

# 기초 PYTHON 프로그래밍

## 19. 파일 입출력

1. 파일 연결 - open() 내장 함수
2. 파일 입력 - 파일로부터 데이터 읽어오기
3. 파일 출력 - 파일에 데이터 저장하기

# 1. 파일 연결

## ◆ open() 내장 함수

파일객체 = open( file, mode )

file

파일명

mode

파일을 열 때의 모드를 의미하며, 다음 문자열 조합으로 사용함.

- r : 읽기 모드 (디폴트)
- w : 쓰기 모드
- a : 쓰기 + 이어쓰기 모드 (append)
- r+ : 읽기와 쓰기를 모두 하고자 할 때

# 1. 파일 연결

## ◆ file open 후에 사용할 수 있는 메소드

읽기	read()	read() - 파일 내용을 모두 읽어서 문자열(str)로 반환
		read(n) - 파일에서 n 바이트 읽어서 문자열(str)로 반환
	readline()	한 줄씩 읽어서 문자열(str)로 반환한다.
	readlines()	파일 전체를 리스트(list)로 반환한다.
쓰기	write()	문자열을 파일에 저장한다.
	writelines()	문자열 리스트를 파일에 저장한다.

## 2. 파일 입력

### ◆ read()와 read(n)

```
f = open('newfile.txt', 'r')
```

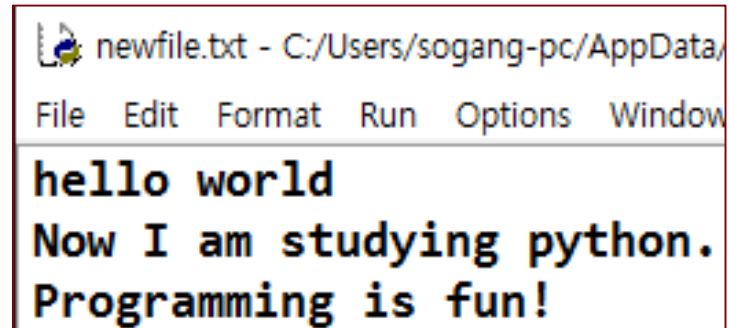
```
a = f.read(4)
```

```
print(a)
```

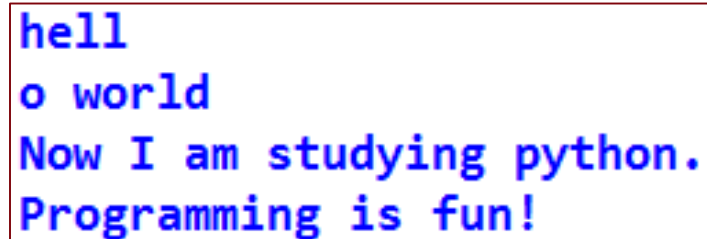
```
b = f.read()
```

```
print(b)
```

```
f.close()
```



newfile.txt - C:/Users/sogang-pc/AppData/  
File Edit Format Run Options Window  
hello world  
Now I am studying python.  
Programming is fun!



hell  
o world  
Now I am studying python.  
Programming is fun!

## 2. 파일 입력

### ◆ 한 줄씩 읽어 오기 - readline()

```
f = open('about.txt', 'r')
```

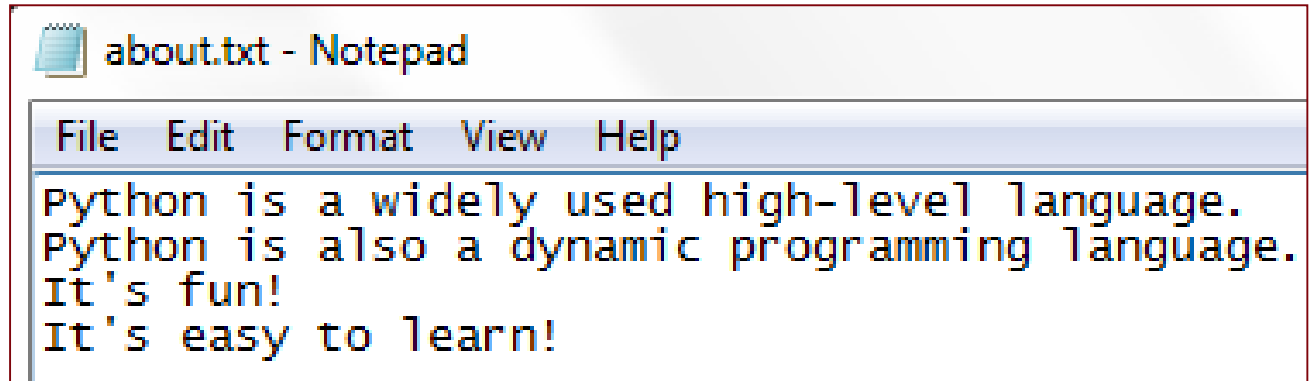
```
a = f.readline()
```

```
print(a)
```

```
b = f.readline()
```

```
print(b)
```

```
f.close()
```



```
about.txt - Notepad
File Edit Format View Help
Python is a widely used high-level language.
Python is also a dynamic programming language.
It's fun!
It's easy to learn!
```

```
Python is a widely used high-level language.

Python is also a dynamic programming language.

>>>
```

