

Python 기초 1일차

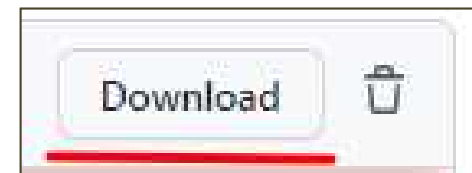
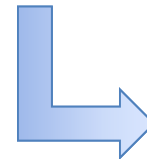
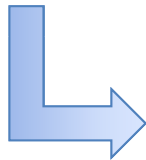
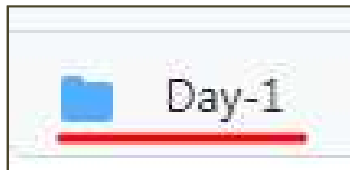
Python과 Google Colab 기본 사용법

홍정하

서강대 인문콘텐츠 융합소프트웨어 연계전공

1일차 강의자료 다운로드

<https://github.com/kleist2000/lecture>



데이터분석/기계학습과 프로그래밍 언어

- ▶ 편리한 도구
 - 엑셀, SPSS 등 소프트웨어 학습/사용 편리
 - 구현 기능만 사용 가능, 새로운 기법/모델의 업데이트 제약
- ▶ 프로그래밍언어
 - 학습/사용 (매우) 어려움
 - 상상하는 것은 무엇이든 구현할 수 있다.
 - 최근 왜 프로그래밍언어 학습자가 급증할까?
- ▶ 프로그래밍언어 중 Python, R
 - 프로그래밍언어 중 비교적 배우고 사용하기 쉽다. (그래도 어렵다)
 - 다른 프로그래밍언어에 비해 데이터분석/기계학습 환경이 잘 갖추어졌다.
 - 다양한 OS 지원, 무료 공개 ⇒ 전 세계에서 가장 많은 사용자, 개발자 ⇒ 학습 자료 풍부, 새로운 기법/모델의 신속한 탑재 ...

프로그래밍 언어: R vs. Python

▶ R

- 통계학자에 의해 개발된 데이터 분석 목적 언어
- 기본 데이터구조: **vector**, **data frame**
- 기본 함수에 통계 함수, 시각화 함수 포함
- 기계 학습 < 통계 분석

