

2. Python 기초 ①

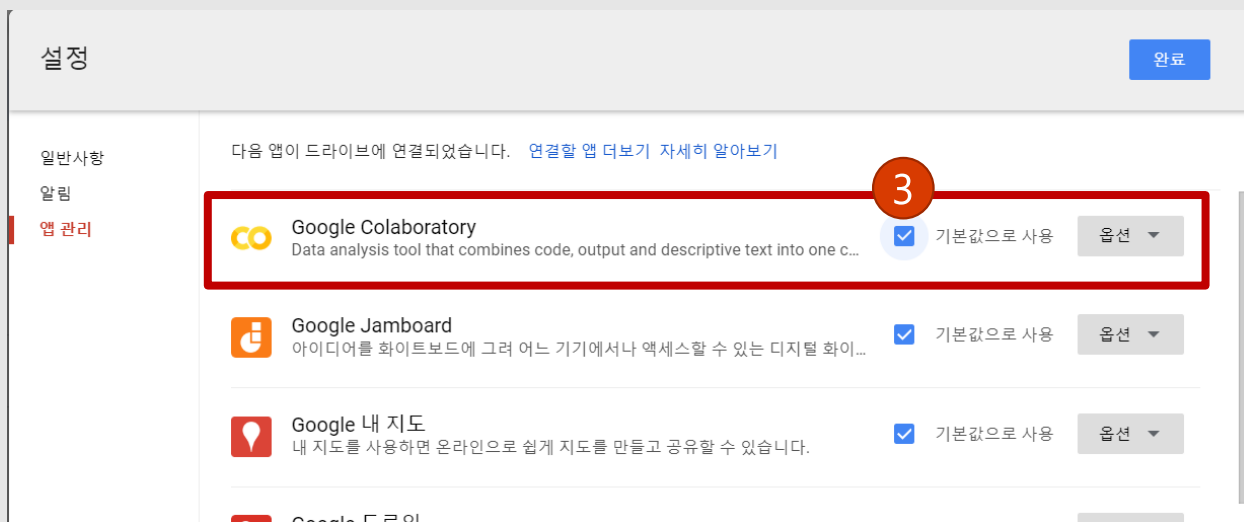
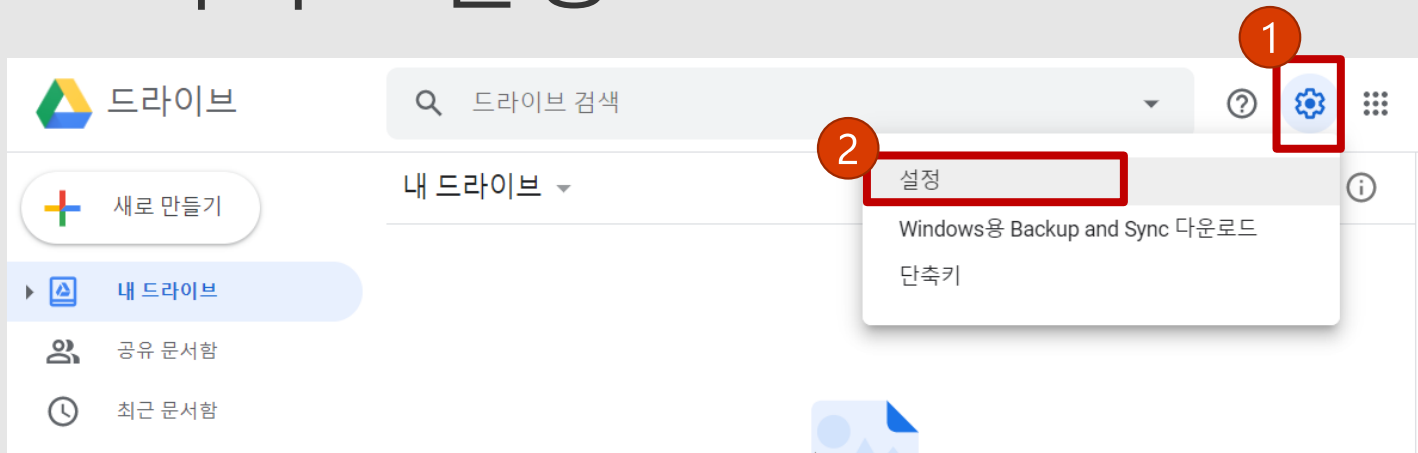


