을 갖춘 데이터 (peter Drucker)
- 데이터를 특정한 맥락에서 해석하거나 처리하여 의미를 부여한 것
- 데이터에 구조와 의미를 부여하여 사용자에게 유용한 형태로 변환된 결과
- ex, "10도, 20도, 30도" 오늘의 기온, 데이터 -> 정보



데이터 분석

지식

- 정보를 바탕으로 이해, 경험, 학습 등을 통해 축적된 통찰이나 원칙
- 특정 문제를 해결하거나 결정을 내리는 데 사용할 수 있는 심층적인 이해 제공
- ex, "기온이 30도를 넘으면 열사병에 걸릴 위험이 높아진다" 정보 -> 지식

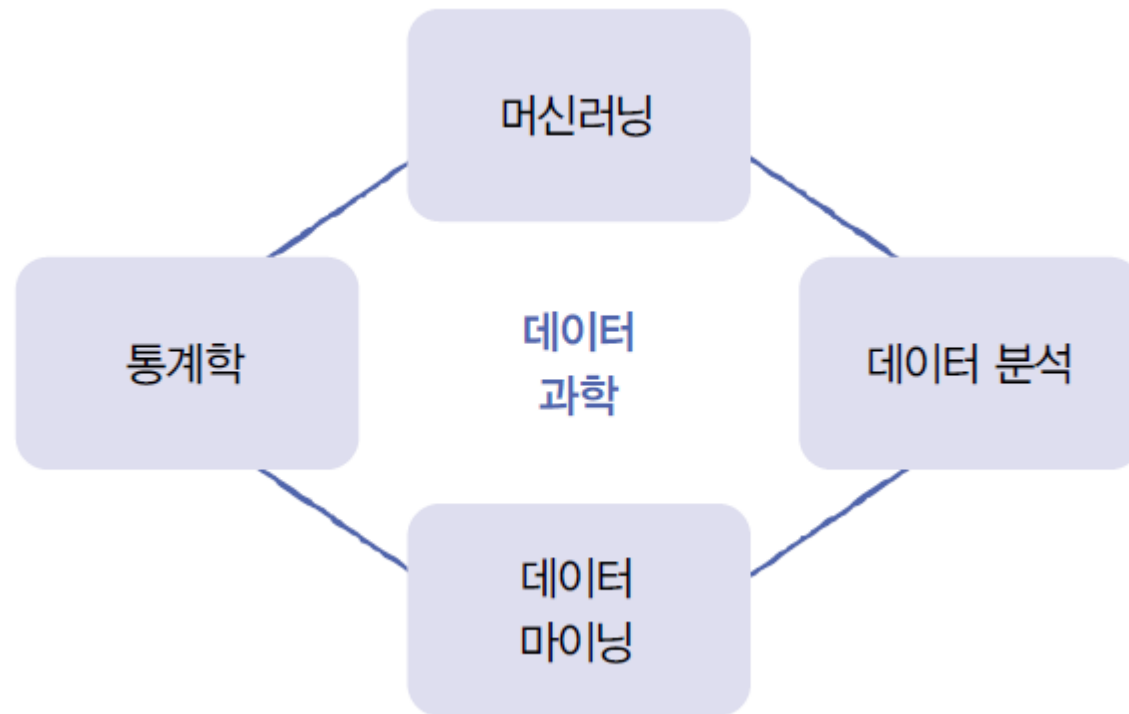


데이터 분석

데이터 분석과 데이터 과학

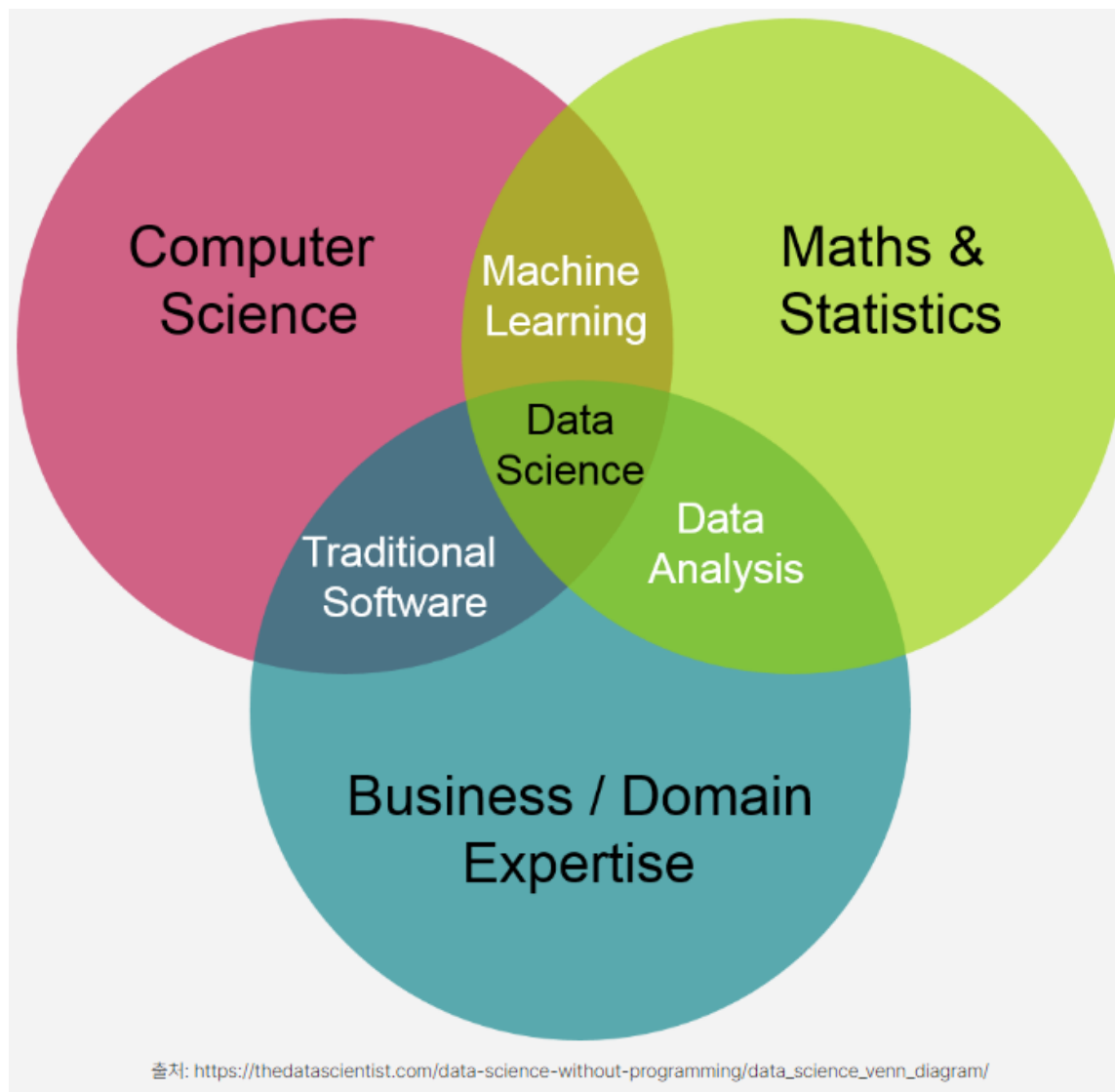
- 데이터 과학

통계학(statistics), 데이터 분석, 머신러닝(machine learning), 데이터 마이닝(data mining) 등을 아우르는 큰 개념



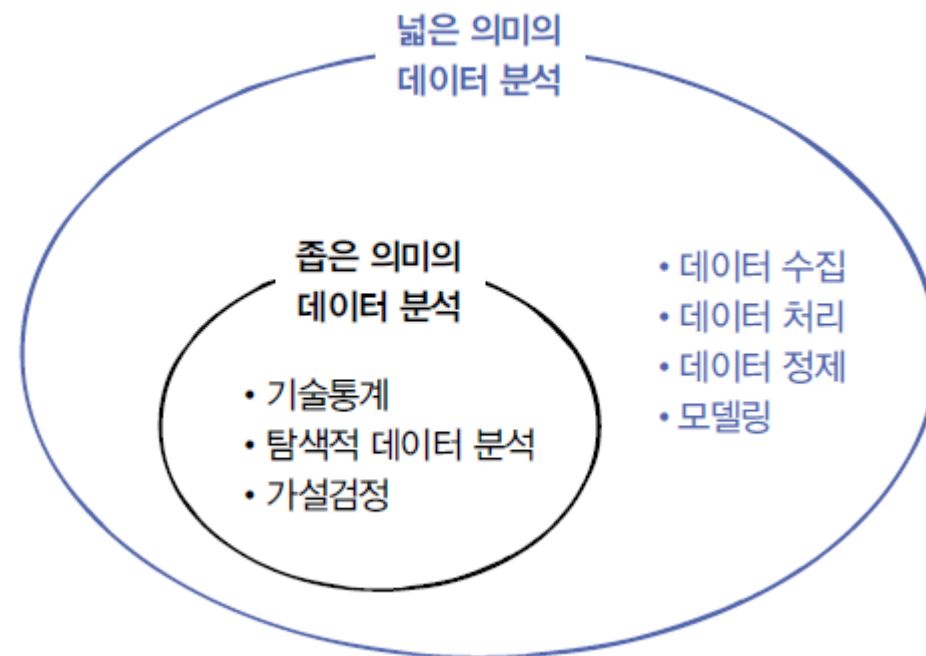
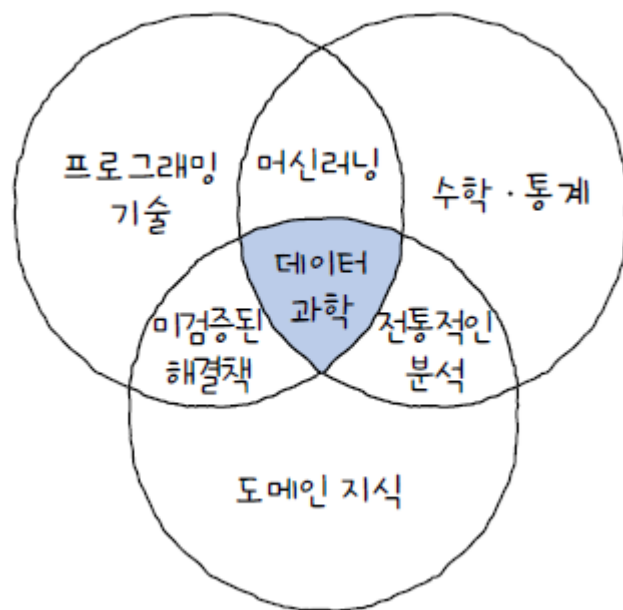
데이터 분석

데이터 분석과 데이터 과학



데이터 분석

데이터 분석가



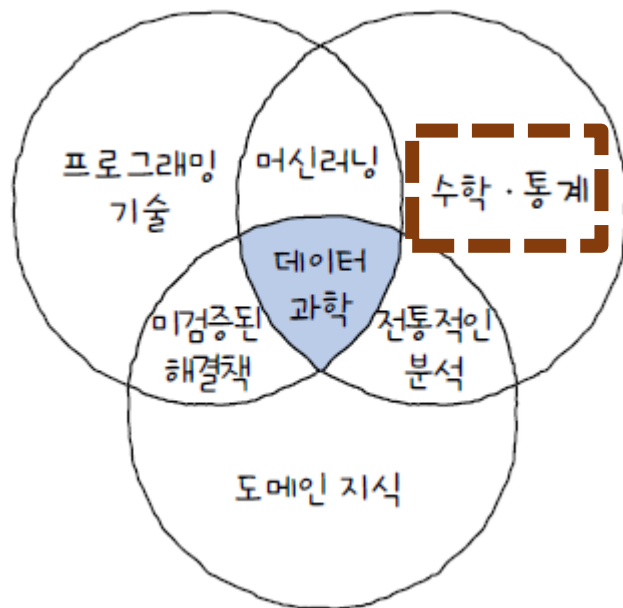
드류 콘웨이의 데이터 과학 벤 다이어그램

<http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>

데이터 분석

데이터 분석가

어는 정도까지???