## 2. 파일 입력

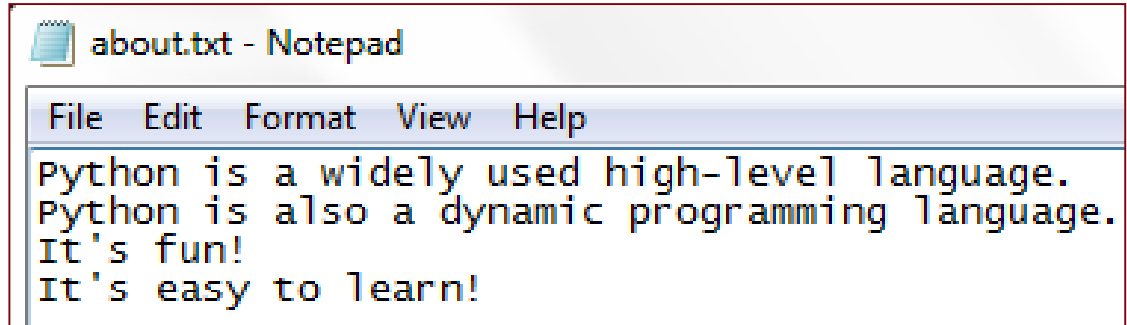
### ◆ 한 줄씩 읽어 오기 - for 구문 이용하기

```
f = open('about.txt', 'r')
```

```
for line in f:
```

```
    print(line)
```

```
f.close()
```



```
about.txt - Notepad
File Edit Format View Help
Python is a widely used high-level language.
Python is also a dynamic programming language.
It's fun!
It's easy to learn!
```

```
Python is a widely used high-level language.
```

```
Python is also a dynamic programming language.
```

```
It's fun!
```

```
It's easy to learn!
```

## 2. 파일 입력

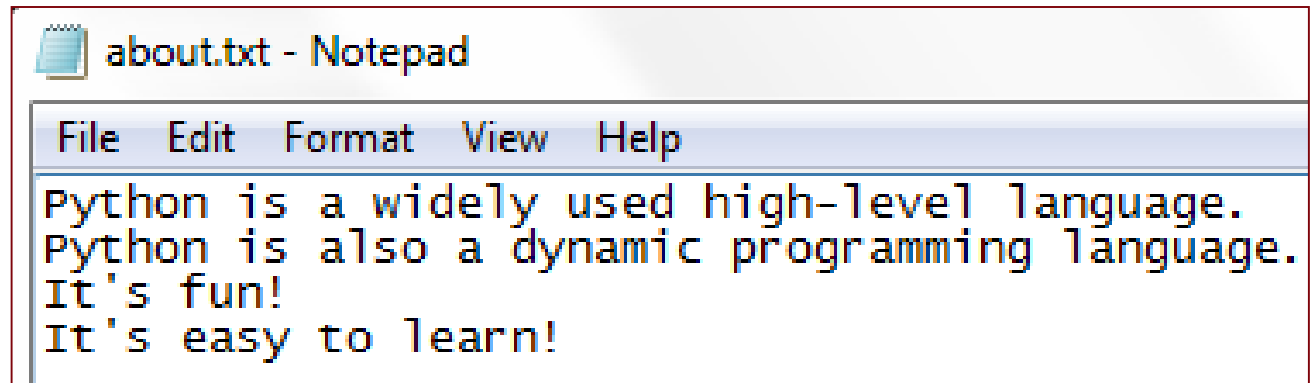
- ◆ readlines() - 파일 전체를 **리스트(list)**로 반환한다.

```
f = open('about.txt', 'r')
```

```
a = f.readlines()
```

```
print(a)
```

```
f.close()
```

