

# 외장함수

## ■ 외장함수

- 파이썬에서 제공하는 기본 라이브러리
- 내장함수와는 다르게 모듈을 불러들이는 import를 먼저 수행해야 됨

---

sys	pickle	os
shutil	glob	time
datetime / dateutil	calendar	random
pathlib	csv	json

## ■ sys

- 파이썬이 제공하는 변수 및 함수를 직접 제어할 수 있게 해주는 모듈
- sys.argv
  - 명령모드로 실행하면서 전달된 인자 확인

```
import sys  
print(sys.argv)
```

```
['c:\\Users\\GGoReb\\anaconda3\\lib\\site-  
packages\\ipykernel_launcher.py', '--ip=127.0.0.1', '--stdin=9052', '--  
control=9050', '--hb=9049', '--Session.signature_scheme="hmac-sha256"',  
 '--Session.key=b"69132ded-ced8-4e25-952c-5e239438a6b1"', '--shell=9051',  
 '--transport="tcp"', '--iopub=9053', '--  
f=c:\\Users\\GGoReb\\AppData\\Roaming\\jupyter\\runtime\\kernel-v2-  
6272J1UkJQn0h701.json']
```

- sys.exit()
  - 프로그램 강제 종료

```
import sys  
print('프로그램 시작')  
sys.exit()  
print('프로그램 종료') # 종료되었으므로 실행되지 않음
```

## ■ sys

### ● sys.path

- 파이썬의 모듈이 저장되어 있는 위치를 나타냄
- 외부 모듈을 사용하기 위해 특정 위치 지정 가능

```
import sys
print(sys.path)

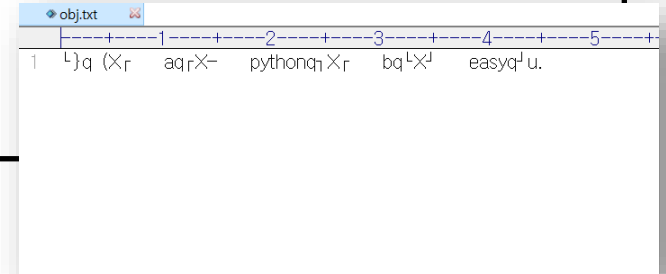
sys.path.append('c:/windows')
print(sys.path)
```

```
'c:\\Users\\GGoReb\\anaconda3\\python39.zip',
'c:\\Users\\GGoReb\\anaconda3\\DLLs',
'c:\\Users\\GGoReb\\anaconda3\\lib', 'c:\\Users\\GGoReb\\anaconda3', '',
'c:\\Users\\GGoReb\\anaconda3\\lib\\site-packages',
'c:\\Users\\GGoReb\\anaconda3\\lib\\site-packages\\win32',
'c:\\Users\\GGoReb\\anaconda3\\lib\\site-packages\\win32\\lib',
'c:\\Users\\GGoReb\\anaconda3\\lib\\site-packages\\Pythonwin',
'c:/windows']
```

## ■ pickle

- 객체의 형태를 그대로 유지하면서 파일에 저장하고 불러올 수 있게 하는 모듈
- pickle.dump()
  - Dictionary 객체의 형태를 그대로 유지하면서 저장

```
import pickle
file = open('obj.bin', 'wb')
data = { 'a': 'python', 'b': 'easy' }
pickle.dump(data, file)
file.close()
```



- pickle.load()
  - 객체의 형태를 그대로 유지하면서 불러오기

```
import pickle
file = open('obj.bin', 'rb')
data = pickle.load(file)
print(data)
file.close()
```

```
{'a': 'python', 'b': 'easy'}
```

## ■ 연습문제

- 좌표(x, y) 정보를 가지는 NowData클래스를 파일로 저장 후 다시 읽어서 정보 확인

```
import pickle
class NowData:
    def __init__(self):
        self.x = 0
        self.y = 0
    def move(self, x, y):
        self.x = x
        self.y = y
    def get_location(self):
        return self.x, self.y

data = NowData()
data.move(100, 200)

# 코드 작성
```

저장된 좌표 (100, 200)