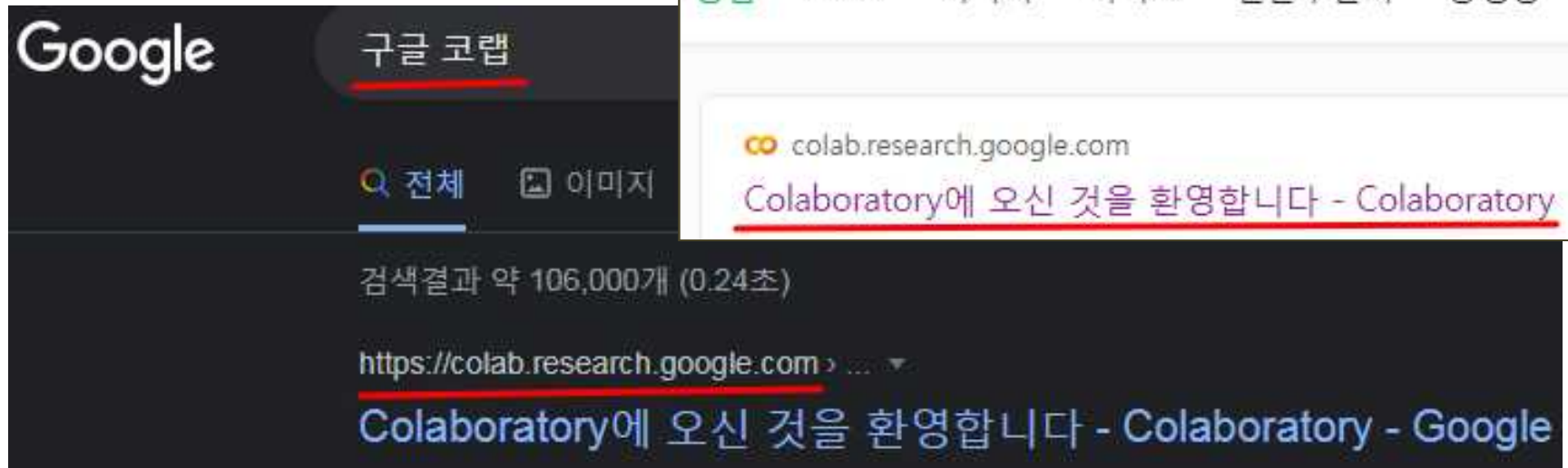
▶ Python

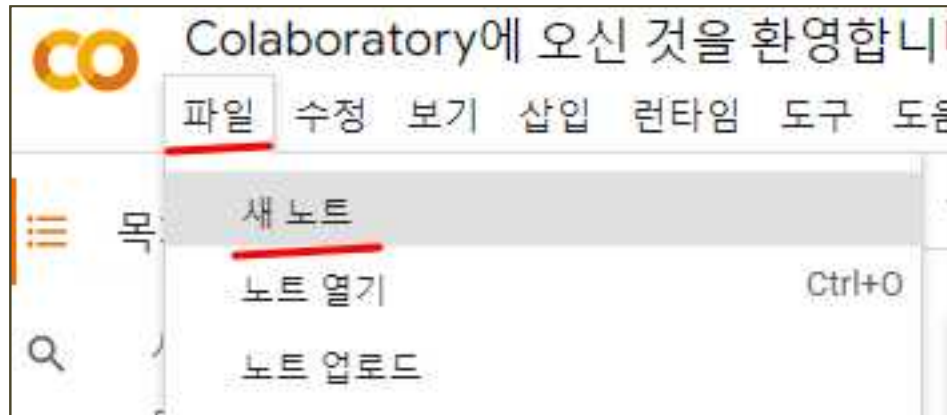
- 일반 목적의 프로그래밍 언어
 - 기본 데이터구조: **string**, **list**, **tuple**, **dictionary**, **set**
 - **numpy**, **scipy**, **matplotlib**, **pandas** 모듈: **ndarray**, **series**, **data frame**, 통계 함수, 시각화 함수 포함
 - 통계 분석 < 기계 학습
- ▶ 학습 난이도/시간: R < Python < Java, C, C++ 등
- ▶ 빅데이터 전문가: “**둘 다 학습해라!!!**”

Google Colab: 설치 없이 Python 사용

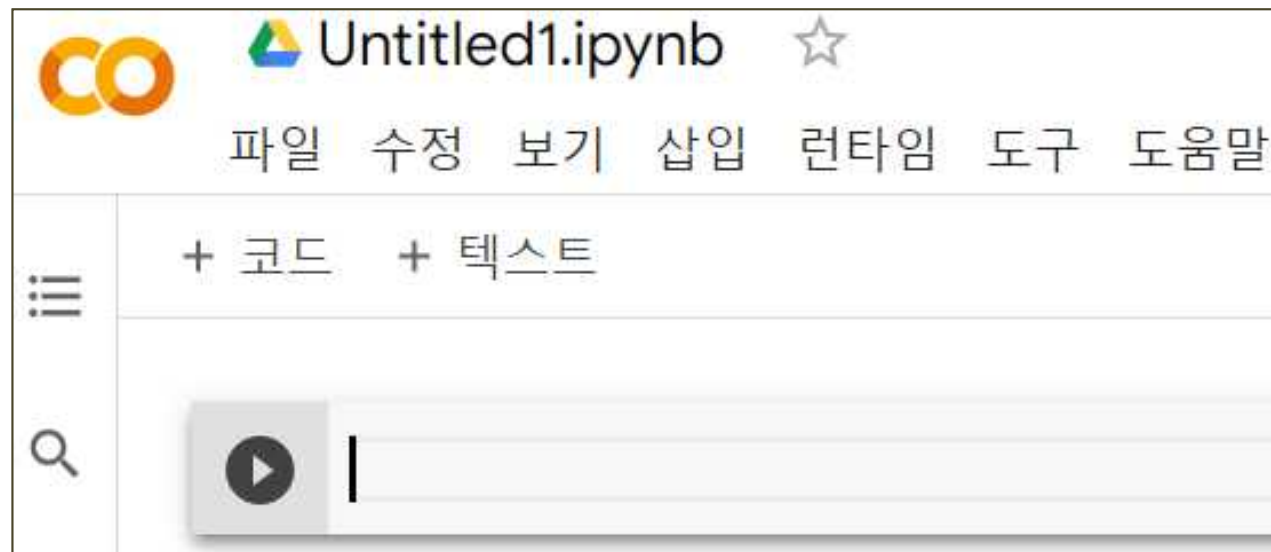
- ▶ Google 계정 로그인 후
- ▶ <https://colab.research.google.com/>
또는

