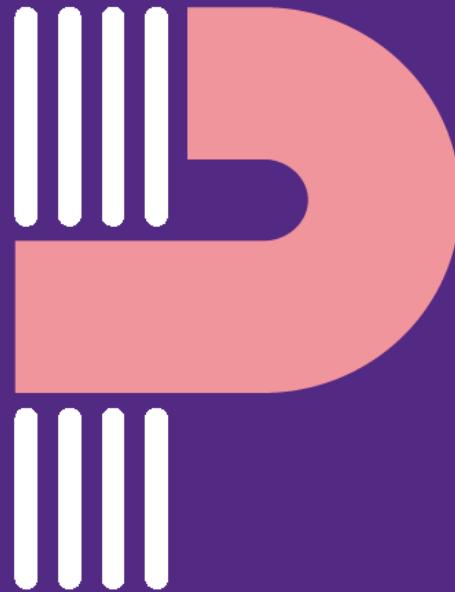


실생활 융합 예제로 배우는

파이썬 마스터



PART 01. 파이썬 기초

01. 프로그래밍과 파이썬

목차

1.1 프로그래밍 언어

1.2 파이썬의 개념

1.3 파이썬 개발 환경

1.4 처음 만나는 프로그램

1.5 [플러스 예제] 문자열 출력

1.6 [플러스 예제] 연산 결과 출력

학습목표

- 프로그래밍의 개념을 설명할 수 있습니다.
- 파이썬의 특징을 설명할 수 있습니다.
- 코랩에서 파이썬 프로그램을 작성하고 실행할 수 있습니다.
- 문자열을 출력하는 프로그램을 작성할 수 있습니다.
- 간단한 연산을 하는 프로그램을 작성할 수 있습니다.

Section 1.1

프로그래밍 언어

1. 프로그래밍 언어

■ 프로그래밍의 이해

- 컴퓨터를 이용하여 문제를 해결하려면 컴퓨터에 명령을 내려야 함
- 프로그램(program): 컴퓨터로 문제를 해결하기 위해 작성하는 명령어의 모음
- 프로그래밍(programming): 프로그램을 작성하는 과정
- 프로그래머(programmer): 프로그램을 작성하는 사람 또는 직업



그림 1-1 프로그램, 프로그래밍, 프로그래머의 개념

1. 프로그래밍 언어

■ 프로그래밍의 언어의 종류

- 프로그래밍 언어(programming language): 프로그램을 작성할 때 사용하는 언어

▪ 저급 언어와 고급 언어

- 저급 언어: 기계중심 언어, 즉 하드웨어 지향적인 언어
 - Ex) 어셈블리어
- 고급 언어: 사람이 이해하기 쉬운 일상적인 언어와 기호를 사용하는 언어
 - Ex) C, C++, 자바, 파이썬 등

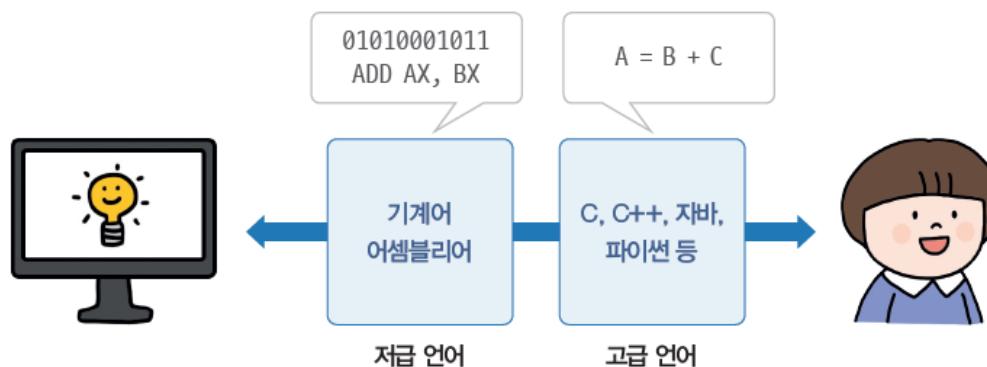


그림 1-2 저급 언어와 고급 언어

1. 프로그래밍 언어

■ 프로그래밍의 언어의 종류

▪ 언어 번역 프로그램

- 고급 언어로 작성된 프로그램은 컴퓨터 내부에서 실행되기 전에 기계어로 번역하는 과정을 거쳐야 함
- 번역을 담당하는 소프트웨어를 언어 번역 프로그램이라고 함
 - 컴파일러(compiler): 프로그램 전체를 한꺼번에 기계어로 번역한 후 실행
 - 인터프리터(interpreter): 프로그램을 한 행씩 읽어 번역한 후 실행