## ■ OS

- 환경변수, 디렉토리, 파일 등의 운영체제 자원을 제어할 수 있게 해주는 모듈
- os.environ
  - 환경변수 확인

```
import os
print(os.environ)
print('-' * 30)
print(os.environ['PATH'])
```

```
environ({'ALLUSERSPROFILE': 'C:\\ProgramData', 'APPDATA': 'C:\\Users\\GGoReb\\AppData\\Roaming', 'APPLICATION_INSIGHTS_NO_C
'COMMONPROGRAMW6432': 'C:\\Program Files\\Common Files', 'COMPUTERNAME': 'DESKTOP-P3J987A', 'COMSPEC': 'C:\\Windows\\system
'C:\\Users\\GGoReb\\anaconda3\\python.exe', 'CONDA_SHLVL': '1', 'DRIVERDATA': 'C:\\Windows\\System32\\Drivers\\DriverData',
'C:\\Users\\GGoReb\\AppData\\Local', 'LOGONSERVER': '\\\\DESKTOP-P3J987A', 'NUMBER_OF_PROCESSORS': '8', 'ONEDRIVE': 'C:\\Us
w64\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\usr\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\bin;C:\\Users\\GGoReb\\anac
Files\\ojdkbuild\\java-11-openjdk-11.0.15-1\\bin;C:\\Program Files\\Bandizip;C:\\Program Files\\Git\\cmd;C:\\Users\\GGoReb\\
w64\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\usr\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\bin;C:\\Users\\GGoReb\\anac
'AMD64', 'PROCESSOR_IDENTIFIER': 'Intel64 Family 6 Model 142 Stepping 12, GenuineIntel', 'PROCESSOR_LEVEL': '6', 'PROCESSOR
'%ProgramFiles%\\WindowsPowerShell\\Modules;C:\\Windows\\system32\\WindowsPowerShell\\v1.0\\Modules', 'PUBLIC': 'C:\\Users\\
'C:\\Users\\GGoReb\\AppData\\Local\\Temp', 'USERDOMAIN': 'DESKTOP-P3J987A', 'USERDOMAIN_ROAMINGPROFILE': 'DESKTOP-P3J987A',
'C:\\Users\\GGoReb\\AppData\\Roaming\\Code\\CachedData\\da76f93349a72022ca4670c1b84860304616aaa2', 'VSCODE_CWD': 'C:\\Users
'{"locale":"ko","availableLanguages":{},"_languagePackSupport":true}', 'VSCODE_PID': '4968', 'WINDIR': 'C:\\Windows', 'ZES_
-----
c:\\Users\\GGoReb\\anaconda3;C:\\Users\\GGoReb\\anaconda3;C:\\Users\\GGoReb\\anaconda3\\Library\\mingw-
w64\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\usr\\bin;C:\\Users\\GGoReb\\anaconda3\\Library\\bin;C:\\Users\\GGoReb\\anaconda3\\Scripts;C
11.0.15-1\\bin;C:\\Program Files\\Bandizip;C:\\Program Files\\Git\\cmd;C:\\Users\\GGoReb\\anaconda3;C:\\Users\\GGoReb\\anaconda3\\Librar
Code\\bin
```

## ■ OS

### ● os.chdir()

- 현재 디렉토리 위치 변경

```
import os  
os.chdir('/study')
```

### ● os.getcwd()

- 현재 디렉토리 위치 확인

```
import os  
os.getcwd()
```

### ● os.system()

- 시스템(운영체제 또는 외부 프로그램) 명령어 호출하기

```
import os  
os.system('echo 내용 작성 > test.txt')
```



## ■ os

### ● os.popen()

- 시스템(운영체제 또는 외부 프로그램) 명령어 호출 결과 받기

```
import os
os.chdir('/study')

content = os.popen('dir/w')

while True:
    line = content.readline()
    if not line:
        break
    print(line, end='')

content.close()
```

c 드라이브의 볼륨에는 이름이 없습니다.