- 가중치 업데이트 (cont'd)

- ② $\frac{\partial \hat{y}_i}{\partial Z_{out}}, \hat{y} = \sigma(Z_{out}) = \frac{1}{1+e^{-Z_{out}}}$

- How to differentiate?

- Then, we get

- $\frac{\partial \hat{y}_i}{\partial Z_{out}} = \frac{e^{-Z_{out}}}{(1+e^{-Z_{out}})^2} = \frac{1}{1+e^{-Z_{out}}} \left(1 - \frac{1}{1+e^{-Z_{out}}}\right) = \hat{y}_i(1 - \hat{y}_i)$

드류 콘웨이의 데이터 과학 벤 다이어그램

<http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram>

데이터 분석

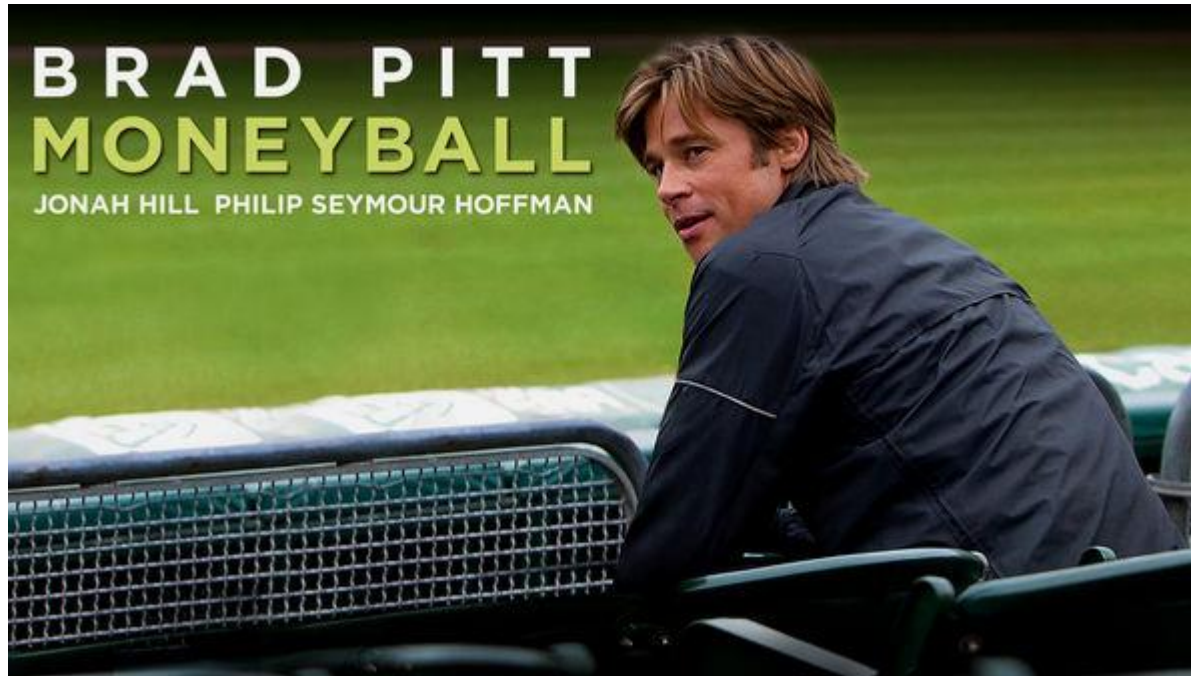
데이터 분석가



영화 아마겟돈

데이터 분석

데이터 분석가



데이터 분석

데이터 분석 프로세스

CRISP-DM

Cross-Industry Standard Process
for Data Mining

데이터 마이닝 작업을
안내하기 위해
업계에서 검증된 방법

Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives <i>Background Business Objectives Business Success Criteria</i>	Collect Initial Data <i>Initial Data Collection Report</i>	Select Data <i>Rationale for Inclusion/ Exclusion</i>	Select Modeling Techniques <i>Modeling Technique Modeling Assumptions</i>	Evaluate Results <i>Assessment of Data Mining Results w.r.t. Business Success Criteria Approved Models</i>	Plan Deployment <i>Deployment Plan</i>
Assess Situation <i>Inventory of Resources Requirements, Assumptions, and Constraints Risks and Contingencies Terminology Costs and Benefits</i>	Describe Data <i>Data Description Report</i>	Clean Data <i>Data Cleaning Report</i>	Generate Test Design <i>Test Design</i>	Review Process <i>Review of Process</i>	Plan Monitoring and Maintenance <i>Monitoring and Maintenance Plan</i>
Determine Data Mining Goals <i>Data Mining Goals Data Mining Success Criteria</i>	Explore Data <i>Data Exploration Report</i>	Construct Data <i>Derived Attributes Generated Records</i>	Build Model <i>Parameter Settings Models Model Descriptions</i>	Determine Next Steps <i>List of Possible Actions Decision</i>	Produce Final Report <i>Final Report Final Presentation</i>
Produce Project Plan <i>Project Plan Initial Assessment of Tools and Techniques</i>	Verify Data Quality <i>Data Quality Report</i>	Integrate Data <i>Merged Data</i>	Assess Model <i>Model Assessment Revised Parameter Settings</i>		Review Project <i>Experience Documentation</i>
	Format Data <i>Reformatted Data</i>	Format Data <i>Reformatted Data</i>			
		<i>Dataset Dataset Description</i>			

CRISP-DM tasks in **bold**, and outcomes in *italic* (table from [CRISP-DM Guide](#))

데이터 분석

인공지능, 기계학습, 딥러닝



꽃 - 김춘수

내가 그의 이름을 불러주기 전에는
그는 다만
하나의 몸짓에 지나지 않았다.

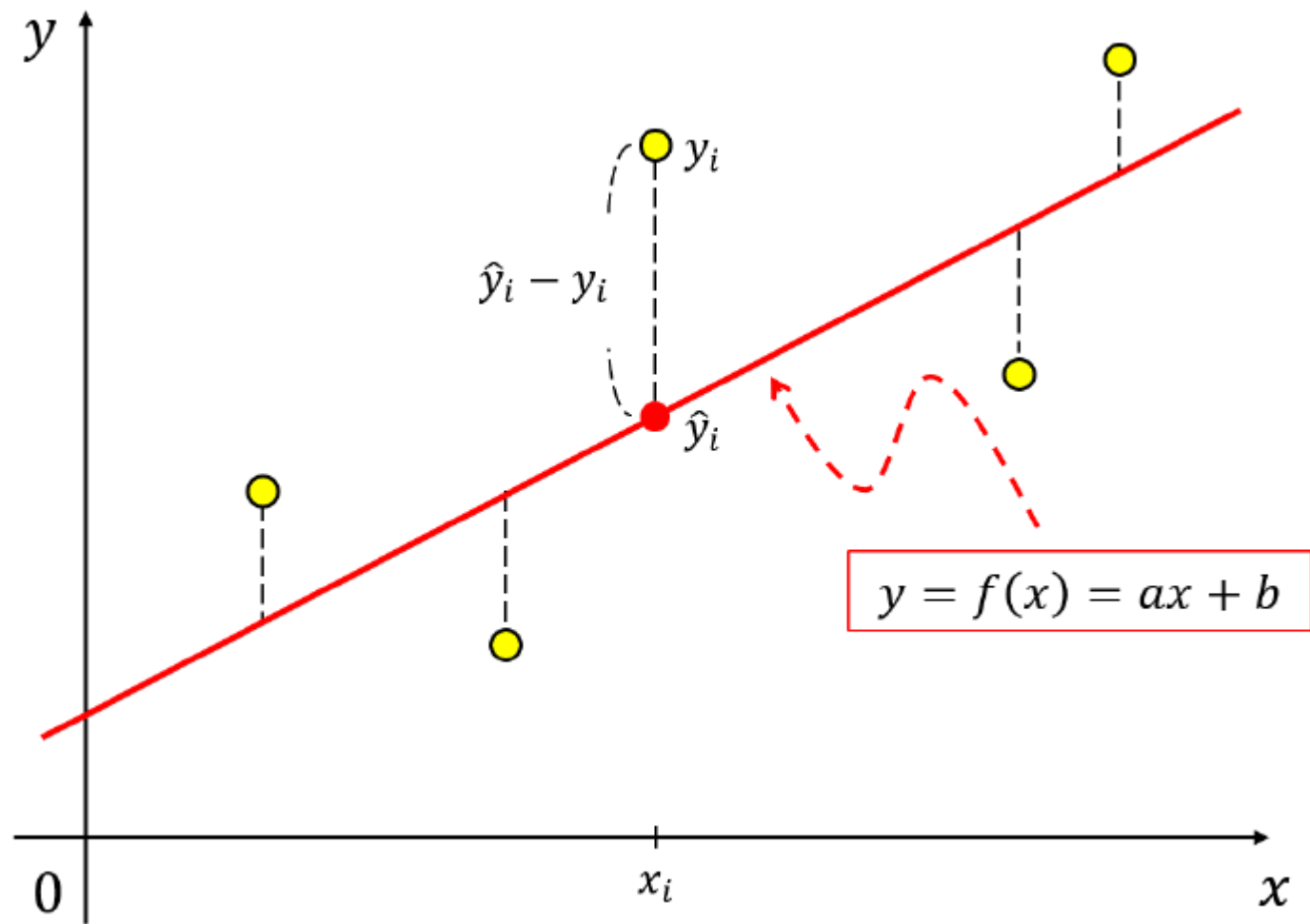
내가 그의 이름을 불러주었을 때,
그는 나에게로 와서
꽃이 되었다.

내가 그의 이름을 불러준 것처럼
나의 이 빛깔과 향기에 알맞는
누가 나의 이름을 불러다오.
그에게로 가서 나도
그의 꽃이 되고 싶다.

우리들은 모두
무엇이 되고 싶다.
너는 나에게 나는 너에게
잊혀지지 않는 하나의 눈짓이 되고 싶다.

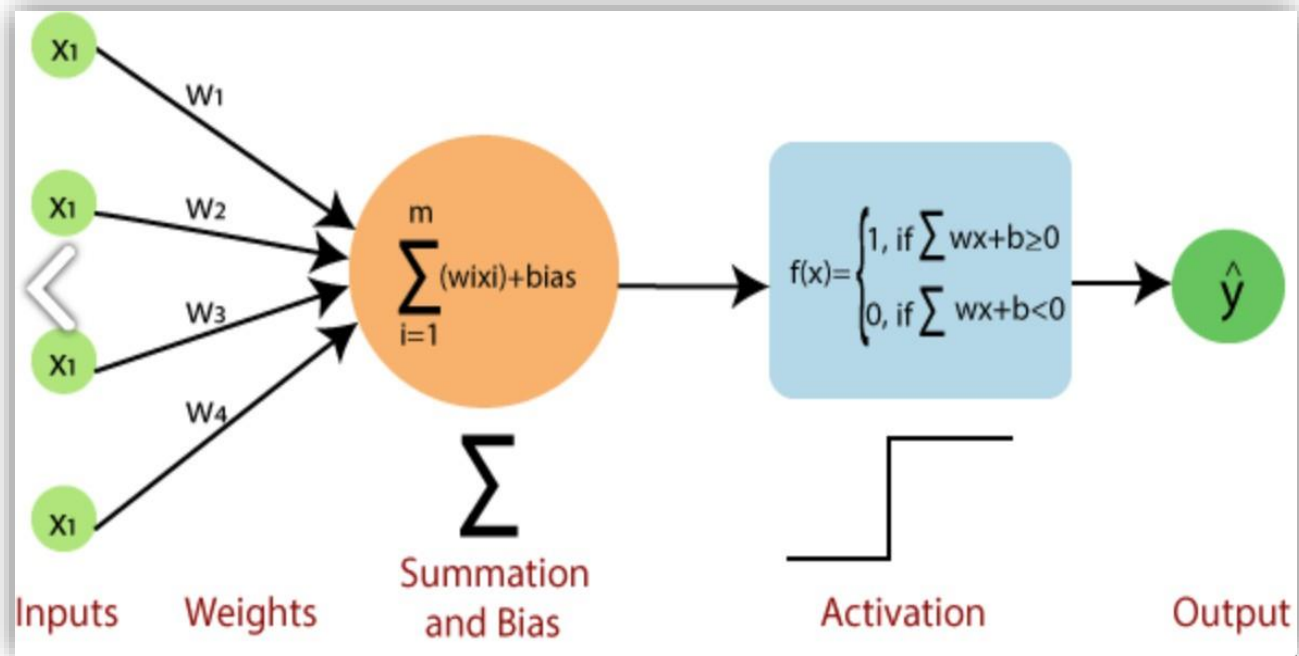
데이터 분석

학습



데이터 분석

학습



통계

Statistics 데이터를 수집, 분석, 해석, 표현하는 과정을 포함하는 학문

1. 한 곳에 몰아서 어림잡아 계산함.
2. 어떤 현상을 종합적으로 한눈에 알아보기 쉽게 일정한 체계에 따라 숫자로 나타냄. 또는 그런 것 - wiki