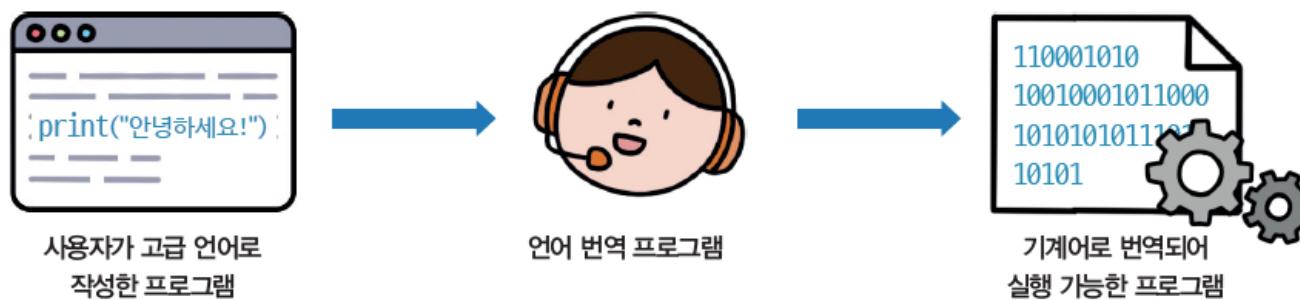


그림 1-3 언어 번역 프로그램의 역할

Section 1.2

파이썬의 개념

2. 파이썬의 개념

■ 파이썬의 탄생

- 파이썬(Python): 네덜란드계 소프트웨어 엔지니어인 귀도 반 로섬 Guido van Rossum이 1991년에 개발한 프로그래밍 언어



그림 1-4 귀도 반 로섬 © gvanrossum.github.io

2. 파이썬의 개념

■ 파이썬의 인기

- TIOBE 인덱스(<https://www.tiobe.com/tiobe-index/>): 프로그래밍 언어의 인기도를 측정하는 지수
- 2024년 1월 기준으로 파이썬이 1위, C가 2위, C++가 3위를 차지함

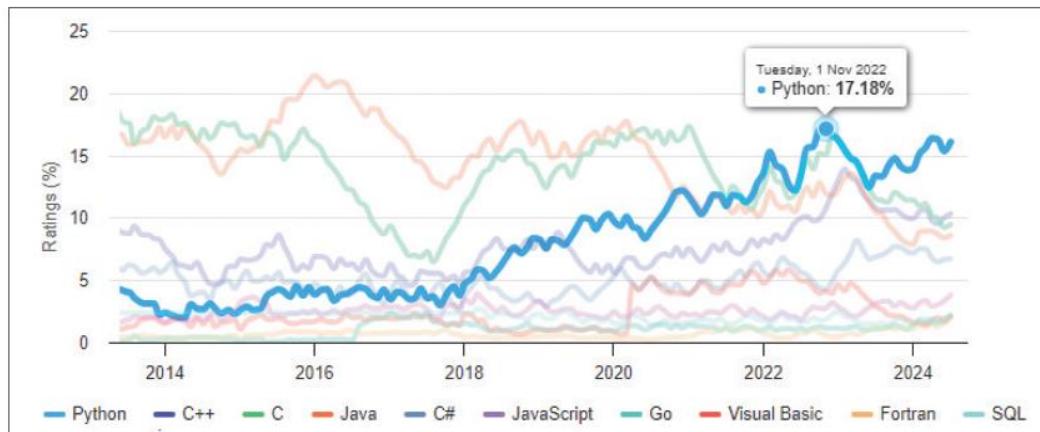


그림 1-5 프로그래밍 언어의 순위 © TIOBE

2. 파이썬의 개념

■ 파이썬의 특징

- 파이썬이 인기 있는 이유
 - 다른 프로그래밍 언어에 비해 문법 구조가 간결하고 읽기 쉬움

표 1-1 대표적인 프로그래밍 언어의 코드 비교

언어	코드
C 언어	#include <stdio.h> int main() { printf("안녕하세요!"); return 0; }
자바	public class Hello { public static void main(String[] args) { System.out.println("안녕하세요!"); } }
파이썬	print('안녕하세요!')