북마크하기





또는 초기창
[최근 사용 - 새 노트]



Google Colab 제약 사항

- ▶ 기본: 무료 – 최대 세션 유지 시간 12시간
- ▶ 유료
 - 기간
 - Colab Pro(\$9.99/월, 100 컴퓨팅 단위)
 - Colab Pro+(\$49.99/월, 500 컴퓨팅 단위)
 - 컴퓨팅 단위: \$9.99/100컴퓨팅 단위, \$49.99/500컴퓨팅 단위
 - 더 빠른 GPU/TPU, 컴퓨팅 용량, 추가 메모리, 백그라운드 실행 등 제공
 - Colab Pro+ 최대 24시간 세션 유지

개인 컴퓨터: Python 설치 및 사용

- ▶ <https://www.python.org/downloads/>
- ▶ OS에 맞는 최신 버전 다운로드



- ▶ Python 및 Jupyter Notebook 설치 및 실행
 - <https://datadoctorblog.com/2022/10/26/Py-Jupyter-Notebook-Install>
 - 웹 검색을 통해 OS별, PC 환경별 설치 및 실행 방법 참조
 - Jupyter Notebook 설치 시 오류는 웹 검색을 통해 해결

특강을 어떻게 활용?

- ▶ 프로그래밍 언어 학습의 기본적 특성
 - 수학 공부와 유사
 - 기초가 충분해야 문제해결 능력 배양
 - 체계적이고 난이도 높은 학습
 - 장기간/꾸준한 자기주도 학습
 - 수업을 책으로 배울 수 없다! 악보를 볼 줄 안다고 악기를 연주할 수 있는 것은 아니다!
- ▶ 학자들을 대상으로 한 대부분 프로그래밍 언어 단기간 특강
 - 프로그래밍 언어 자체보다 특정 분석 기법 사용을 위한 코드 소개 초점
 - 코드 개념 및 원리 이해 없이 또는 약간의 이해 정도로 사용 가능
 - 프로그래밍 언어별 분석 기법 사용 코드는 대부분 웹 검색 가능
 - 무엇이 문제인가?
 - 코드 이해 능력 부족
 - 데이터 처리/분석을 위한 확장성, 문제해결 능력 부족

▶ 특강 내용

◦ 일반적 Python 기초

- 기본 구조: string, list/tuple, dictionary, set
- 코딩 기법: 제어문(if, for), 사용자 정의 함수
- 텍스트/데이터 처리/분석: 정규표현, 파일/경로, comprehension
- 몇 가지 기초 텍스트 분석 기법

▶ 특강 목적

◦ 내용

- 최소한 알아야 하는, 활용도가 높은 코드와 개념
- Python 기초 코드를 활용한 몇 가지 기초 텍스트 분석 기법
- 연습문제

◦ 목적

- 최소한 Python 기초 코드 읽고 이해하기 & 제시 코드 활용
- 향후 학습을 위한 코드, 용어, 개념 익숙해지기
- 지나친 기대/욕심/환상은 금지! 단기간 완벽한 이해는 불가능! ⇒ 아하~ 이런 것이 있구나! 이런 코드를 사용하면 이렇게 되는구나!