통계

- **모집단 (Population)**

연구에서 관심 있는 전체 집단

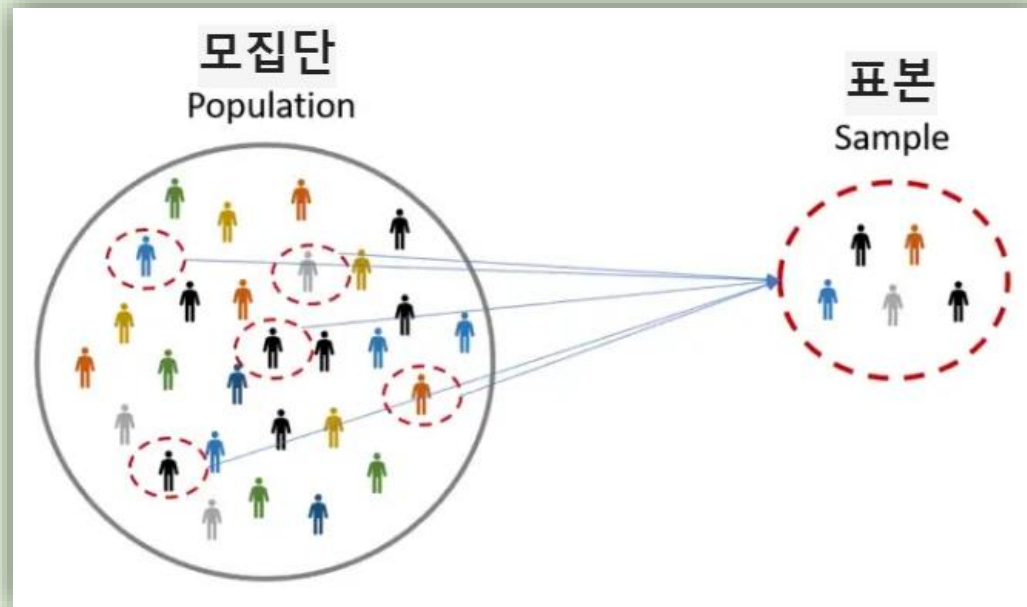
ex, 예를 들어, 한국의 모든 성인을 연구 대상으로 한다면, 모집단은 한국의 모든 성인

- **표본 (Sample)**

모집단에서 추출된 부분 집합.

전체 모집단을 조사하기 힘들 때, 표본을 조사하여 모집단에 대한 추론 진행

ex, 한국의 성인 1,000명을 조사하여 전체 성인의 특성 추론



<https://brunch.co.kr/@saetae/101>

통계

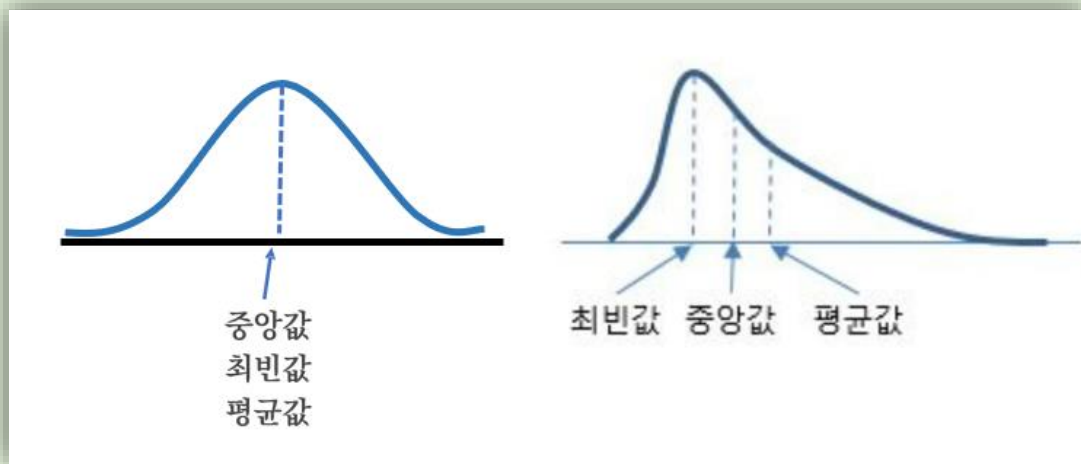
- **평균 (Mean)**

데이터 값들의 합을 데이터의 개수로 나눈 값
데이터의 중심을 대표하는 값

- **중앙값 (Median)**

데이터를 크기 순서대로 정렬했을 때 중앙에 위치한 값

중앙값은 극단값(outlier)의 영향을 받지 않기 때문에 데이터의 중심을 나타내는 또 다른 대표 값으로 사용



통계

- **분산 (Variance)**
데이터가 평균으로부터 얼마나 흩어져 있는지를 나타내는 지표. 편차 제곱의 평균
분산이 크다는 것은 데이터가 평균으로부터 크게 흩어져 있다는 것을 의미
- **표준편차 (Standard Deviation)**
분산값을 제곱근으로 줄임, 데이터의 변동성을 측정하는 데 사용
표준편차가 작을수록 데이터가 평균에 가깝게 모여 있음을 의미
- **확률 (Probability)**
어떤 사건이 발생할 가능성을 나타내는 값으로, 0에서 1 사이의 값을 가짐
ex, 동전을 던졌을 때 앞면이 나올 확률 0.5
- **가설검정 (Hypothesis Testing)**
모집단에 대한 가설을 세우고, 표본 데이터를 통해 그 가설이 맞는지를 검증하는 과정
연구 결과의 신뢰성을 평가하는 중요한 과정
- **신뢰구간 (Confidence Interval)**
모집단의 평균이나 비율이 특정 구간 내에 있을 것이라는 신뢰도를 나타내는 범위
ex, "95% 신뢰구간"은 95%의 확률로 그 구간에 참된 모집단의 평균이 포함된다는 의미

평균, 분산, 표준편차

Start	1번	2번	3번	4번	5번
1년량	175	177	179	181	183
<div>평균</div> <div>$(\frac{\text{1년량의 합}}{\text{인원수}})$</div>	$= \frac{175+177+179+181+183}{5} = 179$				
<div>편차</div> <div>$(\text{1년량} - \text{평균})$</div>	$175 - 179$ $= -4$	$177 - 179$ $= -2$	$179 - 179$ $= 0$	$181 - 179$ $= 2$	$183 - 179$ $= 4$
<div>편차제곱</div> <div>$(\text{편차} * \text{편차})$</div>	$(-4) * (-4)$ $= 16$	$(-2) * (-2)$ $= 4$	$(0) * (0)$ $= 0$	$(2) * (2)$ $= 4$	$(4) * (4)$ $= 16$
<div>분산</div> <div>(편차제곱의 평균)</div>	$= \frac{16+4+0+4+16}{5} = \frac{40}{5} = 8$				
<div>표준편차</div> <div>$(\sqrt{\text{분산}})$</div>	$= \sqrt{8} \approx 2.828$				
결과	인원수=5, 평균=179, 분산=8, 표준편차=2.828				

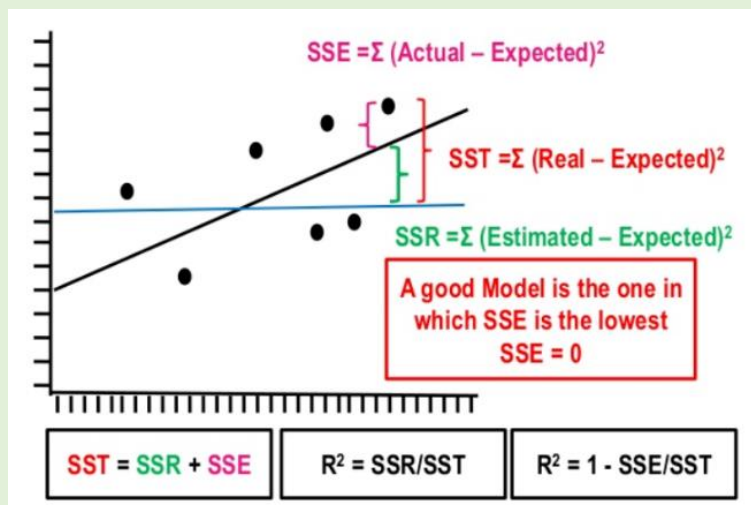
통계

- 상관계수 (Correlation)

- > 두 변수 간의 관계를 나타내는 지표
- > -1에서 1 사이의 값을 가지며, 1에 가까울수록 두 변수 간의 강한 양의 상관계수를, -1에 가까울수록 강한 음의 상관계수를 의미합니다.

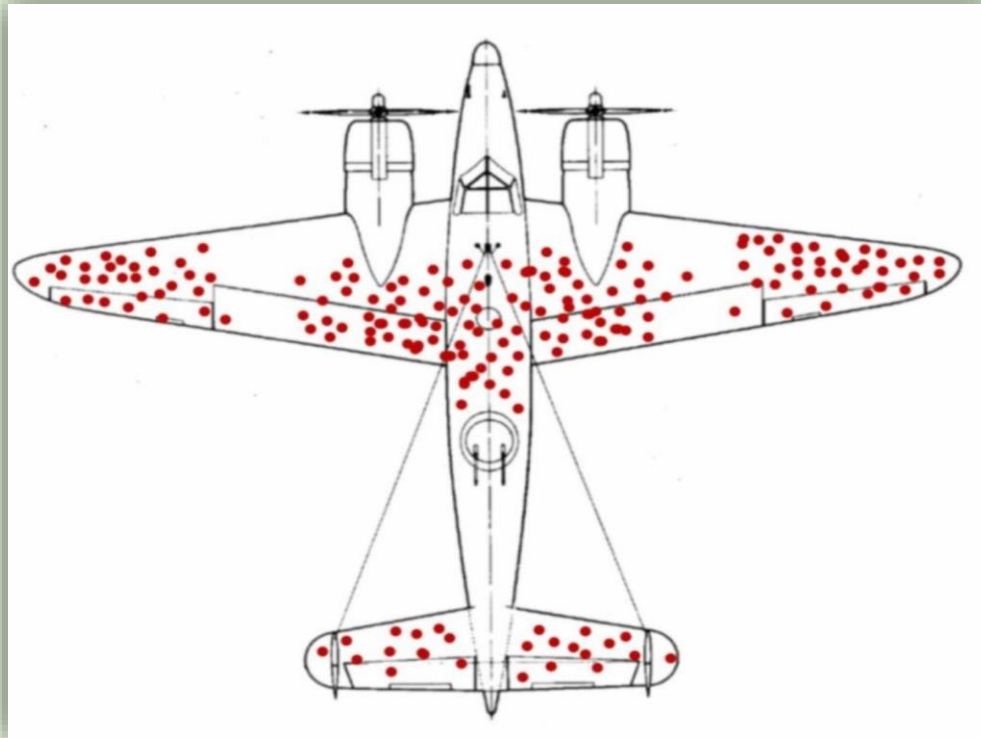
- 회귀분석 (Regression Analysis)

- > 하나 이상의 독립 변수가 종속 변수에 미치는 영향을 분석하는 통계 기법
- > 변수 간의 관계를 모델링하고 미래의 값을 예측할 수 있습니다.