2. 파이썬의 개념

■ 파이썬의 특징

- 파이썬이 인기 있는 이유
 - 다양한 라이브러리를 활용할 수 있음
 - 라이브러리(library): 프로그램에서 자주 사용되는 부분 프로그램들을 모아 놓은 것

표 1-2 데이터 과학과 인공지능 관련 라이브러리

라이브러리	설명
matplotlib 	파이썬의 강력하고 인기 있는 데이터 시각화 라이브러리로, 데이터로부터 차트를 생성하거나 이미지를 출력할 때 사용한다. 과학자, 공학자, 데이터 과학자 등 다양한 분야의 전문가들이 활용하고 있다.

2. 파이썬의 개념

넘파이(NumPy) 	'Numerical Python'의 줄임말인 넘파이(NumPy)는 고성능 수치 계산을 위해 제작되었으며, 특히 벡터 및 행렬 연산에 편리하게 사용할 수 있다. 빠른 속도가 중요한 데이터 과학과 인공지능 분야에서 많이 사용된다.
판다스(Pandas) 	데이터를 처리하거나 데이터를 분석할 때 사용하는 라이브러리로, 대용량의 데이터들을 관리하고 처리하는 데 매우 편리하다. 숫자 테이블과 시계열 처리를 위한 데이터 구조와 연산을 제공한다.
사이킷런(Scikit-learn) 	파이썬에서 사용하는 가장 대표적인 머신러닝용 라이브러리로, 분류·회귀·군집화 등 다양한 머신러닝 알고리즘을 제공한다.
케라스(Keras) 	대표적인 딥러닝 전용 라이브러리로, 사용자가 손쉽게 딥러닝 모델을 개발하고 활용할 수 있도록 인터페이스를 제공한다.
텐서플로(TensorFlow) 	구글(Google)이 2015년에 공개한 딥러닝과 머신러닝용 라이브러리로, 현재는 가장 인기 있는 딥러닝 라이브러리가 되었다. 텐서플로2에서는 케라스를 라이브러리로 지원하고 있어서 텐서플로 사용자가 손쉽게 케라스를 활용할 수 있다.

Section 1.3

파이썬 개발 환경

3. 파이썬 개발 환경

■ 구글 코랩

- 코랩(Colab): 구글에서 제공하는 개발 환경, 브라우저에서 파이썬 프로그램을 작성하고 실행시켜 줌
- 코랩의 장점은 다양한 라이브러리가 기본으로 설치되어 있으며 사용자가 작성한 프로그램이 구글 드라이브에 저장된다는 것