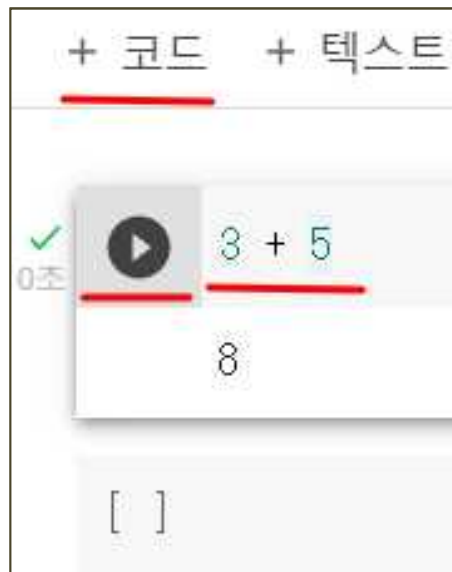
코드 입력 및 실행

- ▶ Google Colab 접속
 - Google 계정 로그인 후
 - <https://colab.research.google.com/>
- ▶ Notebook(노트북) 열기
 - [파일 - 새 노트]
- ▶ 노트북 기본 구성

북마크하기

③ 코드 셀 삽입

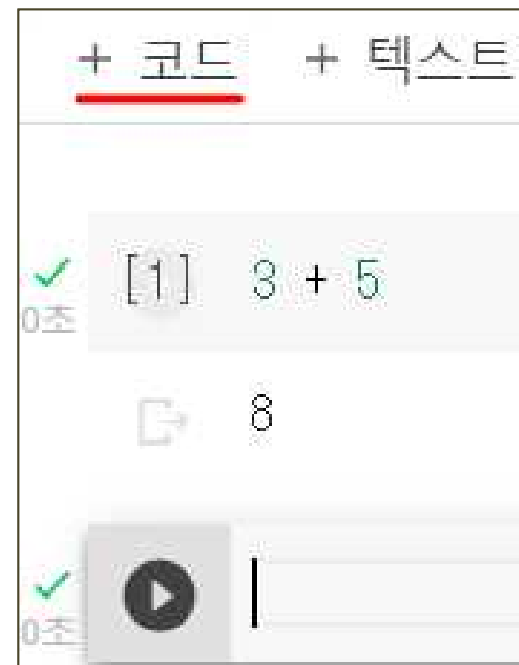
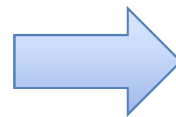
② 셀 실행



