

 kim hanwoong 테스트 코드 작성하기.

aa93fd6 25 days ago

0 contributors

75 lines (58 sloc) | 2.3 KB

# Standard Streams

리눅스에서는 명령어를 실행하는 순간 해당 명령어에 대해서 입출력 통로가 생성됩니다. 총 3개의 통로가 생깁니다. 이 통로를 우리는 Standard Streams 라고 부릅니다. 각 3개의 통로 이름은 아래와 같습니다.

- STDIN : 0, 입력
- STDOUT : 1, 출력
- STDERR : 2, 오류

이 장에서 제가 이야기 하고 싶은 주제와 목적은 하나입니다.!!

"명령어를 만들 때 Standard Streams를 지원하도록 프로그래밍 하자!"

입니다.

기본 출력은 STDOUT, 에러는 STDERR로 출력되도록 코드를 작성하면 좋습니다.

Python에서 STDIN, STDOUT, STDERR를 처리하는 코드를 작성하는 방법을 알아보겠습니다.

## stdout

Python에서 일반적인 출력은 아래 코드 형태를 가지게 됩니다.

```
print("hello stdout")

# 또는

import sys
sys.stdout.write("hello stdout" + '\n')
```

## stderr

자신의 코드에서 에러를 처리할 때는 아래처럼 에러내용을 출력하도록 해주세요. 그렇게 되면 에러 내용은 STDERR로 전달됩니다.

```
# stderr_test.py
sys.stderr.write("hello stderr" + '\n')
```

이렇게 작성하면 표준에러(STDERR)만 모아서 로그파일로 남길 수 있습니다. 일반적으로 로그는 쌓여야 하기 때문에 2>> 리다이렉션을 사용했습니다.

```
$ python stderr_test.py 2>> error.log
```

## stdin

stdin의 값이 있다면 stdin 값을 불러오는 Python 예제입니다.

```
# stdin_test.py
import sys
if not sys.stdin.isatty():
    for line in sys.stdin:
        sys.stdout.write(line)
```

위 코드를 저장후 터미널에서 아래처럼 테스트 해보세요.

```
$ echo "test" | python stdin_test.py
```

아래 내용으로 저장된 shots.txt 파일이 존재할 때 에도 stdin으로 데이터를 받을 수 있습니다.

shots.txt

```
F00_0010
F00_0020
BAR_0010
BAR_0020
```

```
$ cat shots.txt | python stdin_test.py
```

코드를 짤 때 Starndard Streams를 지원하도록 코드를 작성하면, 수많은 리눅스 명령어와 연동할 수 있게됩니다. 거대 한 소프트웨어를 짜지 말고 리눅스 명령어와 어울리는 간단하지만 강력한 명령어를 만드는 것이 유지보수 및 강력한 콘텐츠 파이프라인을 만드는 데 도움이 됩니다.