통계

- 편향, 통계의 오류



- **프로그램(Program)**

- 컴퓨터와 스마트폰에 들어 있는 많은 소프트웨어
- 유튜브, 메시지, 전화, 카카오톡, 엑셀, 한글, 크롬, 메모장, VScode 등

≡ 소프트웨어, 애플리케이션, 앱(App)

프로그래밍 언어

- **프로그래밍(Programming)**
 - 프로그램을 만드는 작업
- **프로그래밍 언어(Programming Language)**
 - 컴퓨터가 이해하는 언어
- **프로그래머(Programmer)**
 - 프로그래밍 언어를 사용하여 소프트웨어나 앱을 개발하는 사람

프로그래밍 언어 - 분류

1. Compiled Language (컴파일 언어)

- 소스 코드를 기계어로 번역하여 실행 가능한 바이너리 파일을 생성하는 언어
- 컴파일 이후에는 별도의 인터프리터 없이 직접 실행
- 컴파일 단계에서 오류 수정 가능. 빠른 실행 속도
- **C, C++, Rust**

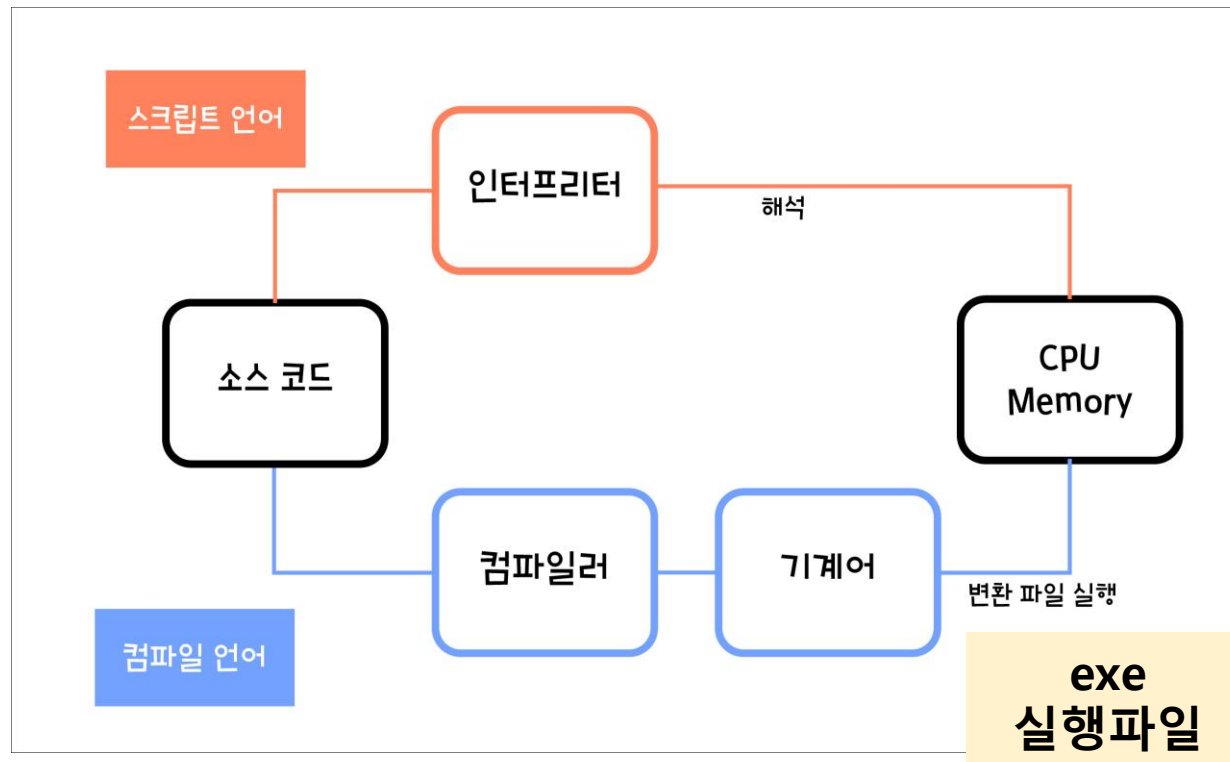
2. Script (스크립트) ≡ Interpreter

- 특정 작업을 자동화하기 위해 작성된 코드 파일
- 컴파일 없이, 인터프리터에 의해 한 줄씩 실행
- 언어는 개발 속도가 빠름, 간단한 작업을 수행하는 데 적합
- **Python, JavaScript, PHP**

프로그래밍 언어 - 분류

■ 컴파일러 언어 vs 스크립트 언어

- 컴파일러 언어 : 한번에 기계어로 번역하여 실행 프로그램 생성, 실행 시 스크립트 언어보다 속도가 빠름
- 스크립트 언어 : 상대적으로 빠른 언어 습득 가능



프로그래밍 언어 - 분류

▪ Rendering (렌더링)

- 텍스트, 이미지, 비디오, 3D 그래픽 등의 데이터를 시각적으로 처리하여 화면에 표시하는 과정
- 웹 개발에서는 HTML, CSS를 사용하여 웹 페이지의 콘텐츠를 화면에 그리는 작업을 의미
- 게임 개발에서는 3D 모델을 화면에 그리는 작업을 의미
- **HTML/CSS/JavaScript** (웹 페이지 렌더링)
- **C++/C#** (게임 개발에서의 그래픽 렌더링)
- **Python** (그래픽 처리와 시각화, 예: Matplotlib, Pygame)

▪ Parsing (파싱)

- 프로그래밍에서 "파싱"은 문자열로 된 데이터를 특정 형식으로 해석하고, 해당 데이터를 구조화된 형식으로 변환하는 과정을 의미
- 일반적으로 텍스트 데이터를 의미 있는 구조(예: 문법 구조, 트리 구조)로 변환하는 작업
- **HTML 파싱: 웹 브라우저 HTML 문서 -> 이를 기반으로 웹 페이지 렌더링**
- JSON 파싱: 애플리케이션 JSON 데이터 파싱 -> 사용 가능한 객체 or 자료 구조로 변환
- **Python, JavaScript, Java**

데이터 분석

데이터 분석을 위한 도구 - 파이썬

파이썬

귀도 반 로섬 (Guido van Rossum)

- 1989년 크리스마스로 연구실이 닫혀 취미활동으로 만든 프로그램 언어
- 1991년에 발표하였으며, 파이썬은 영국 코미디 그룹 "몬티 파이썬"에서 가져옴
- 2000년 Python 2, 2008년 Python 3 발표
- 2005년~2012년까지 구글
- 2013년 ~2019년 10월까지 드롭박스
- 은퇴 했다가 2020년 11월 13일 MS에 다시 입사함
- 구글에서 만들어진 SW 50% 이상이 파이썬
- 드롭박스(Dropbox), 인스타그램(Instagram)
- 간결하여 공동 작업과 유지보수가 매우 쉽고 편함



데이터 분석

데이터 분석을 위한 도구 - 파이썬

왜 파이썬?

배우기 쉽다! Easy!!

```
public class HelloWorld {  
    public static void main(String args[]) {  
        System.out.println("Hello World");  
    }  
}
```

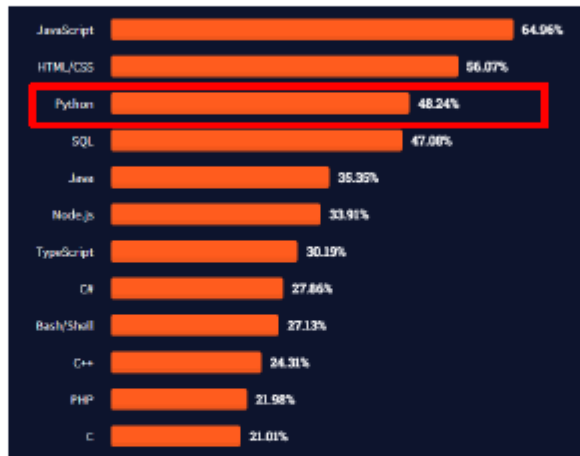
<JAVA의 "Hello World">

```
print("hello World")
```

<Python의 "Hello World">

- 문법이 간단하여 배우기 쉽다.
- 인간다운 언어이다.
- 파이썬은 간결하다

다양한 분야에서 활용됨



- 웹 프로그래밍 (웹 서버)
- GUI 프로그래밍
- 사물인터넷 (IoT)
- 데이터 분석
- 인공지능

OS에 영향이 적음



- 대부분의 프로그래밍 언어는 운영체제에 제한적임
- 여러 운영체제에 사용 가능하지만 사용방법이 다름
- 파이썬은 대부분의 OS에서도 동일한 결과를 얻을 수 있음

프로그래밍 언어 - 통계, 분석

데이터 분석을 위한 도구 - 파이썬

- 프로그래밍 언어
 - 파이썬 - 귀도 반 로섬 Guido van Rossum이 1991년에 만든 범용 프로그래밍 언어
 - R - 1995년에 통계 계산을 위해 개발된 언어
- 프로그래밍 환경: 구글 코랩
- 파이썬 필수 패키지

• 넘파이  NumPy











• 판다스  pandas

• 맷플롯립  matplotlib

• 사이파이  SciPy

• 사이킷런  scikit-learn

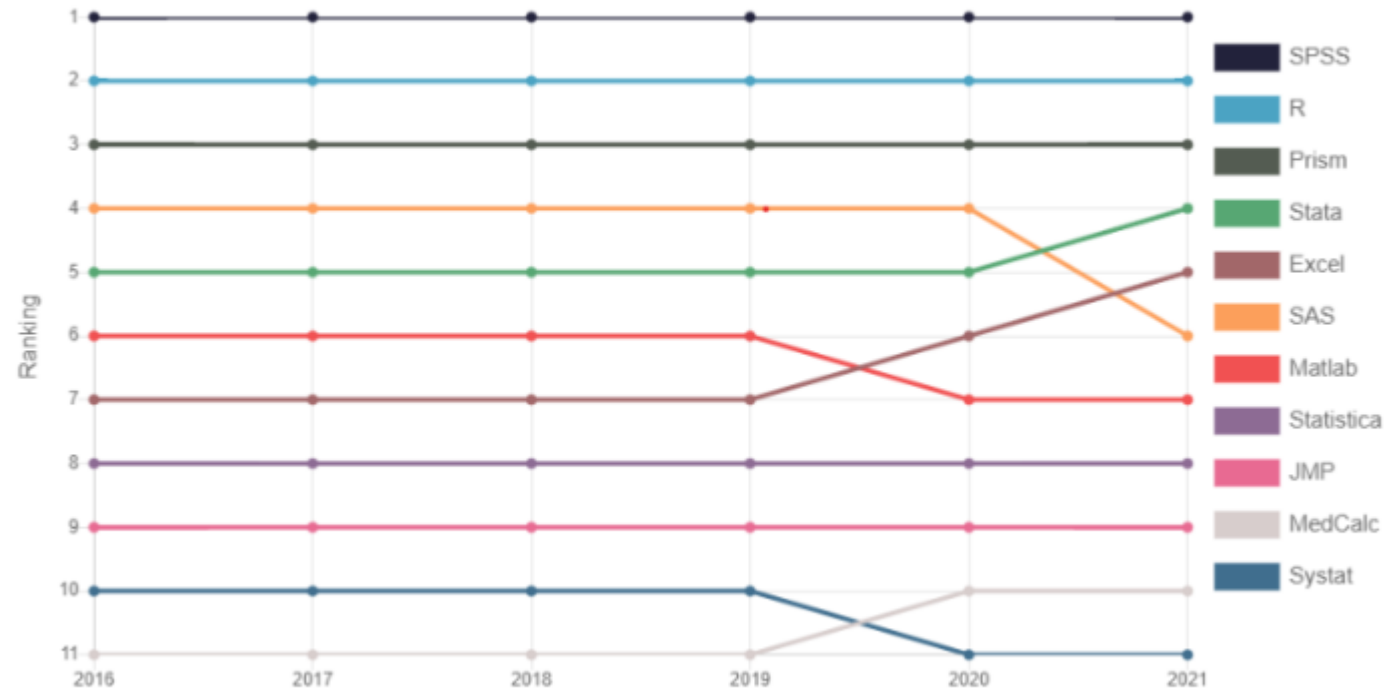
프로그래밍 언어 - 점유율

Mar 2023	Mar 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.83%	+0.57%
2	2			C	14.73%	+1.67%
3	3			Java	13.56%	+2.37%
4	4			C++	13.29%	+4.64%
5	5			C#	7.17%	+1.25%
6	6			Visual Basic	4.75%	-1.01%
7	7			JavaScript	2.17%	+0.09%
8	10	⬆		SQL	1.95%	+0.11%
9	8	⬇		PHP	1.61%	-0.30%
10	13	⬆		Go	1.24%	+0.26%