볼륨 일련 번호: 0EB2-84EC

c:\study 디렉터리

[.]	[..]	test.txt
-----	------	----------

1개 파일	12 바이트
-------	--------

2개 디렉터리	73,833,459,712 바이트 남음
---------	-----------------------

## ■ OS

### ● os.mkdir() / os.makedirs()

- 디렉토리 생성

```
os.mkdir('/study/first')  
os.makedirs('/study/second/a/b/c')
```

### ● os.rmdir()

- 디렉토리 삭제 (디렉토리가 비어있어야 가능)

```
os.rmdir('/study/second/a/b/c')
```

### ● os.rename()

- 디렉토리 또는 파일 이름 변경

```
os.rename('/study/first', '/study/third')
```

### ● os.remove()

- 파일 삭제

```
os.remove('/study/test.txt')
```

## ■ 연습문제

- 아래와 같은 구조로 디렉토리와 파일 생성하기

```
study/  
  python_dir/  
    a/  
      a.txt  
    b/  
      b.txt
```

# 코드 작성

## ■ shutil

### ● shutil.copy()

- 파일 복사

```
import shutil
import os

os.system('echo 내용 작성 > test.txt')
shutil.copy('/study/test.txt', '/study/test2.txt')
```

## ■ glob

### ● glob.glob()

- 파일 목록 확인

```
import glob

fileList = glob.glob('/study/*o*')
print(fileList)

fileList = glob.glob('/study/*[0-9].txt')
print(fileList)
```

## ■ 연습문제

- Windows/System32/drivers/etc/hosts 파일을 study 디렉토리로 복사

```
# 코드 작성
```

- /Users/[사용자명] 디렉토리에서 . 으로 시작하는 디렉토리 또는 파일 개수 확인하기

```
# 코드 작성
```

## ■ time

- 시간과 관련된 여러 기능을 제공하는 모듈

- time.time()

- 1970년 1월 1일 0시 0분 0초를 기준으로 지난 시간을 초 단위로 출력

```
print(time.time())
```

```
1660353606.2661853
```

- time.localtime()

- time.time() 으로 반환된 값을 이용하여 년, 월, 일, 시, 분, 초 단위로 출력

```
date = time.time()
print(time.localtime(date))
print(time.localtime(date).tm_year)
print(time.localtime(date).tm_zone)
```

```
time.struct_time(tm_year=2022, tm_mon=8, tm_mday=13, tm_hour=10,
tm_min=21, tm_sec=54, tm_wday=5, tm_yday=225, tm_isdst=0)
```

## ■ time

### ● time.strftime()

- 형태를 지정하여 시간 정보 출력

```
time.strftime(  
    '출력할 형식 포맷 코드', time.localtime(time.time())  
)
```

코드	의미	출력 예	코드	의미	출력 예
%Y	년	2002	%y	년	02
%m	월	01 - 12	%B	월	January
			%b	월	Jan
%d	일	01 - 31			
%H	시	00 - 23	%I	시	01 - 12
%M	분	00 - 59			
%S	초	00 - 59			
%w	요일	0 - 6	%A	요일	Sunday
			%a	요일	Sun
%p	오전, 오후	AM, PM			
%c	날짜와 시간	Sat Aug 13 10:44:21 2022	%Z, %j, %U, %W ...		

## ■ time

### ● time.sleep()

- 일정시간 동안 대기 후 동작

```
for i in range(1, 6):  
    print(i)  
    time.sleep(1)
```

### ● time.strptime()

- 날짜 형태의 문자열을 time으로 변환

```
t = time.strptime('2022-12-01 111213', '%Y-%m-%d %H%M%S')  
print(t)
```

```
time.struct_time(tm_year=2022, tm_mon=12, tm_mday=1, tm_hour=11,  
tm_min=12, tm_sec=13, tm_wday=3, tm_yday=335, tm_isdst=-1)
```



## ■ datetime / dateutil

### ● datetime.datetime.now()

- 현재 시각 출력

```
import datetime
dt = datetime.datetime.now()
print(dt)
```

- 각 정보 확인

```
print(dt.year, dt.month, dt.day)
print(dt.hour, dt.minute, dt.second)
print(dt.microsecond)
print(dt.weekday()) # 0:월, 1:화, 2:수, 3:목, 4:금, 5:토, 6:일
```

### ● datetime.datetime.strftime()

- 형태를 지정하여 시간 정보 출력

```
datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S %w')
```

## ■ datetime / dateutil

### ● datetime.datetime.strptime()

- 날짜 형태의 문자열을 datetime으로 변환

```
d = datetime.datetime.strptime(  
    '2022-12-01 111213', '%Y-%m-%d %H%M%S')  
print(d)
```

### ● dateutil.parser()

- 날짜 형태의 문자열을 datetime으로 변환

```
from dateutil.parser import parse  
print(parse('2022-12-01 12:13:14'))
```