순서

- ✓ 환경준비
- ✓ Python 시작하기
- ✓ 자료형 : List & Dictionary

[환경준비]

구글 드라이브 설정



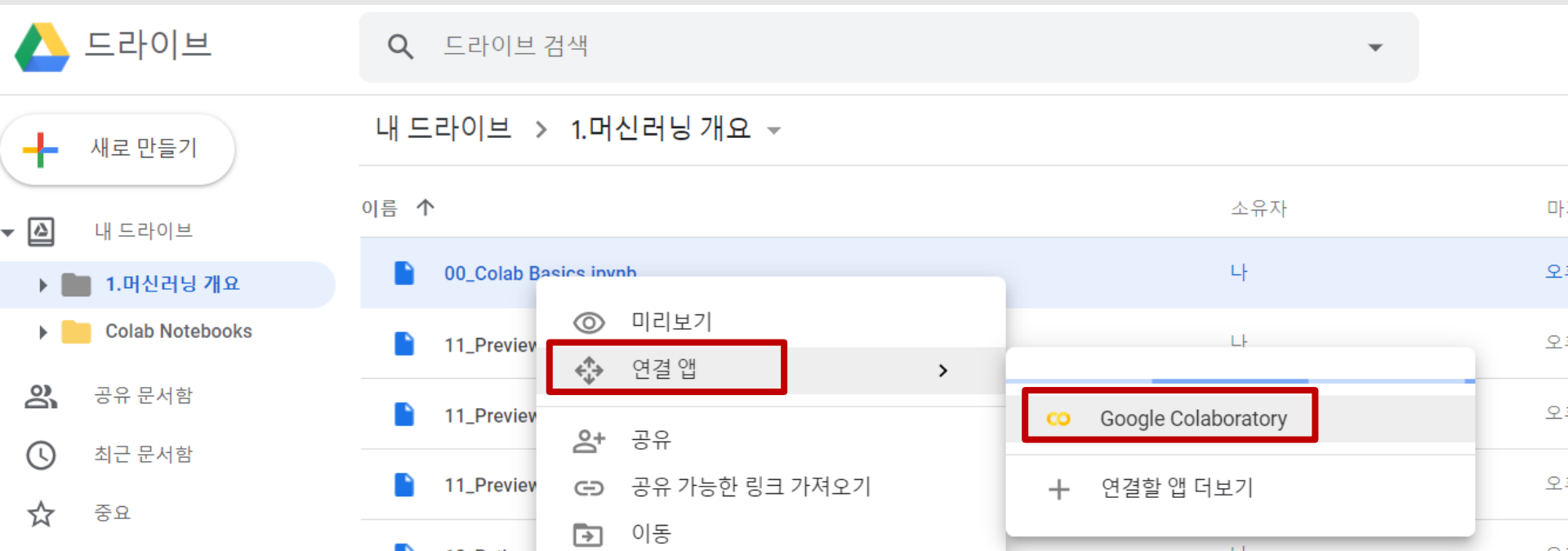
<https://colab.research.google.com>

The screenshot displays the Google Colaboratory web interface. At the top, the URL <https://colab.research.google.com> is shown. The main area features a file list with columns for '제목' (Title), '처음 연 시간' (First opened time), '마지막 연 시간' (Last opened time), and actions. The list includes 'Colaboratory에 오신 것을 환영합니다' (Welcome to Colaboratory) and 'Untitled0.ipynb'. A red box with the number '1' highlights the '새 PYTHON 3 노트' (New PYTHON 3 Note) button in the bottom right corner of the file list. Below the file list, a code cell is visible with the text `[1] print('안녕하세요?')` and the output '안녕하세요?'. A red box with the number '2' highlights the text '저장 : Ctrl + s' (Save : Ctrl + s) in the center of the interface.