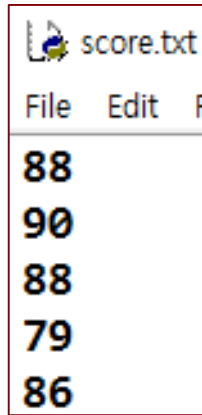


```
['Python is a widely used high-level language.\n', 'Python is also a dynamic programming language.\n', 'It's fun!\n', 'It's easy to learn!\n']
```

## 2. 파일 입력

### ◆ 파일 내용을 통째로 읽어 들여 한 줄씩 리스트에 저장 - 1

〈 입력 파일 〉 한 줄에 정수 한 개씩 저장되어 있음



〈 출력 결과 〉

**[88, 90, 88, 79, 86]**

```
f = open('score.txt', 'r')
score = []
for line in f:
    score.append(int(line))

print(score)
f.close()
```

```
score = []

with open('score.txt') as f:
    for line in f:
        score.append(int(line))

print(score)
f.close()
```

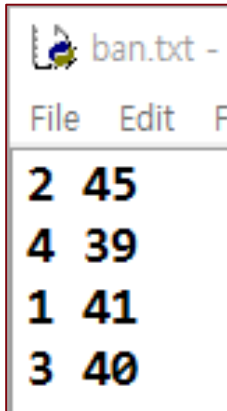
파일명



## 2. 파일 입력

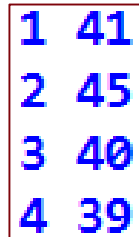
### ◆ 파일 내용을 통째로 읽어 들여 한 줄씩 사전에 저장 - 2

〈 입력 파일 〉 한 줄에 정수가 두 개씩 저장되어 있음 (사전으로 저장)



ban.txt -  
File Edit F  
2 45  
4 39  
1 41  
3 40

〈 출력 결과 〉



1 41  
2 45  
3 40  
4 39

```
D = {}  
with open('ban.txt') as f:  
    for line in f:  
        (key, val) = line.split()  
        D[int(key)] = val  
  
for key, val in D.items():  
    print(key, val)  
f.close()
```

## 2. 파일 입력

### ◆ 파일 내용을 통째로 읽어 들여 한 줄씩 사전에 저장 - 3

〈 입력 파일 〉 첫 번째 자료는 키로, 나머지 자료는 값으로 하여 사전에 저장하고자 함.

```
2 Alice Paul David Bob
4 Cindy Stella Bill
1 Henry Jenny Jessica Erin Tim
3 John Joe Tom
```

```
{'1': ['Henry', 'Jenny', 'Jessica', 'Erin', 'Tim'], '3': ['John', 'Joe', 'Tom'],
'2': ['Alice', 'Paul', 'David', 'Bob'], '4': ['Cindy', 'Stella', 'Bill']}
```

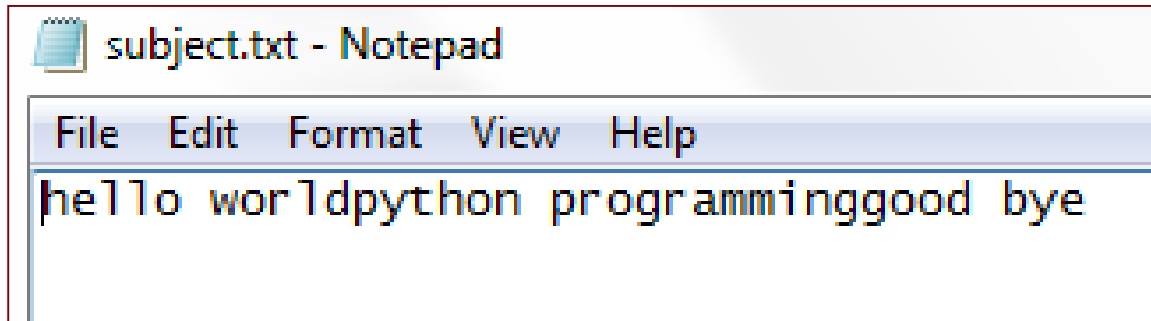
```
D = {}
```

```
f = open('ban_student.txt', 'r')
for line in f:
    items = line.split()
    key, values = items[0], items[1:]
    D[key] = values
print(D)
```

### 3. 파일 출력

#### ◆ write() 예제

```
f = open("subject.txt", "w")  
f.write('hello world')  
f.write('python programming')  
f.write('good bye')  
f.close()
```



### 3. 파일 출력

#### ◆ writelines() 예제

```
f = open("subject2.txt", "w")  
f.writelines(['hello\n', 'world\n', 'python\n',  
             'programming', 'good', 'bye'])  
f.close()
```