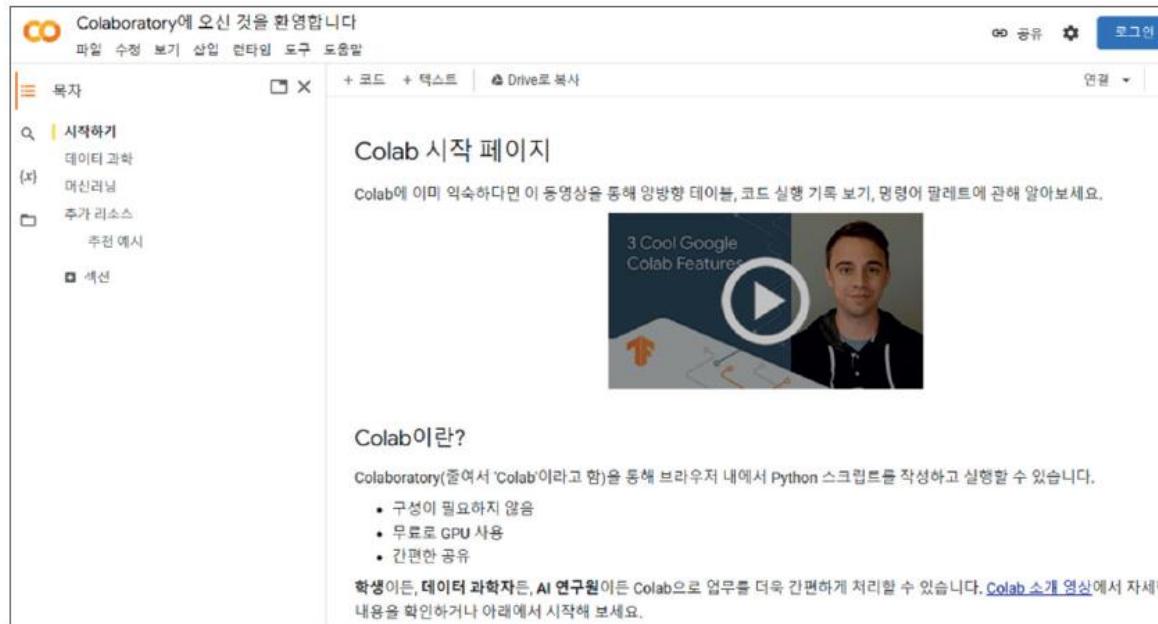


그림 1-6 구글 코랩 시작 페이지

3. 파이썬 개발 환경

■ 구글 코랩 접속하기

- 구글 코랩에 접속하고 화면이 어떻게 구성되어 있는지 살펴보기

01 구글 계정에 로그인. 웹 브라우저는 가능하면 크롬을 이용

02 구글 코랩(<https://colab.research.google.com>)에 접속하고 팝업 창 왼쪽 아래의 [새 노트] 버튼을 클릭

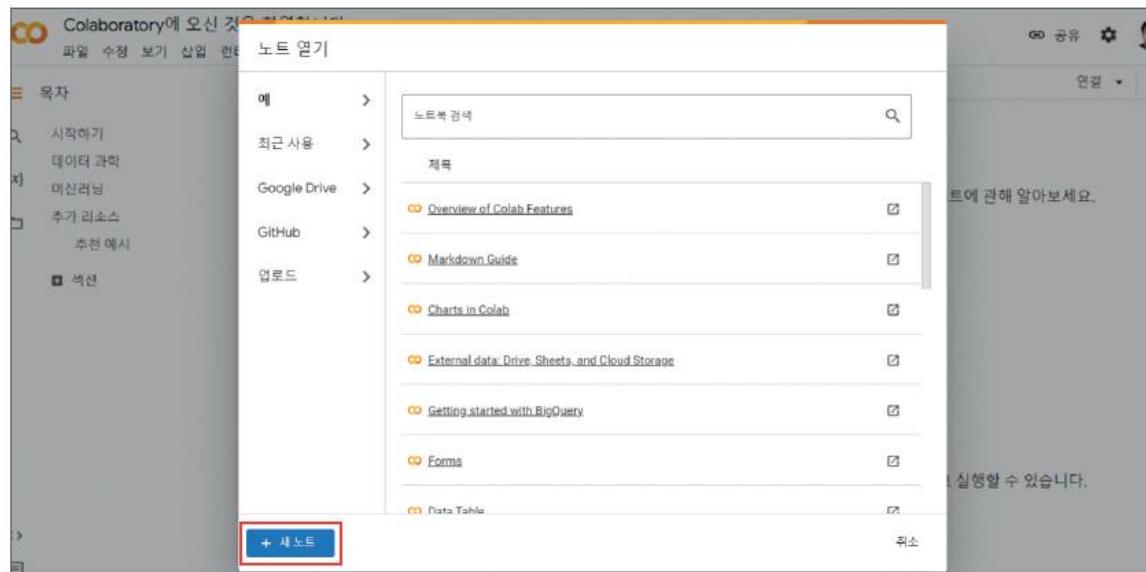


그림 1-7 코랩 새 노트 생성

3. 파이썬 개발 환경

■ 구글 코랩 접속하기

- 구글 코랩에 접속하고 화면이 어떻게 구성되어 있는지 살펴보기

03 Untitled0.ipynb'라는 제목으로 새로운 화면이 열림, 코드 셀에 코드를 입력하고 왼쪽의 [셀 실행] 버튼()을 클릭하거나 단축키 [Ctrl] + [Enter] 를 누르면 입력한 코드가 실행됨

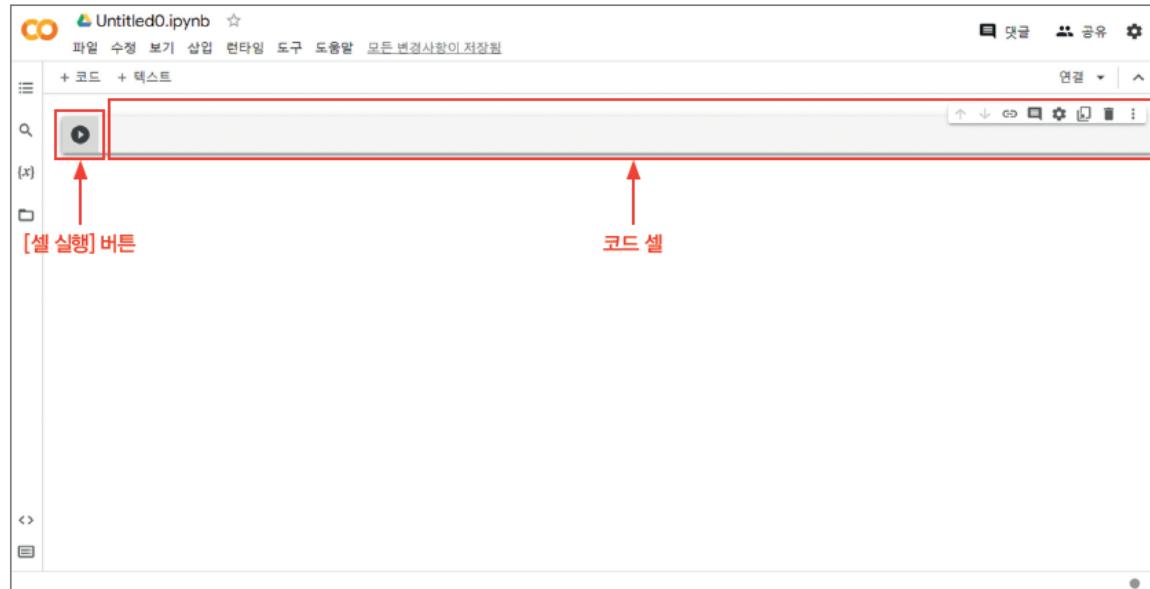


그림 1-8 코랩 메모장의 코드 셀과 [셀 실행] 버튼

Section 1.4

처음 만나는 프로그램

4. 처음 만나는 프로그램

■ 프로그램 작성하고 실행하기

- 코랩에서 짧은 프로그램 하나를 작성하고 실행해 보기

01 코드 셀에 다음과 같은 코드를 입력



The screenshot shows a Jupyter Notebook interface. On the left, there's a sidebar with icons for file operations and a search bar. The main area is a code cell containing the following Python code:

```
print('반갑습니다. 파이썬입니다.')
```