1

2 저장 : Ctrl + s

- ✓ 첫 실행 시, 파일 오른쪽 클릭 > 연결 앱 > Google Colab
- ✓ 그 다음부터는 그냥 더블클릭



Google Colaboratory 단축키

1. 셀 선택, 편집 모드 진입/선택모드

화살표, Enter, Esc

2. 셀을 실행하는 방법들

Alt / Ctrl / Shift + Enter

3. 마크다운 셀 / 코드 셀 전환

[Ctrl + M] M, [Ctrl + M] Y

2. 코드 Line number

[Ctrl + M] L

5. 셀 생성

[Ctrl + M] A,B

6. 셀 삭제

[Ctrl + M] D

7. 링크 삽입 / 이미지 삽입

[링크이름](URL) / **![이미지이름]**(URL)

[Python 시작하기]

Help 이용하기

✓ ? 함수명

- 함수에 대한 설명

```
In [1]: # help '?' 이용,  
        ?print
```

Docstring:
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:

✓ () 안에 Shift+Tab

- Tooltip 처럼 설명이 나타남

```
# () 안에 Shift + Tab  
sum()
```

Signature: sum(iterable, start=0, /)

Docstring:
Return the sum of a 'start' value (default: 0) plus an iterable of numbers

Python 연산자

비교

연산자	설명
>	큼
<	작음
>=	크거나 같음
<=	작거나 같음
!=	같지 않음
==	같음

계산

연산자	설명
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈
//	나눗셈의 몫
%	모듈로 (나눗셈의 나머지)
**	지수 연산자

Basic data types

✓ 변수

- 변수 이름 : 대소문자 구분
- 각 변수는 하나의 단일 값을 저장

Python

```
height = 1.73  
tall = True
```

✓ Types

- Type() : 변수의 타입을 확인
- float - real numbers
- int - integer numbers
- str - string, text
- bool - True, False

Basic data types

✓문자열을 다루는 함수들

- `len()` : 문자열의 길이를 반환
- `.format()` : 문자열 안의 특정 위치에 값을 포함시킴.
- `.capitalize()` : 알파벳 첫문자를 대문자로.
- `.upper()` : 알파벳 전체를 대문자로.
- `.rjust()` : 오른쪽 정렬
- `.replace('o', 'go')` : 'o'를 'go'로 변경
- `.strip()` : 앞 뒤 공백 제거

Basic data types

.format

String

Old

```
'%s %s' % ('one', 'two')
```

New

```
'{} {}'.format('one', 'two')
```

Output

```
o n e   t w o
```

Number

```
'%d %d' % (1, 2)
```

```
'{} {}'.format(1, 2)
```

```
1 2
```

실습 : Python_basics ①

[흐름제어]

Control 1/3

✓조건문

if 조건1:	조건1이 True이면
처리1	처리1
elif 조건2:	조건1이 아니고 and 조건2이면
처리2	처리2
else:	아니면
처리3	처리3

- 제어문 블록은 ':' 와 (들여쓰기) 로 구분

Python

```
my_score = 73

if my_score >= 90:
    my_grade = 'A'
elif my_score >= 80:
    my_grade = 'B'
elif my_score >= 70:
    my_grade = 'C'
elif my_score >= 60:
    my_grade = 'E'
else:
    my_grade = 'F'

print(my_grade)
```


Control 2/3

✓반복문 : for

for i in 리스트 :
처리

첫번째부터 마지막
값까지 하나씩 가져다
처리

- range(5) : 1~5까지 자연수
- [.....,,] : 리스트를 나타냄
- enumerate() 순서가 있는 자료형을 입력받아, 인덱스와 값을 포함하는 오브젝트로 리턴.