## ■ datetime / dateutil

### ● 날짜 간 연산

- 날짜를 지정하여 datetime 생성

```
dt1 = datetime.datetime(2022, 12, 25)
dt2 = datetime.datetime.now()
result = dt1 - dt2
print(result)
```

```
133 days, 12:16:56.215446
```

- 각 정보 출력

```
print(dir(result))
print(result.days)
print(result.total_seconds())
```

- datetime.timedelta()

```
now_time = datetime.datetime.now()
add_time = datetime.timedelta(days=100)
print(now_time + add_time)
```

## ■ 연습문제

- "연도-월-일 시:분:초" 형식으로 현재시각 출력하기

```
# 코드 작성
```

- 현재 일자에서 100일 뒤의 날짜를 "연도-월-일 시:분:초" 형식으로 출력

```
# 1. time.time() + 초  
# 2. datetime.timedelta(days=일)
```

## ■ calendar

- 달력을 볼 수 있게 해주는 모듈
- calendar.prcal()

```
import calendar  
calendar.prcal(2022)
```

```
2022  
  
January          February          March  
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su  
                1 2          1 2 3 4 5 6          1 2 3 4 5 6  
 3 4 5 6 7 8 9      7 8 9 10 11 12 13      7 8 9 10 11 12 13  
10 11 12 13 14 15 16 14 15 16 17 18 19 20      14 15 16 17 18 19 20  
17 18 19 20 21 22 23 21 22 23 24 25 26 27      21 22 23 24 25 26 27  
24 25 26 27 28 29 30 28          28 29 30 31  
31  
  
April            May              June  
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su  
                1 2 3          1          1 2 3 4 5  
 4 5 6 7 8 9 10      2 3 4 5 6 7 8      6 7 8 9 10 11 12  
11 12 13 14 15 16 17 9 10 11 12 13 14 15      13 14 15 16 17 18 19  
18 19 20 21 22 23 24 16 17 18 19 20 21 22      20 21 22 23 24 25 26  
25 26 27 28 29 30      23 24 25 26 27 28 29      27 28 29 30  
30 31  
  
July             August           September  
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su  
                1 2 3      1 2 3 4 5 6 7      1 2 3 4  
 4 5 6 7 8 9 10      8 9 10 11 12 13 14      5 6 7 8 9 10 11  
11 12 13 14 15 16 17 15 16 17 18 19 20 21      12 13 14 15 16 17 18  
18 19 20 21 22 23 24      22 23 24 25 26 27 28      19 20 21 22 23 24 25  
25 26 27 28 29 30 31      29 30 31      26 27 28 29 30  
  
October          November         December  
Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su  
                1 2      1 2 3 4 5 6      1 2 3 4  
 3 4 5 6 7 8 9      7 8 9 10 11 12 13      5 6 7 8 9 10 11  
10 11 12 13 14 15 16 14 15 16 17 18 19 20      12 13 14 15 16 17 18  
17 18 19 20 21 22 23      21 22 23 24 25 26 27      19 20 21 22 23 24 25  
24 25 26 27 28 29 30      28 29 30      26 27 28 29 30 31  
31
```

## ■ calendar

### ● calendar.prmonth()

- 연도, 월을 입력받은 후 해당 월의 달력을 출력

```
calendar.prmonth(2022, 8)
```



August 2022						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

### ● calendar.weekday()

- 연도, 월, 일을 입력받은 후 날짜에 해당하는 요일 정보를 출력

```
print(calendar.weekday(2012, 3, 25))
```

### ● calendar.monthrange()

- 연도, 월, 일을 입력받은 후 해당 월의 첫번째 요일과 마지막 일자를 출력

```
print(calendar.monthrange(2012, 3))
```

## ■ 연습문제

### ● 2004년 8월의 달력 출력하기

August 2004						
Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

### ● 2009년 11월 19일의 요일을 출력하기

```
week_dict = {  
    0: '월', 1: '화', 2: '수', 3: '목',  
    4: '금', 5: '토', 6: '일'  
}
```

```
# 코드 작성
```

November 2009						
Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

## ■ 연습문제

- 2022년 2월의 달력과 요일이 토요일인 날짜만 출력하기

# 코드 작성

February 2020

Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	
29						

토요일 : 1  
토요일 : 8  
토요일 : 15  
토요일 : 22  
토요일 : 29



## ■ random

- 난수를 만들어주는 모듈

- random.random()