The code cell has a play button icon and a green checkmark indicating it's ready to run.

그림 1-9 코드 셀에 코드 입력

4. 처음 만나는 프로그램

■ 프로그램 작성하고 실행하기

- 코랩에서 짧은 프로그램 하나를 작성하고 실행해 보기

02 코드 셀 왼쪽에 있는 [셀 실행] 버튼()을 클릭하거나 [Ctrl] + [Enter]로 코드를 실행

The screenshot shows a Jupyter Notebook interface with a single code cell containing the Python command `print('반갑습니다. 파이썬입니다.')`. The cell has been executed, and its output, `반갑습니다. 파이썬입니다.`, is displayed below the code. A red box highlights the play button icon in the toolbar above the cell, which is labeled '[셀 실행] 버튼' (Cell Execute Button) with a red arrow pointing to it. The output text is also highlighted with a red box and labeled '실행 결과' (Execution Result) with a red arrow pointing to it.

그림 1-10 코드 셀 실행 결과

4. 처음 만나는 프로그램

■ 프로그램 작성하고 실행하기

- 코랩에서 짧은 프로그램 하나를 작성하고 실행해 보기

03 새로운 코드 셀을 추가하고 싶으면 코드 셀 위쪽에 있는 [+ 코드] 버튼을 클릭. 또는 코드 셀 위쪽 라인과 아래쪽 라인에 마우스 커서를 위치시키면 자동으로 [+ 코드] 버튼이 나타남