통계 소프트웨어

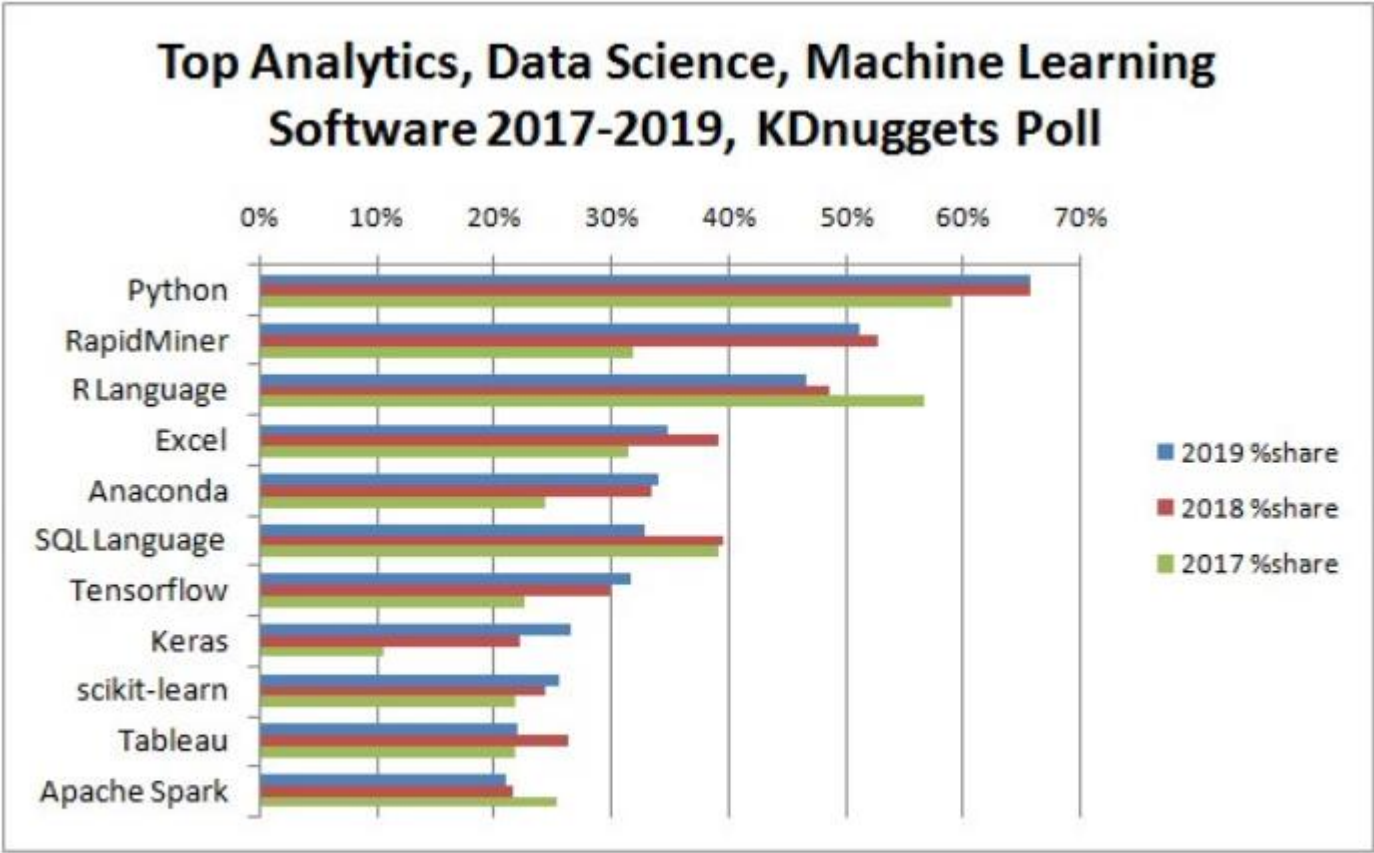
Statistical Software Popularity in 40,582 Research Papers

By [George Choueiry](#) - PharmD, MPH



<https://quantifyinghealth.com/statistical-software-popularity-in-research/>

2022 6월 PyPL
(Popularity of Programming Language) 순위



Rank	Change	Language	Share	Trend
1		Python	27.61 %	-2.8 %
2		Java	17.64 %	-0.7 %
3		JavaScript	9.21 %	+0.4 %
4		C#	7.79 %	+0.8 %
5		C/C++	7.01 %	+0.4 %
6		PHP	5.27 %	-1.0 %
7		R	4.26 %	+0.5 %
8	↑↑↑	TypeScript	2.43 %	+0.7 %
9	↓	Objective-C	2.21 %	+0.1 %
10	↓	Swift	2.17 %	+0.4 %

2017~2019 데이터분석, 데이터과학, 기계학습 소프트웨어 점유율 설문 조사 [출처 : KDnuggets]

프로그래밍 언어 - 점유율











Jan 2024	Jan 2023	Change	Programming Language	Ratings	Change
1	1		 Python	13.97%	-2.39%
2	2		 C	11.44%	-4.81%
3	3		 C++	9.96%	-2.95%
4	4		 Java	7.87%	-4.34%
5	5		 C#	7.16%	+1.43%
6	7	▲	 JavaScript	2.77%	-0.11%
7	10	▲	 PHP	1.79%	+0.40%
8	6	▼	 Visual Basic	1.60%	-3.04%
9	8	▼	 SQL	1.46%	-1.04%
10	20	▲▲	 Scratch	1.44%	+0.86%

그림 1. 2024년 1월 TIOBE 프로그래밍 언어 순위 (출처: <https://www.tiobe.com/tiobe-index/>)

파이썬 개발환경

colab 구글 코랩과 주피터 노트북

특징	구글 코랩	주피터 노트북
설치	클라우드 기반이므로 설치 불필요	내 컴퓨터에 설치
실행 환경	인터넷이 연결된 웹 브라우저	오프라인 상태의 웹 브라우저
파일 저장	구글 드라이브에 자동 저장	내 컴퓨터 하드 디스크에 저장
패키지	사전 설치되어 제공	패키지 관리자로 직접 설치