- $0.0 \leq N < 1.0$  범위의 난수값(실수) 조회

```
import random  
print(random.random())
```

- random.randint(A, B)

- $A \leq N \leq B$  범위의 난수값(정수) 조회

```
print(random.randint(0, 3))
```

- random.choice()

- 반복이 가능한 자료형에서 무작위로 요소 하나 추출

```
print(random.choice([1, 2, 3, 4, 5]))
```

- random.shuffle()

- List 자료형의 요소를 무작위로 정렬

```
items = [1, 2, 3, 4, 5]  
random.shuffle(items)  
print(items)
```

## ■ 연습문제

- 0.0 보다 크거나 같고 5.0 보다 적은 난수(실수) 생성하기

```
# round() 함수를 이용하여 소수점 둘째 자리까지만 표시  
# ex) round(3.1555, 2) → 3.16  
# 코드 작성
```

4.22

- 주사위 두개를 던진 결과를 출력하고 두 눈이 같으면 종료되는 코드 작성하기

```
count = 0  
  
# 코드 작성  
  
print('주사위를 던진 횟수: %s' % count)
```

```
(1, 3)  
(3, 2)  
(2, 3)  
(6, 6)  
주사위를 던진 횟수: 4
```

- 1 ~ 25 요소를 무작위로 섞어서 출력하기

```
# bingo 1 ~ 25 숫자 입력  
bingo = list()  
for i in range(1, 26):  
    if i < 10: i = '0' + str(i)  
    bingo.append(str(i))  
print(bingo)  
  
# 코드 작성
```

```
23 05 25 03 15  
16 06 19 11 07  
02 24 21 18 01  
10 12 09 20 22  
17 14 08 13 04
```

## ■ pathlib

- Python 3.6 부터 표준 라이브러리로 지원
- os의 디렉토리 생성/확인/변경/삭제, glob의 디렉토리/파일 목록 조회, open()의 파일 읽기/쓰기 등의 기능 제공
- Path.cwd()
  - 현재 디렉토리 위치 확인

```
from pathlib import Path
print(Path.cwd())

import os
os.chdir('/study')
print(Path.cwd())
```

- Path('경로').mkdir()
  - 디렉토리 생성

```
Path('c:/study/first').mkdir()
```

## ■ pathlib

- Path('경로').glob() : 파일 목록 확인

```
filelist = list(Path('c:/study').glob('*'))  
print(filelist)
```

- Path('경로').open('모드') : 파일 열기

```
path = Path('c:/study/test.txt')  
file = path.open('r')  
print(file.readlines())
```

- Path('경로').read\_text() : 파일 내용 읽기

```
path = Path('c:/study/test.txt')  
print(path.read_text())
```

- Path('경로').write\_text('내용') : 파일 내용 쓰기

```
path = Path('c:/study/test3.txt')  
path.write_text('내용 입력')
```

## ■ CSV

- 쉼표로 구분된 값 (Comma Separated Values)
- 텍스트로 구성된 간단한 스프레드시트

```
empno,ename,job,sal,comm,deptno  
7369,smith,clerk,800,,20  
7499,allen,salesman,1600,300,30  
7521,ward,salesman,1250,500,30  
7566,jones,manager,2975,,20  
7654,martin,salesman,1250,1400,30  
7698,blake,manager,2850,,30  
7782,clark,manager,2450,,10  
7788,scott,analyst,3000,,20
```

## ■ CSV

### ● csv.reader()

- csv 파일 내용 읽어오기

```
import csv
file = open('emp.csv')
reader = csv.reader(file)
for row in reader:
    print(reader.line_num, row)
file.close()
```

### ● csv.writer()

- csv 파일 내용 쓰기

```
file = open('output.csv', 'w', newline='')
writer = csv.writer(file)
writer.writerow(['name', 'age', 'job'])
writer.writerow(['John', 20, 'student'])
writer.writerow(['Jenny', 30, 'developer'])
writer.writerow(['Nate', 30, 'teacher'])
writer.writerow(['Julia', 40, 'dentist'])
file.close()
```

