Python sys 모듈 정리 노트

KUCSEPotato

January 17, 2026

Contents

1 sys 모듈이란

`sys` 모듈은 **파이썬 인터프리터(Python Interpreter)와 직접 연결된 모듈**이다.

일반적인 라이브러리들이 특정 기능을 제공하는 것과 달리, `sys`는 **프로그램이 실행되는 환경 자체에 접근하기 위한 기능**을 제공한다.

쉽게 말하면,

> **sys는 파이썬 코드와 운영체제 사이를 연결하는 통로이다.**

입력, 출력, 프로그램 종료, 실행 인자, 경로 정보 등은 모두 `sys` 모듈을 통해 제어할 수 있다.

2 왜 sys 모듈이 필요한가

Python에서 가장 기본적인 입출력 함수는 다음과 같다.

- 입력: `input()`
- 출력: `print()`

이 함수들은 매우 편리하지만, 다음과 같은 한계를 가진다.

- 대량 입력 처리 시 속도가 느리다
- 출력 형식을 세밀하게 제어하기 어렵다
- 프로그램 실행 환경 정보를 알 수 없다

이러한 문제를 해결하기 위해 Python은 `sys` 모듈을 통해 **저수준(low-level) 접근 방법**을 제공한다.

3 표준 스트림(Standard Stream)

운영체제 수준에서 모든 프로그램은 다음 세 가지 기본 스트림을 가진다.

- 표준 입력 (Standard Input)
- 표준 출력 (Standard Output)
- 표준 에러 (Standard Error)

Python에서는 이를 다음과 같이 제공한다.

스트림	객체
표준 입력	`sys.stdin`
표준 출력	`sys.stdout`
표준 에러	`sys.stderr`

4 sys.stdin

`sys.stdin`은 **표준 입력 스트림 객체**이다.

기본적으로 키보드 입력을 의미하며, 파일처럼 동작하는 객체이다.

```
import sys

line = sys.stdin.readline()
```

4.1 readline()의 특징

- 한 줄을 문자열로 읽어온다
- 문자열 끝에 개행 문자(\n)가 포함된다
- input()보다 빠르다

input() 함수는 내부적으로 sys.stdin.readline()을 감싼(wrapper) 함수이다. 따라서 속도 면에서는 sys.stdin.readline()이 더 빠르다.

4.2 빠른 입력 패턴

코딩 테스트나 대량 입력 상황에서는 다음과 같은 패턴이 거의 표준처럼 사용된다.

```
import sys
input = sys.stdin.readline
```

이후 모든 입력을 빠른 방식으로 처리할 수 있다.

5 sys.stdout

sys.stdout은 표준 출력 스트림을 의미한다.

```
import sys
sys.stdout.write("Hello")
```

5.1 print와의 차이점

- 자동 개행이 없다
- 공백 구분이 없다
- 문자열만 출력 가능하다

따라서 출력 형식을 직접 구성해야 한다.

```
sys.stdout.write(" ".join(map(str, result)) + "\n")
```

6 sys.stderr

sys.stderr는 에러 메시지 출력을 위한 스트림이다.

```
import sys
print("error occurred", file=sys.stderr)
```

표준 출력과 분리되어 있기 때문에 디버깅 로그 출력이나 오류 메시지에 자주 사용된다.

7 프로그램 종료: sys.exit

sys.exit()는 프로그램을 즉시 종료시킨다.

```
import sys

if error:
    sys.exit(1)
```

내부적으로는 SystemExit 예외를 발생시키는 방식으로 동작한다.

- 0: 정상 종료
- 1 이상: 비정상 종료

8 실행 인자: sys.argv

sys.argv는 프로그램 실행 시 전달된 인자들의 목록이다.

```
# python main.py 10 20
import sys
print(sys.argv)
```

출력 결과는 다음과 같다.

['main.py', '10', '20']

- argv[0]: 실행 파일 이름
- argv[1:]: 전달된 인자

9 실행 환경 정보

- sys.version : 파이썬 버전
- sys.platform : 운영체제 정보
- sys.path : 모듈 탐색 경로

이 정보들은 환경 의존적인 프로그램을 작성할 때 유용하다.

10 sys 모듈을 사용하는 대표적인 상황

- 코딩 테스트에서 대량 입력 처리
- 출력량이 매우 많은 문제
- 커맨드라인 도구 개발
- 디버깅 및 로그 분리
- 시스템 스크립트 작성

11 정리

- sys는 파이썬 실행 환경에 직접 접근하는 모듈이다.
- 빠른 입출력 처리를 위해 반드시 익혀야 한다.
- input()과 print()는 내부적으로 sys를 사용한다.
- 고성능 파이썬 코드의 시작은 sys 이해이다.