input cell ①

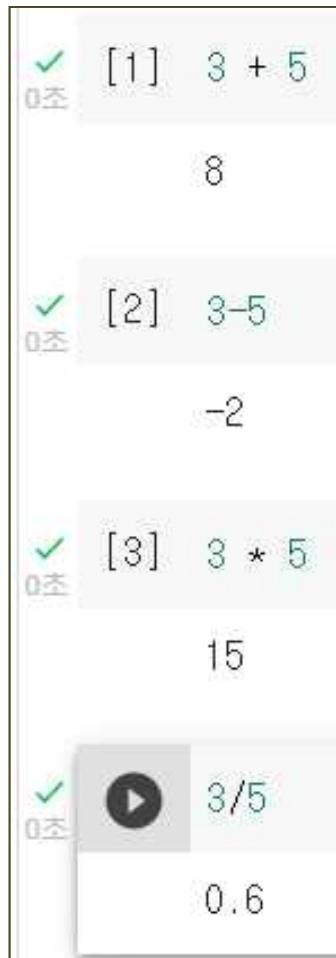
output cell

input cell



연습문제 1

다음 코드를 입력 및 실행



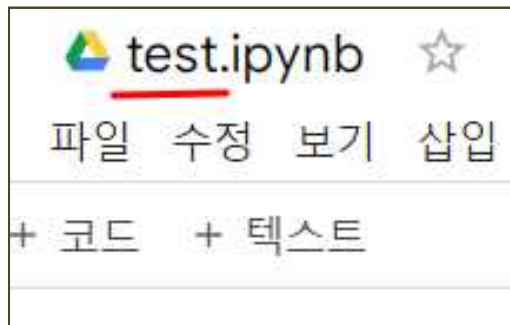
The screenshot shows a vertical list of four code cells, each with a green checkmark, a status indicator, and a result. The first cell shows the expression `[1] 3 + 5` with a status of '0초' and a result of '8'. The second cell shows `[2] 3-5` with a status of '0초' and a result of '-2'. The third cell shows `[3] 3 * 5` with a status of '0초' and a result of '15'. The fourth cell shows `[4] 3/5` with a status of '0초' and a result of '0.6'. A play button icon is visible next to the fourth cell.

Index	Expression	Status	Result
[1]	<code>3 + 5</code>	0초	8
[2]	<code>3-5</code>	0초	-2
[3]	<code>3 * 5</code>	0초	15
[4]	<code>3/5</code>	0초	0.6

산술 연산자

노트북 저장

▶ 노트북 파일명 변경

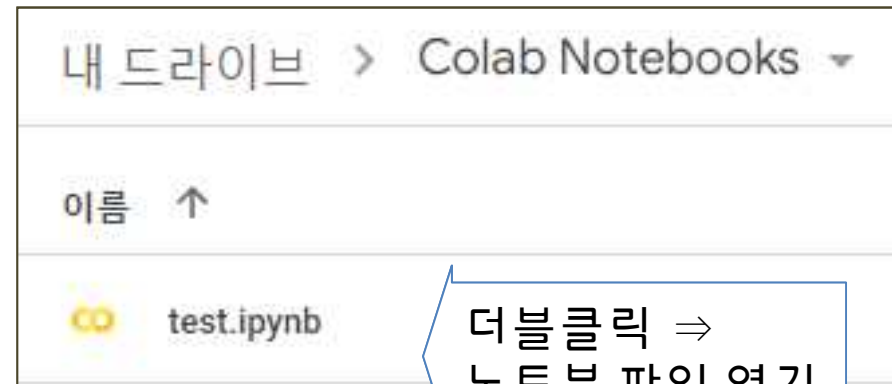


클릭 후 변경

▶ 노트북 파일 저장

◦ [파일-저장]

- 확장자 ipynb 파일
- Google Drive: Colab Notebooks 폴더



더블클릭 ⇒
노트북 파일 열기

노트북 열기

▶ Google Colab 초기 화면



▶ [파일 - 노트 열기]

atomic data type과 함수 type

```
[6] 30
    30

[7] type(30)
    int

[8] 30.5
    30.5

[9] type(30.5)
    float
```

```
[10] 'apple'
     'apple'

[11] type("apple")
     str

[12] "서울 특별시"
     '서울 특별시'

[13] type('서울 특별시')
     str
```

```
[14] True
     True

[15] type(True)
     bool

[16] False
     False

[17] type(False)
     bool
```


변수명 variable

- 길이 제한 없음
- 대소문자 구별
- 영문자, 숫자, 밑줄 조합
- 반드시 문자/밑줄 시작

```
[18] 3
      3
[20] a = 3
[21] a
      3
[22] a
      3
```

```
[22] a
      3
[23] a = 5
[24] a
      5
[25] a = a + 3
[26] a
      8
```

```
b
NameError
<ipython-input-27-89e6c98d9288> in
--> 1 b
NameError: name 'b' is not defined
SEARCH STACK OVERFLOW
[28] b = a + 3
[29] b
      11
```