## ■ CSV

### ● csv.DictReader()

- Dictionary 형태로 내용 읽기

```
import csv
file = open('emp.csv')
reader = csv.DictReader(file)
for row in reader:
    print(row)
file.close()
```

```
empno,ename,job,sal,comm,deptno
7369,smith,clerk,800,,20
7499,allen,salesman,1600,300,30
7521,ward,salesman,1250,500,30
7566,jones,manager,2975,,20
7654,martin,salesman,1250,1400,30
7698,blake,manager,2850,,30
7782,clark,manager,2450,,10
7788,scott,analyst,3000,,20
7839,king,president,5000,,10
7844,turner,salesman,1500, 0,30
7876,adams,clerk,1100,,20
```

```
{ empno : '7369',   ename : 'smith', 'job': 'clerk', 'sa
{ empno : '7499',   ename : 'allen', 'job': 'salesman',
{ empno : '7521',   ename : 'ward', 'job': 'salesman', '
{ empno : '7566',   ename : 'jones', 'job': 'manager', '
{ empno : '7654',   ename : 'martin', 'job': 'salesman',
{ empno : '7698',   ename : 'blake', 'job': 'manager', '
{ empno : '7782',   ename : 'clark', 'job': 'manager', '
{ empno : '7788',   ename : 'scott', 'job': 'analyst', '
{ empno : '7839',   ename : 'king', 'job': 'president',
{ empno : '7844',   ename : 'turner', 'job': 'salesman',
{ empno : '7876',   ename : 'adams', 'job': 'clerk', 'sa
```

## ■ CSV

### ● csv.DictReader()

- header가 없는 경우

```
import csv
file = open('emp_no_header.csv')
reader = csv.DictReader(file)
for row in reader:
    print(row)
file.close()
```

```
7369,smith,clerk,800,,20
7499,allen,salesman,1600,300,30
7521,ward,salesman,1250,500,30
7566,jones,manager,2975,,20
7654,martin,salesman,1250,1400,30
7698,blake,manager,2850,,30
7782,clark,manager,2450,,10
7788,scott,analyst,3000,,20
7839,king,president,5000,,10
7844,turner,salesman,1500, 0,30
7876,adams,clerk,1100,,20
7900,james,clerk,950,,30
```

```
{ 7369 : '7499', smith : 'allen', 'clerk': 'salesman',
{ 7369 : '7521', smith : 'ward', 'clerk': 'salesman',
{ 7369 : '7566', smith : 'jones', 'clerk': 'manager',
{ 7369 : '7654', smith : 'martin', 'clerk': 'salesman',
{ 7369 : '7698', smith : 'blake', 'clerk': 'manager',
{ 7369 : '7782', smith : 'clark', 'clerk': 'manager',
{ 7369 : '7788', smith : 'scott', 'clerk': 'analyst',
{ 7369 : '7839', smith : 'king', 'clerk': 'president',
{ 7369 : '7844', smith : 'turner', 'clerk': 'salesman',
{ 7369 : '7876', smith : 'adams', 'clerk': 'clerk', '8
{ 7369 : '7900', smith : 'james', 'clerk': 'clerk', '8
```



## ■ CSV

### ● csv.DictReader()

- header 전달

```
import csv
file = open('emp_no_header.csv')
reader = csv.DictReader(
    file, ['empno', 'ename', 'job', 'sal', 'comm', 'deptno'])
for row in reader:
    print(row)
file.close()
```

```
7369,smith,clerk,800,,20
7499,allen,salesman,1600,300,30
7521,ward,salesman,1250,500,30
7566,jones,manager,2975,,20
7654,martin,salesman,1250,1400,30
7698,blake,manager,2850,,30
7782,clark,manager,2450,,10
7788,scott,analyst,3000,,20
7839,king,president,5000,,10
7844,turner,salesman,1500, 0,30
7876,adams,clerk,1100,,20
7900,james,clerk,950,,30
```

```
{ empno : '7369',  ename : 'smith', 'job': 'clerk', 'sa
{ empno : '7499',  ename : 'allen', 'job': 'salesman',
{ empno : '7521',  ename : 'ward', 'job': 'salesman', '
{ empno : '7566',  ename : 'jones', 'job': 'manager', '
{ empno : '7654',  ename : 'martin', 'job': 'salesman',
{ empno : '7698',  ename : 'blake', 'job': 'manager', '
{ empno : '7782',  ename : 'clark', 'job': 'manager', '
{ empno : '7788',  ename : 'scott', 'job': 'analyst', '
{ empno : '7839',  ename : 'king', 'job': 'president',
{ empno : '7844',  ename : 'turner', 'job': 'salesman',
{ empno : '7876',  ename : 'adams', 'job': 'clerk', 'sa
```