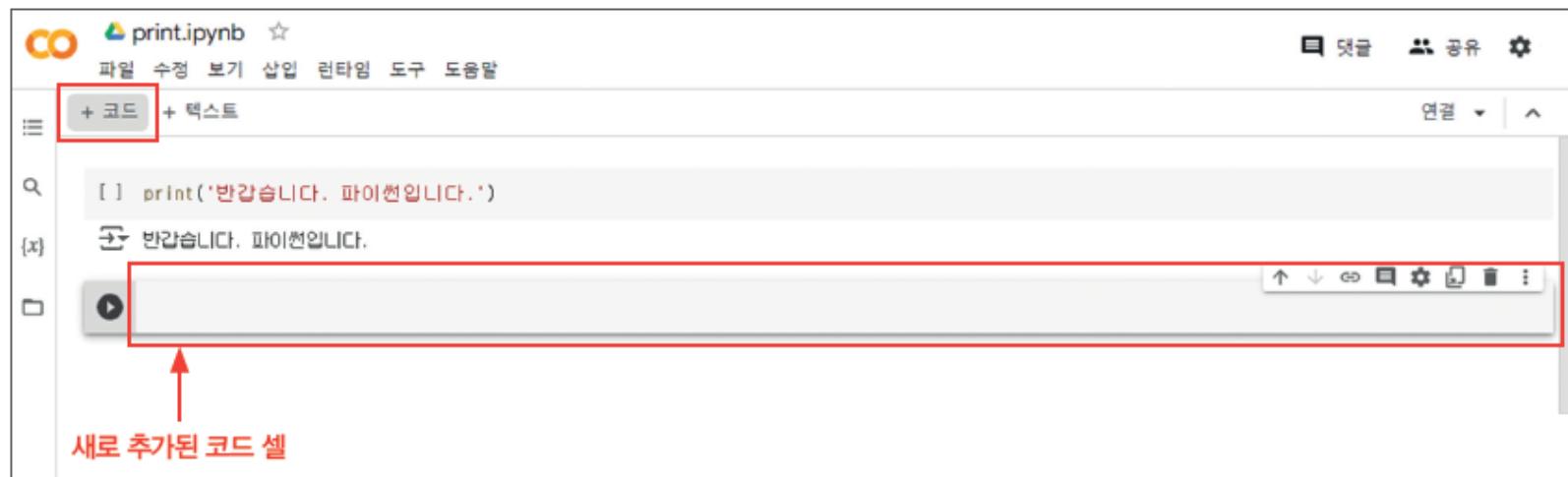


그림 1-11 코드 셀 추가

4. 처음 만나는 프로그램

■ 노트 파일 다루기

- 앞에서 작성한 프로그램은 구글 드라이브에 'Untitled0.ipynb'라는 이름으로 자동 저장됨
- Untitled0.ipynb 이름을 마우스로 클릭한 후 'Untitled0'을 'print'로 변경

The screenshot shows the Google Colab interface. At the top, there's a navigation bar with the CO logo, the file name 'print.ipynb' (which is highlighted with a red box), and a star icon. Below the navigation bar is a menu bar with Korean options: 파일 (File), 수정 (Edit), 보기 (View), 삽입 (Insert), 런타임 (Runtime), 도구 (Tools), and 도움말 (Help). On the left side, there's a sidebar with icons for file operations like + 코드 (New Code) and + 텍스트 (New Text). The main workspace contains a code cell with the following content:

```
print('반갑습니다. 파이썬입니다.')
```

Below the code cell, the output is displayed in a box:

반갑습니다. 파이썬입니다.

