

파일처리

파이썬은 파일과 관련된 표준함수를 제공합니다. 크게 텍스트 파일과 바이너리 파일이 그 것인데, 우선 텍스트 파일에 대해서 알아봅시다.

파이썬의 문자열을 파일로 쓸 때 문자열은 바이트로 변경되는데, 이때 인코딩 방식에 따라 다른 바이트 결과를 가질 수 있고 파일을 읽어 들일 때 인코딩 방식에 따라 다른 문자열로 읽혀 들일 수 있습니다. 따라서, 파일을 오픈할 때 encoding 파라미터를 지정해서 어떤 인코딩 방식을 사용하는지 명시적으로 지정하는 것이 좋겠습니다.

1. 파일 열기

파일을 열 때는 `open()` 함수를 사용하합니다.

파일 객체 = `open(문자열: 파일 경로, 문자열: 읽기모드)`

Aa	모드	설명
w(x)		write(새로 쓰기모드)
a		append(추가 이어쓰기 모드)
r		read(읽기 모드)

* 바이너리 파일을 읽기 위해서는 파일모드를 `rb` 로, 쓰기 위해서는 `wb` 로 지정합니다.
파일을 처리를 위해 흔히 사용되는 메서드들로서, `read(n)` 매소드는 `n` 바이트만큼의 데이터를 읽어 들이고, `ftell()`은 현재의 파일 위치를 리턴하고, `fseek()`는 특정 위치로 파일 포인터를 옮기는 일을 한다. (주: `read(n)` 메서드가 텍스트 파일에 사용되면 `n` 개의 (바이트가 아닌) 문자를 읽어 들인다. 만약 `n` 을 생략하면 모든 데이터를 읽어 들인다.)

2. 파일 닫기

파일을 닫을 때는 `close()` 함수를 사용합니다.

```
# 기본 인코딩 확인하기
import sys
encode = sys.getdefaultencoding()
print(encode)

file = open('basic.txt', 'w')
```

```
file.write('hello python3')
file.close()

f = open('./some_file.txt', 'w')
for i in range(1, 11):
    data = '%d번째 줄입니다 \n' % i
    f.write(data)
f.close()
```

* open() 함수로 파일을 열고 사용시, close()로 파일을 닫아주어야 합니다. 프로그램이 종료될때 열려있는 파일을 모두 닫고 종료됩니다만, close()로 닫아주는것이 좋습니다.

3. with 키워드

프로그램이 길어지면 open()과 close() 사이의 많은 코드가 들어가므로, 실수할 수 있습니다. 이 실수를 방지하고자 하는 키워드가 with 입니다.

with open('문자열: 파일경로, 문자열: 모드) as 파일객체:

실행할 문장

```
with open('advanced.txt', 'w') as f:
    f.write('hello python3 world')
```

4. 텍스트 읽기 - read()

파일에 텍스트를 쓸때는 write() 함수를 사용하고, 읽을때는 read()를 사용합니다.

```
with open('basic.txt', 'r') as f:
    contents = f.read()
    # line = f.readline()
    # lines = f.readlines() / list type
    print(contents)

#sys 모듈로 라인 출력하기
import sys
sys.stdout.writelines(line[1:2])
```

텍스트를 사용해서 데이터를 구조적으로 표현하는 방법은 CSV, XML, JSON을 사용하는 방법이 있는데, 이중에서 CSV에 대해서 살펴보겠습니다.

CSV(Comma Separated Value) - 콤마(쉼표)로 구분된 객체를 뜻합니다.

이름, 키, 몸무게

홍길동, 170, 72

김철수, 180, 74

....계속..

CSV 파일은 한 줄에 하나의 데이터를 나타내며, 각각의 줄은 쉼표를 사용해 데이터를 구분하고 이 때 첫번째 줄에 헤더(header)를 넣어 각각 데이터의 의미를 설명합니다. 기본 데이터 1000개 정도를 자동으로 입력해보겠습니다.

```
import random
hanguls = list('가나다라마바사아자차카타파하')
with open('info.txt', 'w') as f:
    for i in range(1000):
        name = random.choice(hanguls) + random.choice(hanguls)
        weight = random.randrange(50, 100)
        height = random.randrange(140, 200)
        #데이터 쓰기
        f.write('{}, {}, {}\n'.format(name, weight, height))
```

info.txt가 생성되면 한 줄씩 반복문을 통해서 읽어 들여봅니다.

```
...
for 한 줄을 나타내는 문자 in 파일 객체:
    실행할 코드
...
with open('info.txt', 'r', encoding='utf-8') as f:
    for line in f:
        (name, weight, height) = line.strip().split(',')

        if not name or not weight or not height:
            continue
        bmi = int(weight) / ((int(height) / 100) ** 2)
        result = ''
        if bmi >= 25:
            result = '과체중'
        elif 18.5 <= bmi:
            result = '정상 체중'
        else:
            result = '저체중'

        print('\n'.join([
```

```
'이름 : {}',
'몸무게: {}',
'키:{}',
'BMI:{}',
'결과:{}'
]).format(name, weight, height, bmi, result))

'''

이름 : 아나
몸무게: 79
키: 168
BMI:27.99036281179139
결과:과체중
이름 : 가사
몸무게: 88
키: 183
BMI:26.27728507868255
결과:과체중
'''
```

5. CSV 포맷 파일 불러오기

```
# csv (comma separate value)
# csv 모듈
```