치환문
assignment statement

기본 데이터 구조 data structure

```
[30] [1, 'abc', True]
```

```
[1, 'abc', True]
```

```
[31] type([1, 'abc', True])
```

```
list
```

```
[32] (1, 'abc', True)
```

```
(1, 'abc', True)
```

```
[33] type((1, 'abc', True))
```

```
tuple
```

```
[40] {1: 'a', 2: 'b', 3: 'b'}
```

```
{1: 'a', 2: 'b', 3: 'b'}
```

```
[41] type({1: 'a', 2: 'b', 3: 'b'})
```

```
dict
```

```
[42] {'b', 'a', 1}
```

```
{1, 'a', 'b'}
```

```
[43] type({1, 'a', 'b'})
```

```
set
```

문자열 string ⇒
데이터 구조

데이터 구조와 for 제어문

```
[44] for x in 'abc':  
    print(x)
```

```
a  
b  
c
```

```
[45] for x in [1, 'abc', True]:  
    print(x)
```

```
1  
abc  
True
```

```
[46] for x in {1: 'a', 2: 'b', 3: 'b'}:  
    print(x)
```

```
1  
2  
3
```

```
[47] for x in {1, 'a', 'b'}:  
    print(x)
```

```
b  
1  
a
```

데이터 구조와 함수 len

```
[48] a = 'abc'
      b = [1, 'abc', True]
      c = {1: 'a', 2: 'b', 3: 'b'}
      d = {1, 'a', 'b'}
```

```
[49] len(a)
```

```
3
```

```
[50] len(b)
```

```
3
```

```
[51] len(c)
```

```
3
```

```
[52] len(d)
```

```
3
```



```
len(30)
```

```
TypeError
```

```
<ipython-input-53-dceec4bf64cb> in <module>
```

```
----> 1 len(30)
```

```
TypeError: object of type 'int' has no len()
```



함수 function

```
sum([1, 2, 3, 4])
```

```
10
```

```
sum((1, 2, 3, 4))
```

```
10
```

```
sum({1: 10, 2: 20})
```

```
3
```

```
sum({1, 2, 3, 4})
```

```
10
```

```
int(3.5) # 정수변환
```

```
3
```

주석문
comment

```
float('3.5') #실수변환
```

```
3.5
```

```
# 문자열변환
```

```
str(3.5)
```

```
'3.5'
```

```
# 소수점 자릿수 및 반올림
```

```
round(3.5469, 2)
```

```
3.55
```

```
# 거듭제곱
```

```
# 또는 2 ** 3
```

```
pow(2, 3)
```

```
8
```

```
divmod(10, 3) # 몫과 나머지
```

```
(3, 1)
```

```
mean([1, 2, 3, 4])
```


```
NameError
```

확장 치환 연산자 augmented assignment operator

```
[19] a = 1
```

```
[20] a = a + 2  
a
```

3

```
 b = 3  
b += 5  
b
```

8

```
[22] a -= 5  
a
```

-2

```
[23] b *= 2  
b
```

16

```
[24] a /= b
```

```
[25] a
```


-0.125

```
[26] b **= 2 # 거듭제곱  
b
```

256

```
[27] b //= 2 # 몫  
b
```

128

```
 b %= 16 # 나머지  
b
```

0

연습문제 2

다음 네모 칸에 들어갈 출력결과는?

```
type('True')
```

```
type((3, 5, [7, 9]))
```

```
type({(1,2): [1, 2], 3: 'a'})
```

```
for i in [[1, 2], {3, 4, 5},  
          {1:'a', 10:'b', 100:'c'}]:  
    print(len(i))  
    print(sum(i))
```

```
a = 6  
b = 3  
a /= b
```

a

b