## ■ 연습문제

### ● "대검찰청\_범죄자\_직업\_20171231.csv" 내용 중

"범죄분류"가 "공인증개사법"과 "병역법" 인 데이터 현황 확인하기

```
import csv
file = open('emp_no_header.csv')
reader = csv.DictReader(
    file, ['empno', 'ename', 'job', 'sal', 'comm', 'deptno'])
for row in reader:
    print(row)
file.close()
```

#### 대검찰청\_범죄자\_직업\_20171231.csv

범죄분류, 농임수산업, 광업, 제조업, 건설업, 도소  
절도, 1528, 1, 184, 949, 187, 49, 373, 55,  
장물, 5, 0, 4, 12, 90, 11, 10, 1, 3, 11, 6, 0,  
사기, 2552, 37, 1334, 9480, 1038, 553, 12  
횡령, 688, 2, 253, 963, 163, 76, 178, 44, 4  
배임, 207, 0, 95, 390, 46, 9, 36, 5, 4, 28, 1  
손괴, 1149, 2, 131, 1090, 99, 31, 256, 31,  
살인, 19, 0, 1, 9, 2, 1, 5, 1, 0, 0, 5, 0, 1, 2,  
강도, 10, 0, 1, 8, 6, 5, 7, 0, 7, 0, 3, 0, 2, 2,  
방화, 32, 0, 3, 13, 1, 0, 7, 0, 3, 0, 1, 1, 2, 3  
성폭력, 401, 0, 115, 340, 70, 35, 215, 33, 1  
폭행, 3393, 6, 771, 4865, 569, 236, 2050,

{'범죄분류': '공인증개사법', '농임수산업': '33', '광업': '0', '제조업': '1', '건설업': '28',  
'도소매업': '3', '무역업': '0', '요식업': '4', '숙박업': '0', '유흥업': '0', '금융업': '1',  
'부동산업': '607', '의료보건업': '1', '차량정비업': '0', '노점': '0', '행사': '0', '기타사  
업': '160', '교원(사람)': '0', '사무원': '6', '기술자': '0', '점원': '0', '공원': '0', '운전  
자': '2', '경비원': '3', '외판원': '9', '국공영기업체직원': '0', '일반회사원': '56', '금융기  
관직원': '0', '유흥업종사자': '0', '요식업종사자': '0', '일용노동자': '13', '기타피고용자':  
'134', '의사': '0', '변호사': '1', '교수': '0', '종교가': '3', '언론인': '1', '예술인': '1',  
'기타전문직': '43', '공무원': '0', '군인': '0', '학생': '1', '주부': '23', '전경\_의경': '0',  
'공익요원': '0', '무직자': '215', '미상': '237'}

{'범죄분류': '병역법', '농임수산업': '22', '광업': '0', '제조업': '17', '건설업': '48', '도소  
매업': '17', '무역업': '2', '요식업': '51', '숙박업': '2', '유흥업': '28', '금융업': '2', '부  
동산업': '11', '의료보건업': '1', '차량정비업': '10', '노점': '0', '행사': '0', '기타사업':  
'428', '교원(사람)': '2', '사무원': '14', '기술자': '66', '점원': '62', '공원': '20', '운전  
자': '27', '경비원': '10', '외판원': '55', '국공영기업체직원': '0', '일반회사원': '1435', '금  
융기관직원': '3', '유흥업종사자': '41', '요식업종사자': '126', '일용노동자': '295', '기타피고  
용자': '1083', '의사': '8', '변호사': '3', '교수': '0', '종교가': '6', '언론인': '1', '예술  
인': '42', '기타전문직': '177', '공무원': '4', '군인': '0', '학생': '237', '주부': '41', '전  
경\_의경': '0', '공익요원': '129', '무직자': '1694', '미상': '841'}