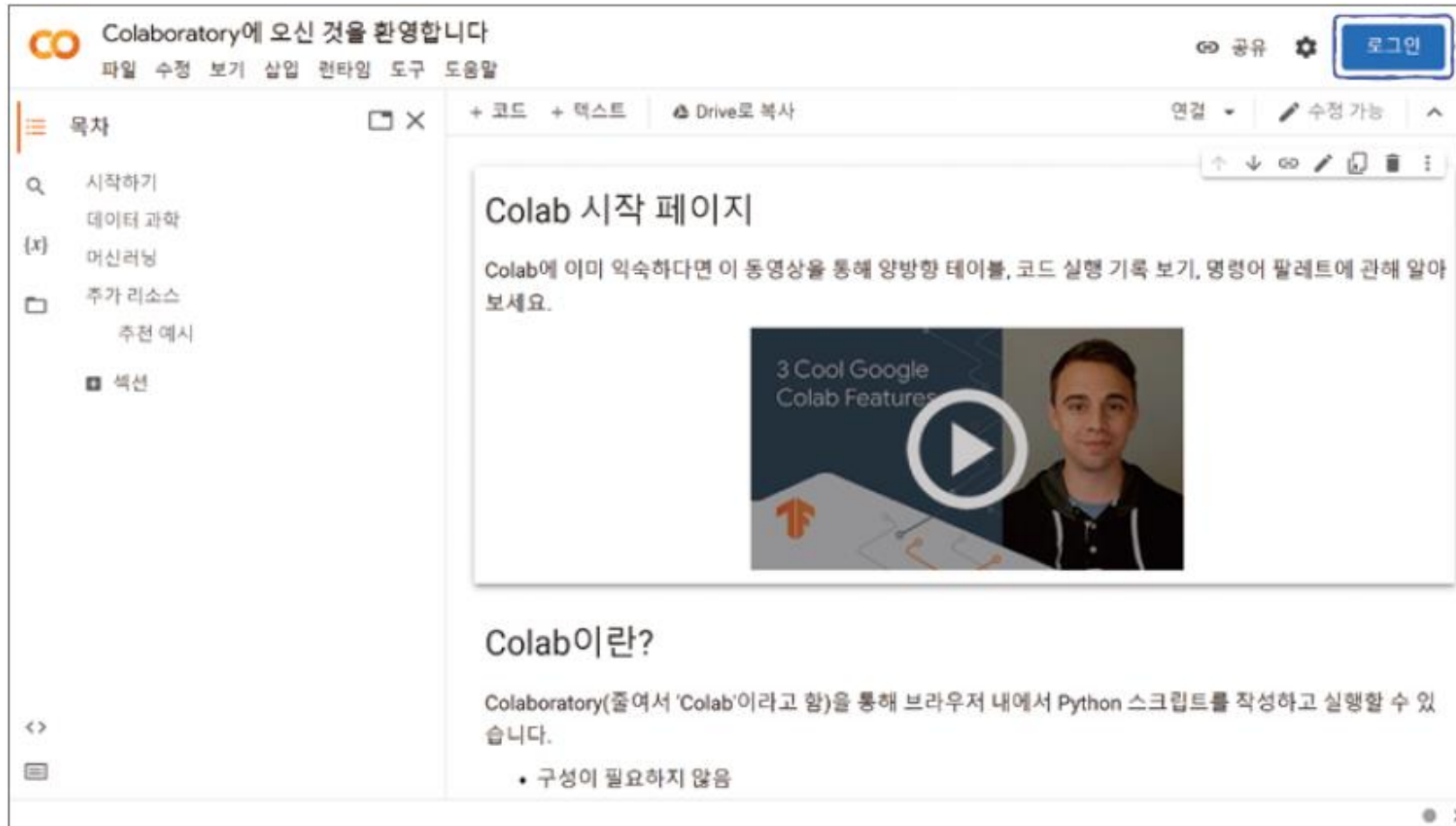
colab 구글 코랩과 주피터 노트북

구글 코랩

- 구글 코랩 웹 페이지에 접속

구글 코랩 웹 페이지의 오른쪽 상단에서 [로그인] 버튼을 클릭하여 구글 계정에 로그인

<https://colab.research.google.com/?hl=ko>



구글 코랩 웹 시작 페이지



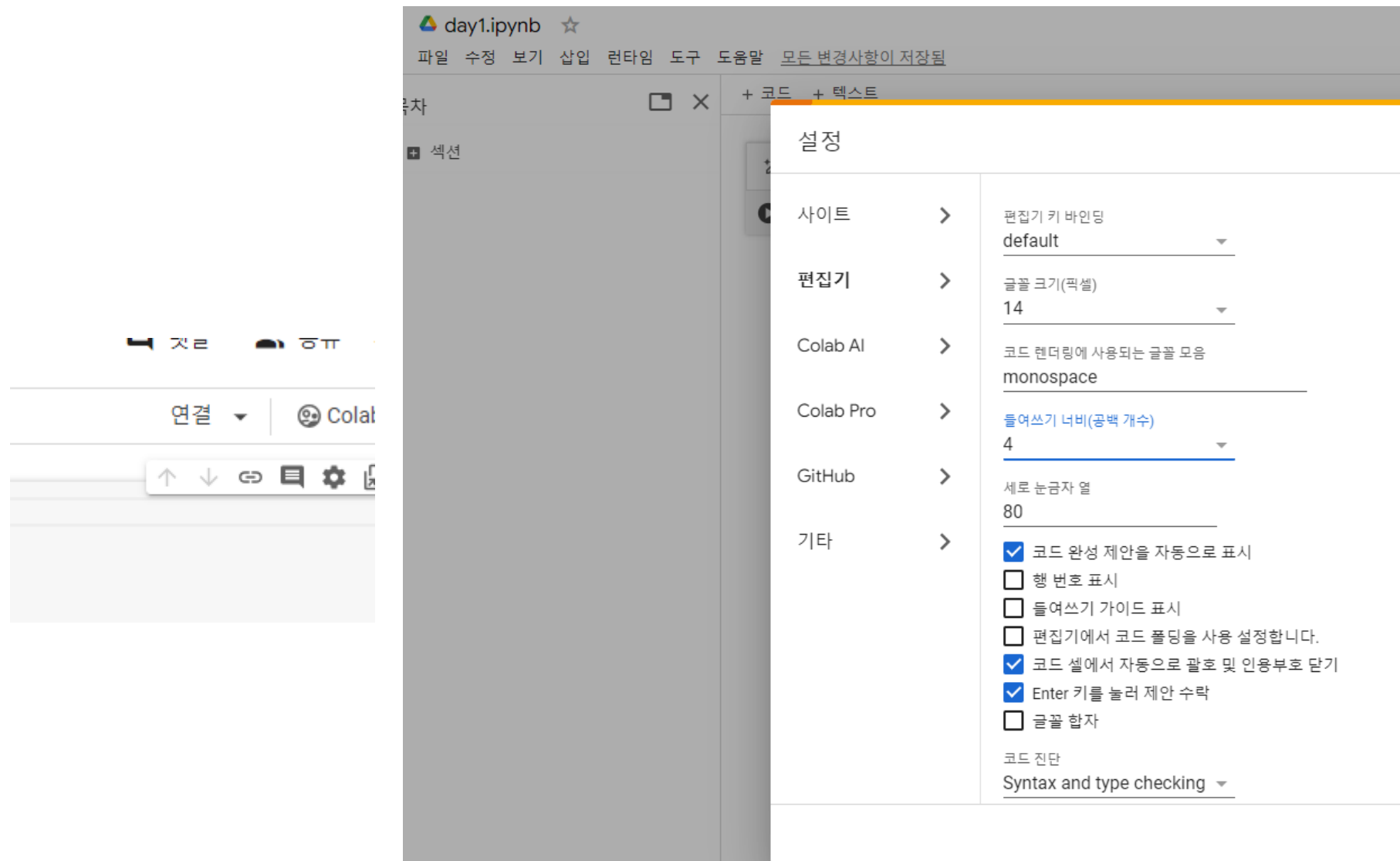
colab 구글 코랩과 주피터 노트북

The screenshot shows the Google Colaboratory interface with several handwritten annotations in Korean:

- Colaboratory에 오신 것을 환영합니다** (Welcome to Colaboratory) is highlighted with a box and an arrow pointing to **노트북 제목** (Notebook title).
- 파일 수정 보기 삽입 런타임 도구 도움말** (File Edit View Insert Runtime Tools Help) is highlighted with a box and an arrow pointing to **메뉴바** (Menu bar).
- 목차** (Table of contents) is highlighted with a box and an arrow pointing to **목차 닫기** (Close table of contents).
- 시작하기** (Getting started) is highlighted with a box and an arrow pointing to **시작하기** (Getting started).
- 코드 셀** (Code cell) is highlighted with a box and an arrow pointing to the first code cell.
- 텍스트 셀** (Text cell) is highlighted with a box and an arrow pointing to the second text cell.
- 선택된 셀은 음영으로 표시됩니다.** (Selected cell is highlighted) is written next to the second text cell.

The interface includes a sidebar with a search bar, a list of notebooks, and a table of contents. The main area displays a notebook titled "시작하기" (Getting started) with two cells: a code cell and a text cell. The code cell contains a Python script that calculates the number of seconds in a day and the number of seconds in a week. The text cell contains instructions on how to run the code and use variables.

colab 구글 코랩과 주피터 노트북



The image shows a Google Colab interface with the settings menu open. The settings menu is titled '설정' and contains the following sections and options:

- 사이트**: 편집기 키 바인딩 (default)
- 편집기**: 글꼴 크기(픽셀) (14)
- Colab AI**: 코드 렌더링에 사용되는 글꼴 모음 (monospace)
- Colab Pro**: 들여쓰기 너비(공백 개수) (4)
- GitHub**: 세로 눈금자 열 (80)
- 기타**:
 - ☒ 코드 완성 제안을 자동으로 표시
 - ☐ 행 번호 표시
 - ☐ 들여쓰기 가이드 표시
 - ☐ 편집기에서 코드 폴딩을 사용 설정합니다.
 - ☒ 코드 셀에서 자동으로 괄호 및 인용부호 닫기
 - ☒ Enter 키를 눌러 제안 수락
 - ☐ 글꼴 합자
- 코드 진단**: Syntax and type checking

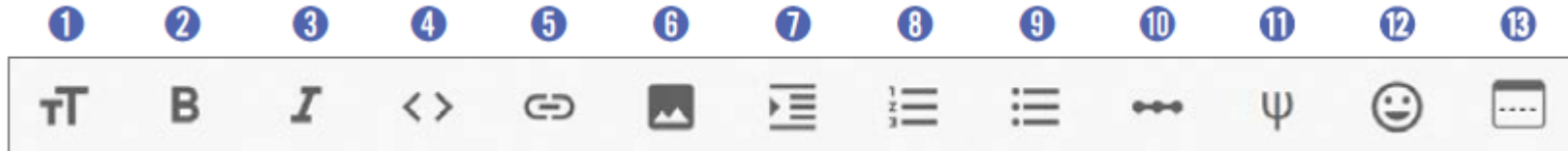
colab 구글 코랩과 주피터 노트북

텍스트 셀

- 셀(cell) - 코랩에서 실행할 수 있는 최소 단위
- 텍스트 셀을 수정하려면 원하는 셀로 이동한 후 Enter 키를 누르거나 마우스로 더블 클릭
- 텍스트 셀에는 HTML(Hypertext Markup Language)과 마크다운(Markdown)을 혼용해서 사용 가능



- 코랩에서 제공하는 텍스트 셀 툴바



- ① 제목 전환
- ② 굵게
- ③ 기울임꼴
- ④ 코드로 형식 지정
- ⑤ 링크 삽입
- ⑥ 이미지 삽입
- ⑦ 들여쓰기
- ⑧ 번호 매기기 목록 추가
- ⑨ 글머리기호 목록 추가
- ⑩ 가로줄 추가
- ⑪ LaTeX
- ⑫ 이모티콘 삽입
- ⑬ 마크다운 미리보기 위치 변경

colab 구글 코랩과 주피터 노트북

- 현재 선택한 셀의 오른쪽 위의 현재 셀에 적용할 수 있는 기능



- ① 위(아래)로 셀 이동
- ② 셀에 연결
- ③ 편집기 설정 열기
- ④ 수정
- ⑤ 탭에서 셀 미러링
- ⑥ 셀 삭제
- ⑦ 셀 작업 더 보기

마크다운 서식

마크다운 서식	설명	예시
# 제목1	<h1> 태그와 동일합니다.	제목1
## 제목2	<h2> 태그와 동일합니다.	제목2
### 제목3	<h3> 태그와 동일합니다.	제목3
#### 제목4	<h4> 태그와 동일합니다.	제목4
##### 제목5	<h5> 태그와 동일합니다.	제목5
혼공분석	굵게 씁니다.	혼공분석
<i>*혼공분석*</i> 또는 <u>_혼공분석_</u>	기울임꼴로 씁니다.	<i>혼공분석</i>
~~혼공분석~~	취소선을 추가합니다.	혼공분석
<code>`print("Hello World!")`</code>	코드 서체로 씁니다.	<code>print("Hello World!")</code>

마크다운 서식

- 실습
다양한 마크다운 활용

텍스트 셀 툴바

	제목 전환		굵게
	기울임꼴		코드로 형식 지정
	링크 삽입		이미지 삽입
	들여쓰기		번호 매기기 목록 추가
	글머리기호 목록 추가		가로줄 추가
	LaTeX		이모티콘 삽입
 	마크다운 미리보기 위치 변경		

colab 구글 코랩과 주피터 노트북

- 새로운 코드 셀을 추가

The screenshot shows the Google Colaboratory interface. At the top, there's a header with the Colab logo and text in Korean: "Colaboratory에 오신 것을 환영합니다" (Welcome to Colaboratory). Below this, there are navigation links: "파일" (File), "수정" (Edit), "보기" (View), "삽입" (Insert), "런타임" (Runtime), "도구" (Tools), "도움말" (Help), and "변경사항을 저장할 수 있음" (Can save changes). On the right, there are icons for "공유" (Share) and "설정" (Settings). Below the header, there's a toolbar with buttons for "+ 코드" (Add Code), "+ 텍스트" (Add Text), and "Drive로 복사" (Copy to Drive). To the right of these buttons are indicators for "RAM" and "디스크" (Disk) usage, and a "수정 가능" (Editable) status. The main area is titled "시작하기" (Getting Started) and contains introductory text in Korean. Below the text, there's a code cell with the following Python code:

```
seconds_in_a_week = 24 * 60 * 60 * 7
seconds_in_a_week
```

The output of the cell is "604800". Below the code cell, there are buttons for "+ 코드" (Add Code) and "+ 텍스트" (Add Text). A blue arrow points to the "+ 코드" button with the text "현재 셀 아래에 마우스 커서를 올리면 버튼이 나타납니다." (When you move the mouse cursor below the current cell, the button appears.). Below the buttons, there's more text in Korean explaining how to run the code. At the bottom, there's a status bar showing "0초" (0 seconds) and "오전 8:30에 완료됨" (Completed at 8:30 AM).

Colaboratory에 오신 것을 환영합니다

파일 수정 보기 삽입 런타임 도구 도움말 변경사항을 저장할 수 있음

+ 코드 + 텍스트 Drive로 복사

RAM 디스크 수정 가능

시작하기

(x) 지금 읽고 계신 문서는 정적 웹페이지가 아니라 코드를 작성하고 실행할 수 있는 대화형 환경인 Colab 메모장입니다.

예를 들어 다음은 값을 계산하여 변수로 저장하고 결과를 출력하는 간단한 Python 스크립트가 포함된 코드 셀입니다.

```
seconds_in_a_week = 24 * 60 * 60 * 7
seconds_in_a_week
```

604800

+ 코드 + 텍스트

현재 셀 아래에 마우스 커서를 올리면 버튼이 나타납니다.

위 셀의 코드를 실행하려면 셀을 클릭하여 선택한 후 코드 왼쪽의 실행 버튼을 누르거나 단축키 'Command/Ctrl+Enter'를 사용하세요. 셀을 클릭하면 코드 수정을 바로 시작할 수 있습니다.

특정 셀에서 정의한 변수를 나중에 다른 셀에서 사용할 수 있습니다.