Python

```
for i in range(5):  
    print(i)
```

```
fruits = ['apple', 'banana', 'cherry']  
print(type(fruits))  
for i in fruits:  
    print(i)
```

```
for index, value in enumerate(fruits):  
    print(index, value)
```

Control 3/3

✓반복문 : while

while 조건 :

처리

조건변경문

조건이 참인 경우
처리

Python

```
i = 0
while i < len(fruits):
    print(fruits[i])
    i += 1
```

실습 : Python_basics ②

[자료형 : List와 Dictionary]

자료형(Container) Lists

✓ 기존 변수의 한계

- 단일 값 저장 → 많은 데이터 처리 곤란

✓ List : [a, b, c]

- 값의 집합
- 하나의 리스트에 **여러 형식의** 데이터 저장 가능

Python

```
age1 = 44  
age2 = 44  
age3 = 14  
age4 = 12  
age5 = 8
```

```
Family_age = [44,44,14,12,8]
```

```
Family = ['Dad',44, 'Mom',44,  
          'Son1',14, 'Son2',12, 'Son3',8]
```

자료형(Container) Lists

✓리스트의 데이터 사용하기

- Index number

```
family = ['Dad', 44, 'Mom', 44, 'Son1', 14, 'Son2', 12, 'Son3', 8]
```

Index :	0	1	2	3	4	5	6	7	8
9									
	-10	-9	-8	-7	-6	-5	-4	-3	-2
-1									

Python

```
Family[0]  
'Dad'
```

```
Family[-3]  
12
```

자료형(Container) Lists

✓리스트의 데이터 사용하기

- Slicing : [Start Index(포함) : End Index(미포함)]

```
family = ['Dad', 44, 'Mom', 44, 'Son1', 14, 'Son2', 12, 'Son3', 8]
```

Index :	0	1	2	3	4	5	6	7	8	9
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Python

```
Family[2:6]  
['Mom', 44, 'Son1', 14]
```

Python

```
Family[6:]  
['Son2', 12, 'Son3', 8]
```

자료형(Container) Lists

✓리스트의 데이터 변경하기

```
family = ['Dad', 44, 'Mom', 44, 'Son1', 14, 'Son2', 13, 'Son3', 8]
```

Index :	0	1	2	3	4	5	6	7	8	9
	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Python

