

파이썬 프로그래밍

파일 입출력



한국기술교육대학교
온라인평생교육원

▣ 표준 출력 방향 전환

1. 표준 출력을 파일로 저장하기

- sys 모듈의 표준 입출력 관련 객체
 - sys.stdout: 표준 출력
 - sys.stderr: 표준 에러 출력
 - sys.stdin: 표준 입력
- 예를 들어, sys.stdout을 파일 객체로 변환하면 모든 표준 출력(print 출력)은 해당 파일로 저장된다.

```
import sys

f = open('t.txt', 'w')
stdout = sys.stdout # 표준 출력 저장해 두기
sys.stdout = f      # 파일 객체로 표준 출력 변경
print 'Sample output'
print 'Good'
print 'Good'
f.close()
sys.stdout = stdout # 필요하면 표준 출력 원상 복구
```

- stdout 변수는 모니터(표준 출력) 화면에 해당하는 레퍼런스 값 가짐
- print → 표준 출력 (stdout 쪽으로 내용 출력)
- print → 표준 출력 (stdout 쪽으로 내용 출력)
- w → 쓰기 모드
- sys.stdout에서 stdout 변수는 sys 내에 존재하는 변수
- sys.stdout이 화면이 아닌 파일쪽(t.txt) 출력이 됨
- print는 이제 sys.stdout로 내용 보내서 출력
- 파일쪽으로 출력이 되었던 부분을 다시 모니터로 복원

▣ 표준 출력 방향 전환

1. 표준 출력을 파일로 저장하기

```
f = open('t.txt')  
print f.read()
```

```
Sample output  
Good  
Good
```

- print를 직접 이용하여 출력을 다른 객체로 전환하기

```
print >> sys.stderr, "Warning, action field not supplied"
```

```
Warning, action field not supplied
```

- >> (부등호 2개) → 표준 출력이 아닌 다른 쪽으로 출력
- stderr → 표준 에러
- stdout, stderr → 이클립스 내에서는 console로 내용 출력
- stderr → 빨간색으로 내용 출력함
- stderr → 표준 에러 쪽에 reference를 가지고 있는 객체

▣ 표준 출력 방향 전환

1. 표준 출력을 파일로 저장하기

- 동일 방법으로 표준 출력(print)을 파일 객체로 전환

```
f = open('t.txt', 'w')  
print >> f, 'spam string'  
f.close()
```

```
f = open('t.txt')  
print f.read()  
f.close()
```

```
spam string
```

- >> : 리다이렉트 기호

▣ 표준 출력 방향 전환

2. StringIO 모듈 사용하기

- StringIO 모듈의 StringIO 클래스 객체
 - 파일 객체처럼 입출력 가능한 문자열 객체
 - StringIO에 지원되는 메소드는 파일 객체가 지원하는 메소드와 거의 동일하다.
 - getvalue() 메소드
 - * 현재까지 담아 놓은 전체 내용을 반환한다.

- 문자열을 자기가 받고 내보낼 수 있는 파일객체와 비슷한 객체

```
import StringIO

f = StringIO.StringIO()
f.write("abc")
f.seek(0)
s = f.read()
print s
print

s2 = f.getvalue()
print s2
```

```
abc
```

```
abc
```

- 하드 디스크에 존재하는 파일 X → 메모리에 존재하는 특정 영역
- StringIO 객체가 지니고 있는 영역
- abc를 썼으니까 파일포인터의 위치는 맨 뒤
- seek(0) → 파일포인터 위치를 앞으로 당김
- getvalue() → 현재 지니고 있는 전체 내용 반환

▣ 표준 출력 방향 전환

2. StringIO 모듈 사용하기

- 표준 출력으로 문자열 객체에 내용 작성하기

```
import sys
import StringIO

stdout = sys.stdout # 표준 출력 저장해 두기
sys.stdout = f = StringIO.StringIO()

print type(f)
print 'Sample output'
print 'Good'
print 'Good'

sys.stdout = stdout
```

- 기존 표준출력 → f (파일이 아니라 StringIO 객체)

▣ 표준 출력 방향 전환

2. StringIO 모듈 사용하기

```
s = f.getvalue()

print 'Done-----'
print s
```

```
Done-----
<type 'instance'>
Sample output
Good
Good
```

- `getvalue()` → 현재 지니고 있는 전체 내용 반환