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

0초 오전 8:30에 완료됨

colab 구글 코랩과 주피터 노트북

노트북

- 코랩 노트북, 구글 클라우드의 가상 서버 사용
- 구글 클라우드의 컴퓨트 엔진Compute Engine에 연결
- 서버의 메모리 약 12GB, 디스크 공간은 100GB
- 코랩 노트북을 사용해 무료로 가상 서버 활용 가능



- 실습

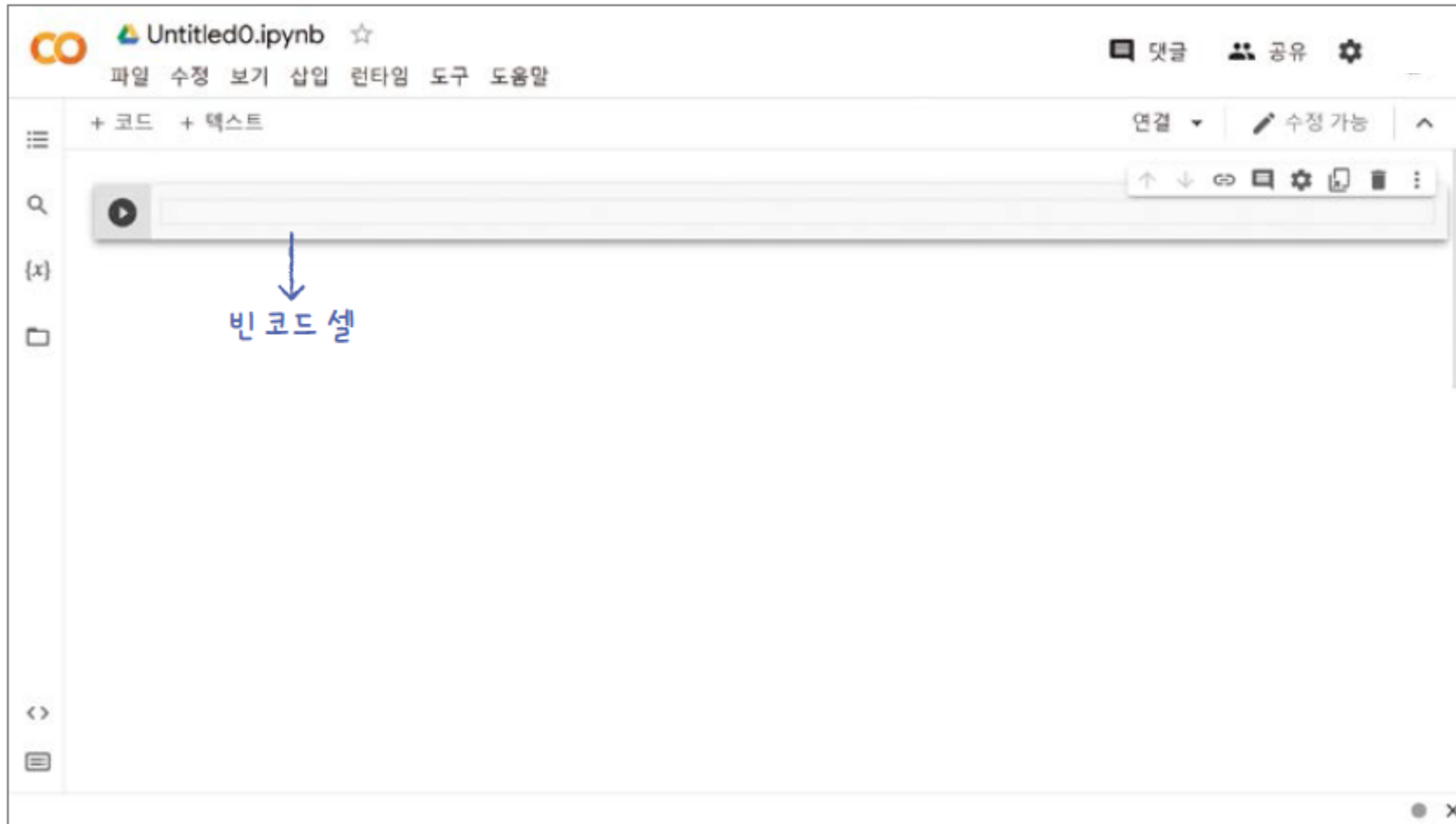
새 노트북 만들기

01 노트북 화면에서 [파일] - [새 노트] 메뉴를 선택하여 새로운 노트북 만들기



- 실습

02 웹 브라우저의 새로운 탭에 Untitled0.ipynb란 이름의 새로운 노트북 생성



colab 구글 코랩과 주피터 노트북

- 실습

03 코드 셀에 'Hello World'를 출력



- 실습

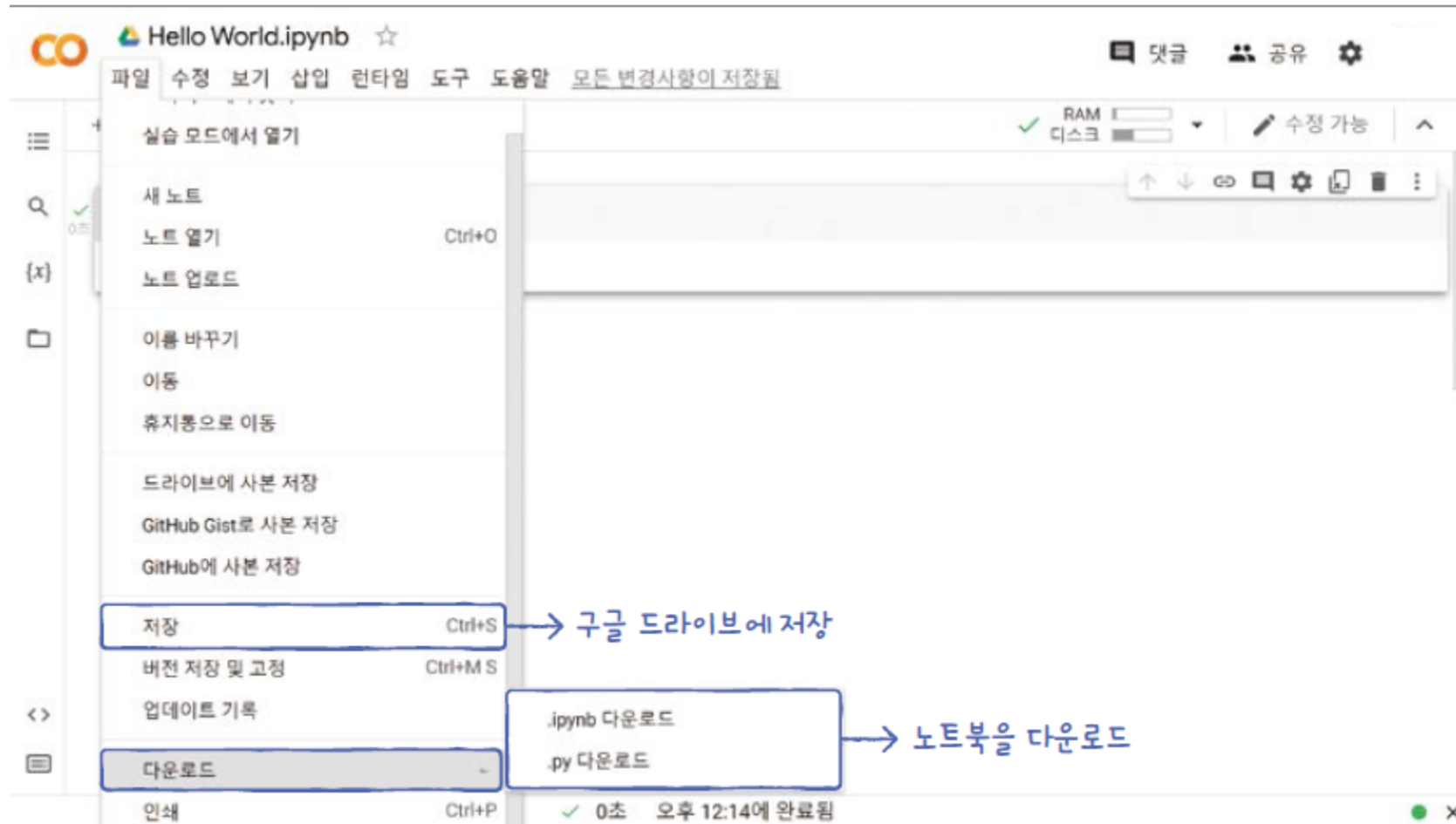
04 노트북 이름 변경 – 제목을 마우스로 클릭한 후 [Hello World]를 입력



colab 구글 코랩과 주피터 노트북

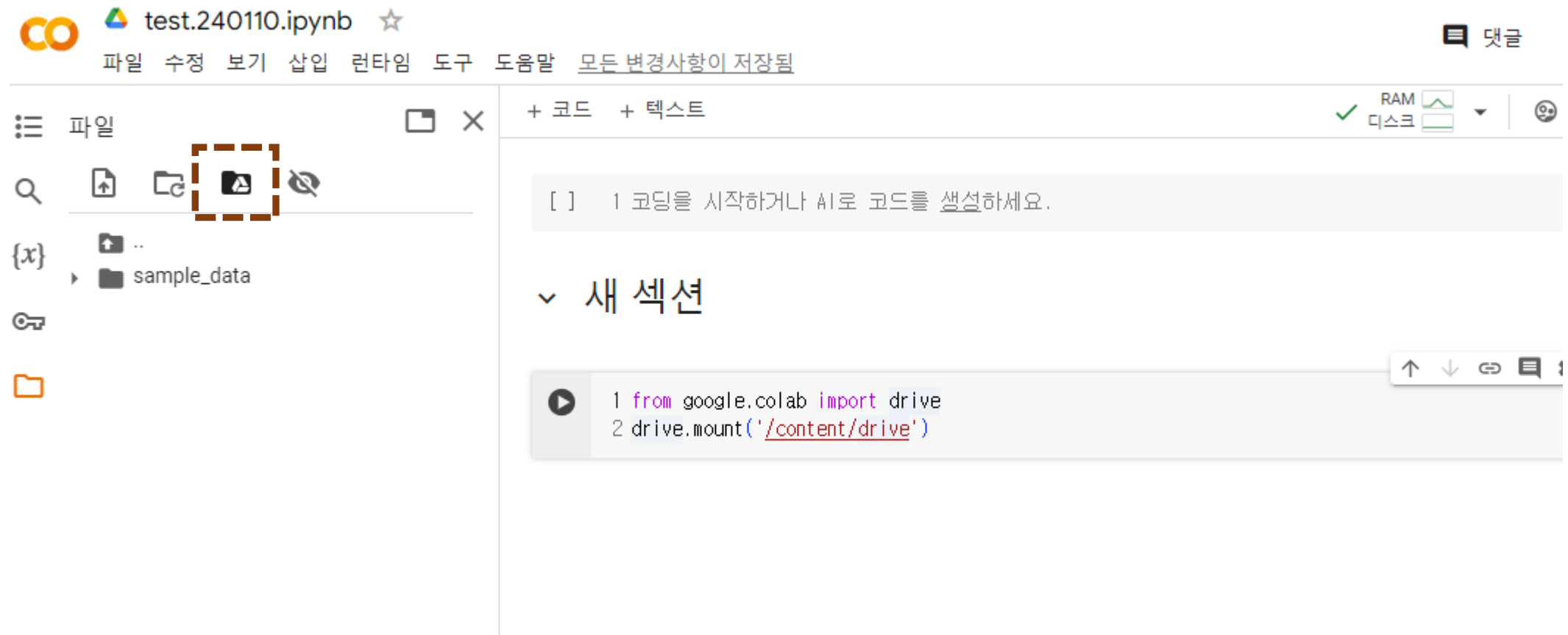
• 실습

05 컴퓨터에 다운로드 or 구글 드라이브에 저장



• 실습

05 컴퓨터에 다운로드 or 구글 드라이브에 저장



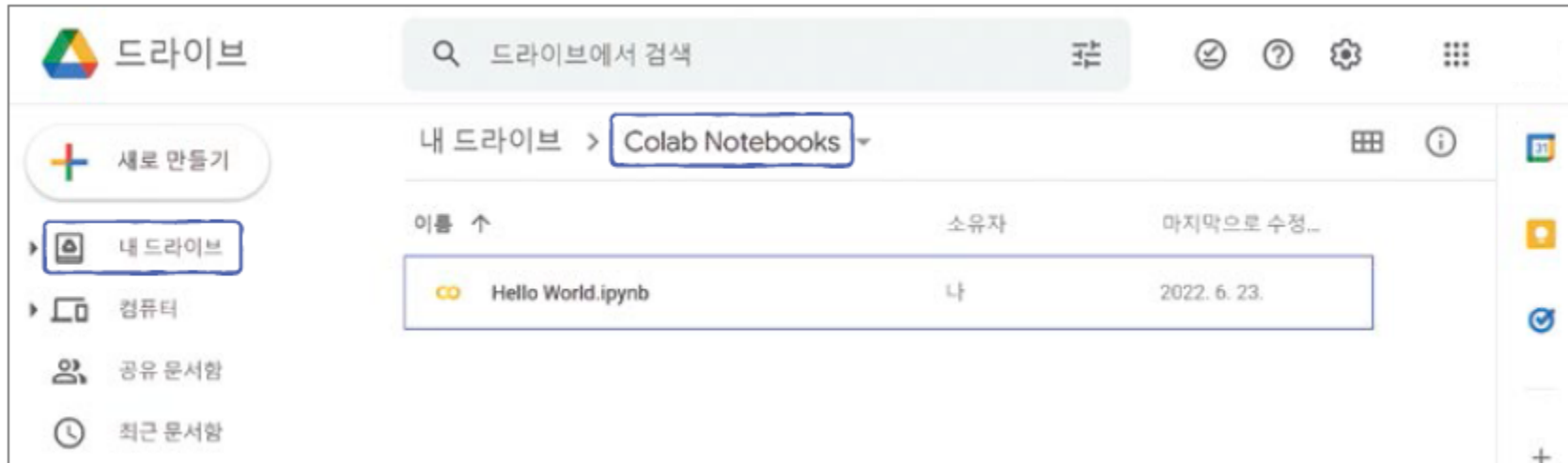
The screenshot displays the Google Colab interface for a notebook titled 'test.240110.ipynb'. The top navigation bar includes icons for file management, editing, viewing, saving, running, tools, and help, along with a status message '모든 변경사항이 저장됨' (All changes saved). On the left, a file explorer sidebar shows a folder named 'sample_data'. In the center toolbar, the 'Save to Drive' icon (a triangle with a circle) is highlighted with a dashed orange box. The main workspace on the right contains a code cell with the following text: '[] 1 코딩을 시작하거나 AI로 코드를 생성하세요.' Below this is a section header '새 섹션' (New Section). At the bottom, there is a code cell with a play button icon and the following Python code:

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

colab 구글 코랩과 주피터 노트북

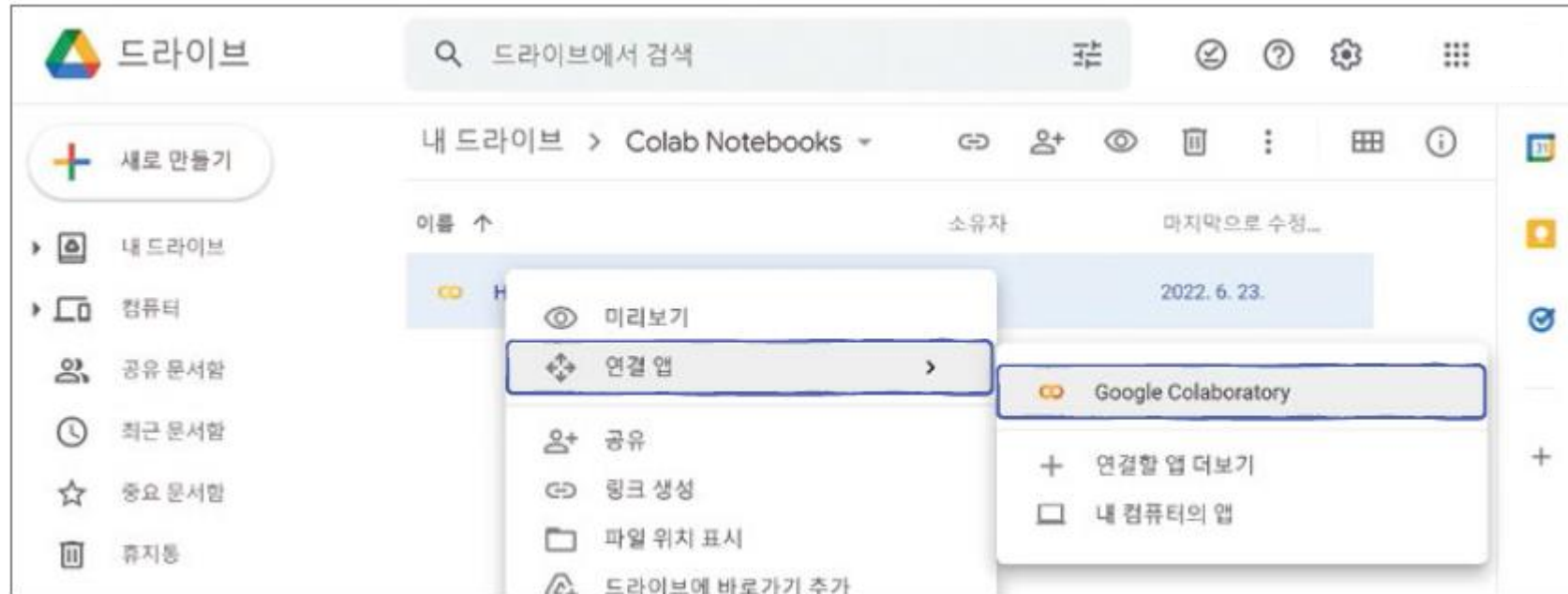
- 실습

06 [내 드라이브] - [Colab Notebooks] 폴더에서 확인



• 실습

07 저장된 노트북 코랩으로 불러오기



정리

- **코랩** : 웹 브라우저 기반의 파이썬 코드 실행 환경
- **노트북** : 코랩의 프로그램 작성 파일.
대화식으로 프로그램 작성, 데이터 분석 / 교육에 적합.
노트북에는 코드, 코드 실행 결과, 문서를 모두 저장하여 보관 가능
- **구글 드라이브** : 구글이 제공하는 클라우드 파일 저장 서비스.
코랩에서 만든 노트북을 구글 드라이브의 'Colab Notebooks' 폴더에 저장, 필요한 경우 다시 호출
- **마크다운** : HTML과 같이 일반 텍스트, 서식 있는 문서를 작성하는 데 사용

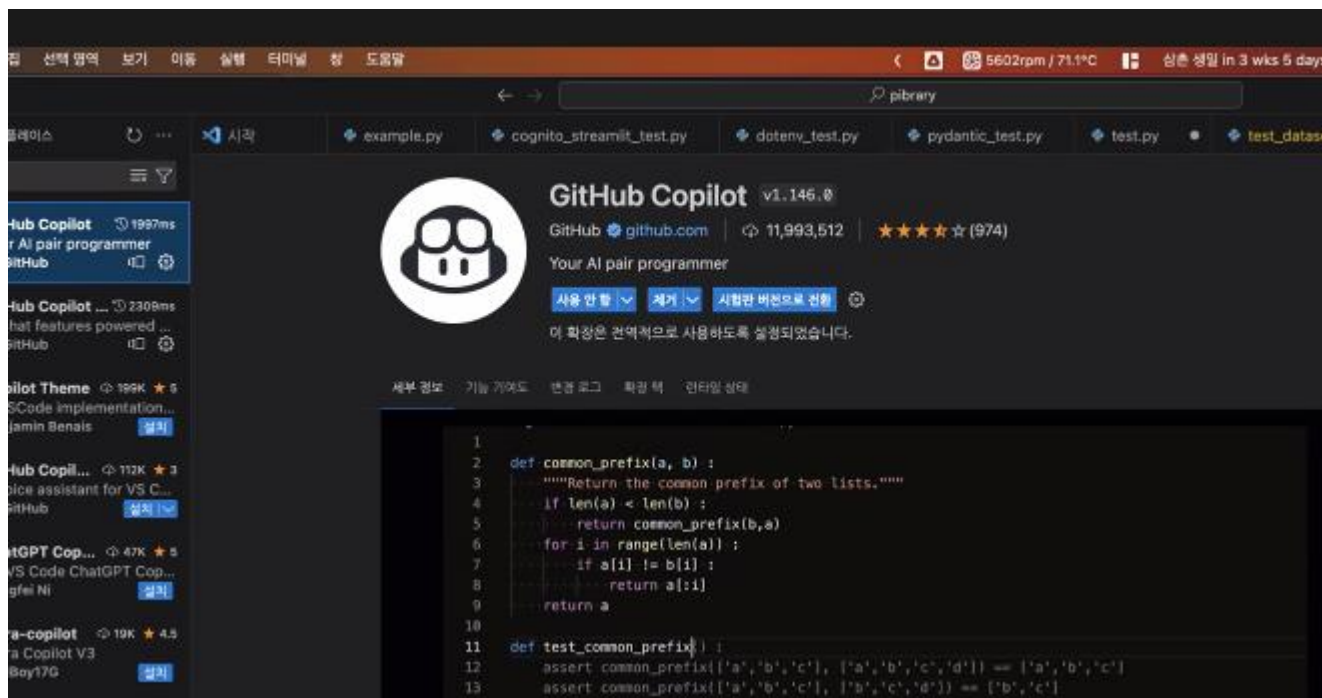
비주얼 스튜디오 코드 (Visual Studio Code)

- 마이크로소프트가 마이크로소프트 윈도우, macOS, 리눅스용으로 개발한 소스 코드 편집기. 디버깅 지원과 Git 제어, 구문 강조 기능, SSH 접속 등이 포함되어 있으며, 사용자가 편집기의 테마와 단축키, 설정 등을 수정할 수 있다. - wiki



파이참 (PyCharm)

- 파이썬 개발에 널리 사용되는 통합 개발 환경(IDE, Integrated Development Environment) 또는 개발 도구
코드 편집기, 디버거, 컴파일러, 인터프리터 등을 포함, 자동 완성, 검색 등의 다양한 기능 제공



라이브러리

데이터 분석을 위한 도구 - 파이썬

- 실습

1.01.Python.Library.ipynb

데이터 분석을 위한 도구 - 파이썬

1. 라이브러리 (Library)

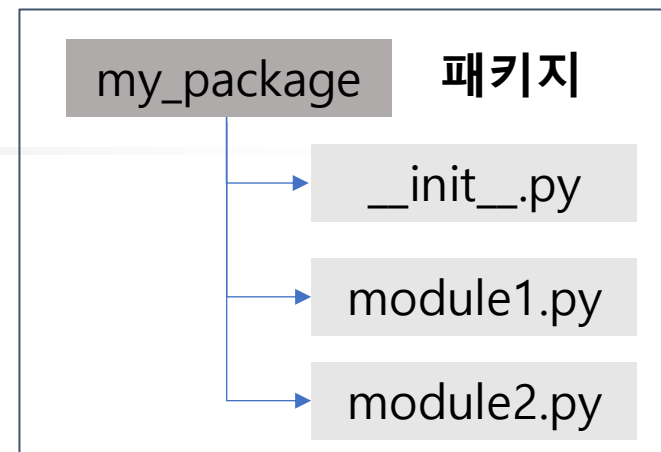
- 기능적인 코드 집합. 여러 모듈 또는 패키지들로 구성
- Python에서는 일반적으로 하나의 모듈이나 패키지가 라이브러리 역할 수행
- ex, NumPy : 수학적 기능을 제공하는 라이브러리. 하나의 모듈(numpy)로 구성

2. 패키지 (Package)

- 디렉터리(폴더) 구조를 통해 여러 모듈을 논리적으로 그룹화(패키징)
- 패키지는 내부에 `__init__.py` 파일을 포함, 이 파일을 통해 디렉터리가 패키지로 인식.
- ex, pandas : 여러 모듈로 구성된 패키지. pandas 내부에는 다양한 서브 패키지와 모듈이 포함

3. 모듈 (Module)

- .py 확장자를 가진 Python 파일, Python 코드가 들어 있는 단일 파일을 의미
- 함수, 클래스, 변수 등을 포함
- ex, math.py라는 파일이 있을 경우 -> math



데이터 분석을 위한 도구

- 실습

1.01.Python.Module.welcome.ipynb



The screenshot shows a Jupyter Notebook titled "1.01.Python.Module.welcome.ipynb". The interface includes a top bar with the Colab logo, file management icons, and a status bar. The left sidebar shows a file explorer with a folder named "sample_data" and a file named "welcome.py". The main area displays two code cells. The first cell defines a function `def welcome_message(course):` that prints a message. The second cell imports the `welcome` module and calls `welcome.welcome_message("Python Basics Part 1")`. The output of the second cell is displayed below the code.

```
[1] 1 def welcome_message(course):  
    2 print("Thank you for subscribing to our " + course + " course. You will get all the details in an email shortly.")
```

```
1 import welcome  
2 welcome.welcome_message("Python Basics Part 1")  
3 # Output  
4 # Thank you for subscribing to our Python Basics Part 1 course. You will get all the details in the email shortly.
```

Thank you for subscribing to our Python Basics Part 1 course. You will get all the details in an email shortly.

- TASK

다양한 모듈 설계 해보기

자료구조와 알고리즘

THANK YOU