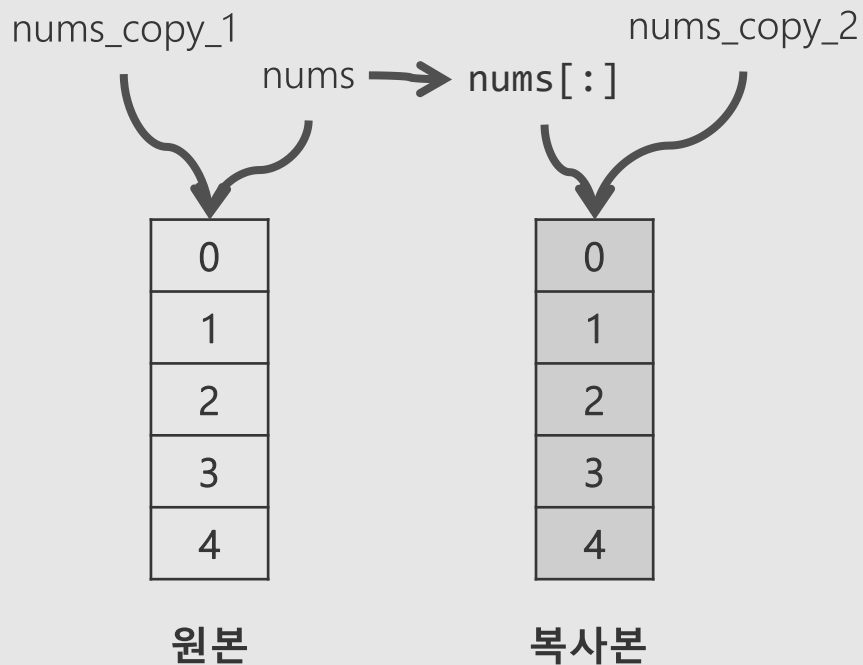
```
Family[7] = 13
```

Python

```
Family[8:] = ['Son4', 9]
```


자료형(Container) Lists

✓리스트 복사하기



Python

```
nums_copy_1 = nums
nums_copy_2 = nums[:]
print(nums_copy_1)    #[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums_copy_2)    #[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

nums_copy_1[0] = 'a'
print(nums_copy_1)    #[a, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums_copy_2)    #[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums)           #[a, 1, 2, 3, 4, 5, 6, 7, 8, 9]

nums_copy_2[0] = 'b'
print(nums_copy_1)    #[a, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums_copy_2)    #[b, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(nums)           #[a, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

실습 : Python_basics ③

자료형(Container) Lists

✓리스트와 반복문

- 반복할 범위를 리스트로 지정하여 반복

Python

```
animals = ['cat', 'dog', 'monkey']  
for animal in animals:  
    print(animal)
```

- 반복문 확장문법

for x in nums:

squares.append(x ** 2)

→ squares = [x ** 2 for x in nums]

Python

```
nums = [0, 1, 2, 3, 4]  
squares = []
```

```
for x in nums:  
    squares.append(x ** 2)
```

```
print(squares)
```

```
squares = [x ** 2 for x in nums]
```

실습 : Python_basics ④

자료형(Container) Dictionary

✓ Dictionary 구조 : {key1 : value1, key2 : value2, ... }

Python

```
# dictionary 생성
d = {'cat': 'cute', 'dog': 'furry'}
print(d)
```

```
# dictionary 특정 값 조회
print(d['cat'])      # "cute"
```

```
# key 값이 들어있는지 확인
print('cat' in d)    # "True"
print('sheep' in d)  # "False"
```

```
# 새로운 (key, value) 추가
d['fish'] = 'wet'
print(d['fish'])      # "wet"
```

```
# key가 없는 경우, default value 사용
print(d.get('monkey', 'N/A')) # "N/A"
print(d.get('fish', 'N/A'))   # "wet"
```

```
# 요소 삭제
del d['fish']
print(d.get('fish', 'N/A')) # "N/A"
```

자료형(Container) Dictionary

✓ Dictionary 와 반복문

- list와 비슷하나, key와 value로 제어

Python

```
d = {'person': 2, 'cat': 4, 'spider': 8}
for key in d:
    value = d[key]
    print('A {} has {} legs'.format(key, value))

# 딕셔너리에서 루프로 키와 값을 동시에 가져오려면 .items()
for key, value in d.items():
    print('A %s has %d legs' % (key, value))
```

- 확장문법

Python

```
animals = {'cat': 11, 'dog': 7, 'fish': 15, 'horse': 3}
animals10 = {k:v for k, v in animals.items() if v > 10}
print(animals_over_10) # "{ 'cat': 11, 'fish': 15}"
```

실습 : Python_basics ⑤

함수 – Method

✓ call functions *on* objects

Python

```
name = "hky"
```

```
age = 45
```

```
customer = ["hky", 44, "nkm", 44]
```

Type

Methods

str

.capitalize(), .replace()

float

.bit_length()

list

.index(), .count()

Packages

✓ 개요

- Directory of Python Scripts
- Each script = module
- Specify functions, methods, types
- Thousands of packages available

✓ 설치

- Download get-pip.py
- Terminal:
python3 get-pip.py
pip3 install numpy

Packages

✓ Import

- 설치된 패키지 사용하기 위해 코드 페이지에 로딩

✓ import numpy

Python

```
import numpy
```

```
array([1,2,3])
```

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

```
numpy.array([1,2,3])
```

```
import numpy as np
```

```
np.array([1,2,3])
```

• from numpy import array

Python

```
from numpy import array
```

```
np_fam = array(fam_ext)
```