## ■ CSV

### ● csv.DictWriter()

- Dictionary 형태로 내용 쓰기

```
import csv
file = open('output_dict.csv', 'w', newline='')
writer = csv.DictWriter(file, ['name', 'age', 'job'])
writer.writeheader()
writer.writerow({'name': 'John', 'age': 20, 'job': 'student'})
writer.writerow({'name': 'Jenny', 'age': 30, 'job': 'developer'})
writer.writerow({'name': 'Nate', 'age': 30, 'job': 'teacher'})
writer.writerow({'name': 'Julia', 'age': 40, 'job': 'dentist'})
file.close()
```

## ■ json

- 경량 텍스트 기반의 구조
- 읽기 쉬운 key : value 형식

```
{  
  "key1": "value1",  
  "key2": 100,  
}
```

```
[  
  'a', 'b', 'c'  
]
```

```
{  
  "key1": "value1",  
  "key2": 100,  
  "key3": {  
    "inner_key1": [1, 2, 3],  
    "inner_key2": "all"  
  }  
}
```

```
[  
  {  
    "id": 2100,  
    "address": "경기 용인시 처인구 포곡읍 포곡로 127",  
    "name": "대림공구열쇠",  
    "kind": "기타-기계공구"  
  },  
  {  
    "id": 3310,  
    "address": "경기 용인시 처인구 금령로",  
    "name": "고성방가노래연습장",  
    "kind": "레저업소-노래방"  
  },  
  {  
    "id": 20131,  
    "address": "경기 용인시 기흥구 이현로",  
    "name": "까페디플레이트",  
    "kind": "일반휴게음식-서양음식"  
  }  
]
```

## ■ json

### ● json.loads()

- JSON 문자열을 파이썬의 객체(Dictionary 또는 List)로 변환

```
import json
json_string = '''
{
    "age": 10,
    "name": "GGoReb"
}
'''

result = json.loads(json_string)
print(type(result))
print(result)
```

```
<class 'dict'>
{'age': 10, 'name': 'GGoReb'}
```

## ■ json

### ● json.loads()

- JSON 문자열을 파이썬의 객체(Dictionary 또는 List)로 변환

```
import json
json_string = '''
[
    { "age": 10, "name": "A" },
    { "age": 11, "name": "B" },
    { "age": 12, "name": "C" }
]
'''

result = json.loads(json_string)
print(type(result))
print(result)
```

```
<class 'list'>
```

```
[{'age': 10, 'name': 'A'}, {'age': 11, 'name': 'B'}, {'age': 12, 'name': 'C'}]
```

## ■ json

### ● json.load()

- JSON 형식의 파일 데이터를 파이썬의 객체(Dictionary 또는 List)로 변환

```
import json
with open('data1.json') as f:
    result = json.load(f)
print(type(result))
print(result)
```

```
<class 'dict'>
{'age': 10, 'name': 'GGoReb'}
```

```
import json
with open('data2.json') as f:
    result = json.load(f)
print(type(result))
print(result)
```

```
<class 'list'>
[{'age': 10, 'name': 'A'}, {'age': 11, 'name': 'B'}, {'age': 12, 'name': 'C'}]
```

## ■ json

### ● json.dumps()

- 파이썬의 객체(Dictionary 또는 List)를 JSON 문자열로 변환

```
import json
data = {
    "age": 10,
    "name": "GGoReb"
}
result = json.dumps(data)
print(type(result))
print(result)
```

```
<class 'str'>
{"age": 10, "name": "GGoReb"}
```



## ■ json

### ● json.dumps()

- 파이썬의 객체(Dictionary 또는 List)를 JSON 문자열로 변환

```
import json
data = [
    { "age": 10, "name": "A" },
    { "age": 11, "name": "B" },
    { "age": 12, "name": "C" }
]
result = json.dumps(data)
print(type(result))
print(result)
```

```
<class 'str'>
```

```
[{"age": 10, "name": "A"}, {"age": 11, "name": "B"}, {"age": 12, "name": "C"}]
```

## ■ json

### ● json.dump()

- 파이썬의 객체(Dictionary 또는 List)를 JSON 형식의 파일 데이터로 변환

```
import json
data = {
    "age": 10,
    "name": "GGoReb"
}
with open('mydata1.json', 'w') as f:
    json.dump(data, f)
```

```
import json
data = [
    { "age": 10, "name": "A" },
    { "age": 11, "name": "B" },
    { "age": 12, "name": "C" }
]
with open('mydata2.json', 'w') as f:
    json.dump(data, f)
```