그림 1-12 파일 이름 변경

4. 처음 만나는 프로그램

■ 파일 불러오기

- 코랩 노트는 구글 드라이브의 [내 드라이브]-[Colab Notebooks] 폴더에 저장됨
- 구글 드라이브에서 불러오기

01 구글 드라이브(<https://drive.google.com>)의 [내 드라이브]-[Colab Notebooks] 폴더로 이동

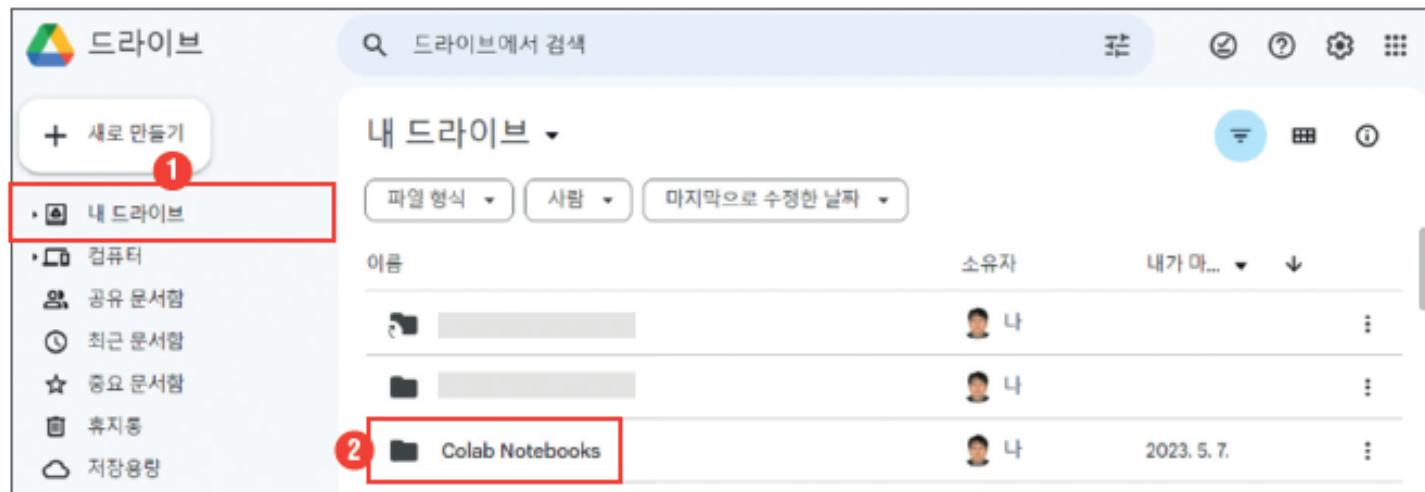


그림 1-13 구글 드라이브의 [Colab Notebooks] 폴더

4. 처음 만나는 프로그램

■ 파일 불러오기

- 코랩 노트는 구글 드라이브의 [내 드라이브]-[Colab Notebooks] 폴더에 저장됨
- 구글 드라이브에서 불러오기

02 print.ipynb를 더블클릭하면 구글 코랩에서 코랩 노트 print.ipynb가 열림

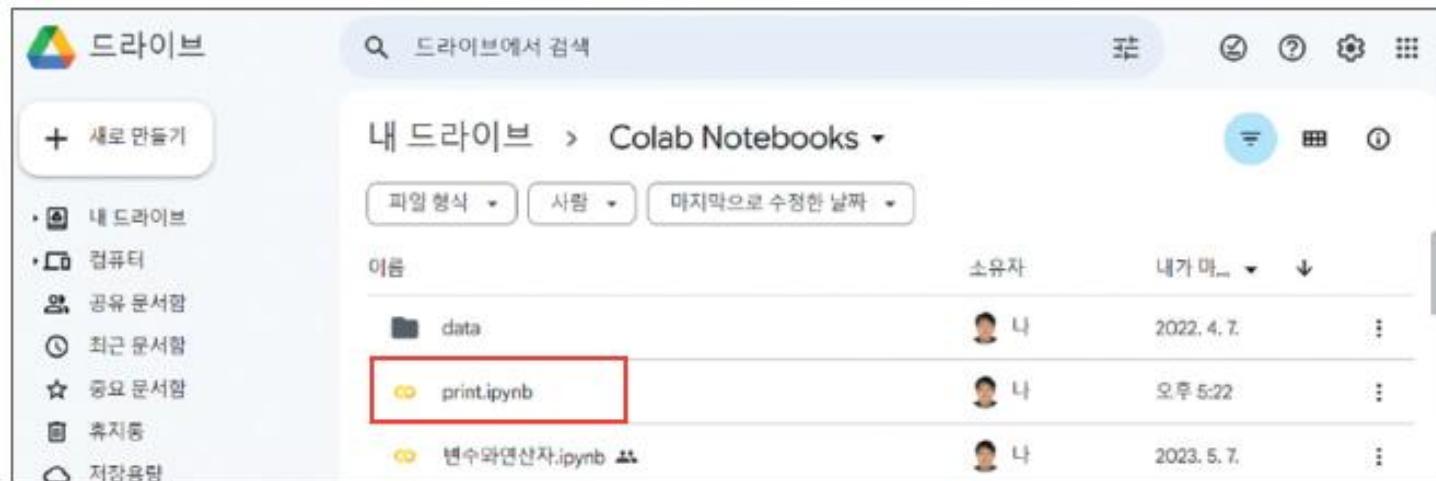


그림 1-14 코랩 노트 열기

4. 처음 만나는 프로그램

■ 파일 불러오기

▪ 코랩에서 불러오기

01 구글 코랩(<https://colab.research.google.com>)에 접속하면 팝업 창에 최근에 사용한 파일 목록이 나타남

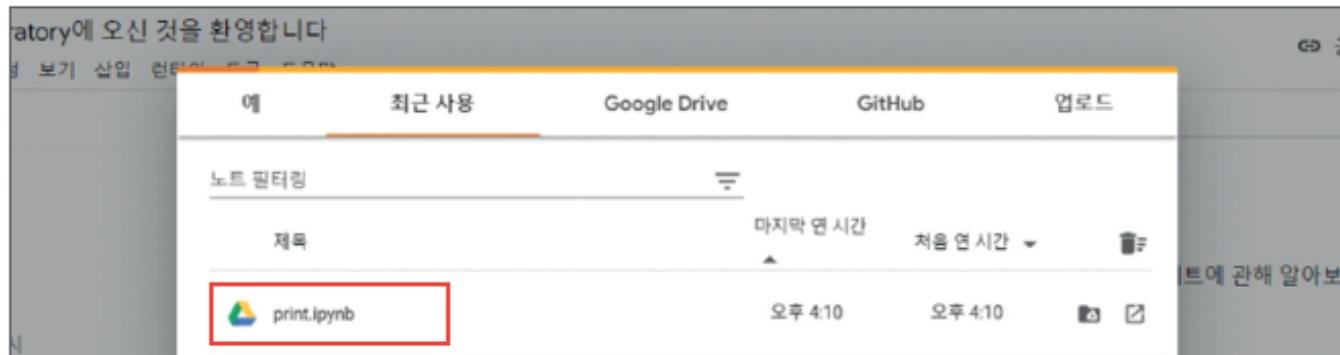


그림 1-15 최근에 사용한 파일 목록

02 print.ipynb를 클릭하면 print.ipynb 코랩 노트가 열림

4. 처음 만나는 프로그램

■ 프로그램 이해하기

```
print('반갑습니다. 파이썬입니다.')
```

- 이 코드는 print 함수를 이용해서 '반갑습니다. 파이썬입니다.'라는 문자열을 화면에 출력
- 문자열: 작은 따옴표('') 또는 큰 따옴표("")로 둘러싼 문자들
- 만약 여러 개의 값을 출력하려면 값들을 콤마(,)로 구분하여 나열

```
print('반갑습니다.', '파이썬입니다.')
```

```
반갑습니다. 파이썬입니다.
```

4. 처음 만나는 프로그램

■ 프로그램 이해하기

- 다음과 같이 두 개의 print 함수를 이용하면 '반갑습니다.'를 출력한 후에 줄바꿈을 하고 '파이썬입니다.'를 출력

```
print('반갑습니다.')
print('파이썬입니다.)
```

반갑습니다.

파이썬입니다.

4. 처음 만나는 프로그램

■ 프로그램 이해하기

- 프로그램은 위에서부터 아래로 한 문장씩 실행
- 두 개의 print 함수를 이용했을 때 '반갑습니다.'가 먼저 출력된 다음 '파이썬입니다.'가 출력됨

실행 순서



```
print('반갑습니다.')
print('파이썬입니다.')
```

4. 처음 만나는 프로그램

■ 프로그램 이해하기

- print 함수 안에 문자열이 아닌 수식이 있으면 수식의 연산 결과가 출력됨

```
print(3*7)
```

```
21
```

```
print(3*7)
```



Section 1.5

[플러스 예제] 문자열 출력

5. [플러스 예제] 문자열 출력

- 다음은 print 함수를 이용해서 문자열을 출력하는 프로그램

```
01 print('홍길동님, 안녕하세요!')
```

```
홍길동님, 안녕하세요!
```

동작 과정

- print 함수의 소괄호(()) 안에 위치한 값을 출력합니다.

```
print('홍길동님, 안녕하세요!')  
      ↓  
      값
```

Section 1.6

연산 결과 출력

6. [플러스 예제] 연산 결과 출력

- print 함수를 이용해서 수식의 연산 결과를 출력하는 프로그램

```
01 print(5*100)
```

```
500
```

동작 과정

- 1 print 함수의 소괄호(()) 안에 위치한 내용이 수식이면 연산 결과를 출력합니다.

```
print(5*100) ← 연산 결과
```