Chapter13: CSV와 로그관리

Comma Separate Value

- CSV, 필드를 쉼표(,)로 구분한 텍스트 파일

- 엑셀 양식의 데이터를 프로그램에 상관없이 쓰기 위한 데이터 형식

- 통칭 **character-separated values(CSV)**라 부름

CSV객체로 CSV 처리

-Text파일 형태로 데이터 처리시 문장 내에 들어가 있는 **","** 등에 대해 전처리 과정이 필요

-파이썬에서는 간단히 CSV파일을 처리하기 위해 **csv 객체**를 제공함

```
import csv
reader = csv.reader(f,
         delimiter=',', quotechar='"',
         quoting=csv.QUOTE_ALL)
```

Attribute	Default	Meaning
delimiter	,	글자를 나누는 기준
lineterminator	₩r₩n	줄 바꿈 기준
quotechar	"	문자열을 둘러싸는 신호 문자
quoting	QUOTE_MINIMAL	데이터 나누는 기준이 quotechar에 의해 둘러싸인 레벨

Logging – 로그남기기

-프로그램이 실행되는 동안 일어나는 정보를 기록에 남기기

-Console 화면에 출력(print), 파일에 남기기(저장), DB에 남기기(database)

-기록된 로그를 분석하여 의미 있는 결과를 도출 할 수 있음

*** 기록을 print로 남기는 것도 가능하다, 그러나 console 창에 남기는 기록은 분석 시 사용불가

Logging level

-프로그램 진행 상황에 따라 다른 Level의 Log를 출력함

ex) debug > info > warning > error > critiacal

Configparser & Argparser

-실제 프로그램을 실행할 땐 여러 설정이 필요(데이터 파일 위치, 파일 저장 장소, Operation Type)

- 정보를 설정해주는 방법 Configparser (파일의 설정 값을 저장), Argparser (실행시점에 설정 값을 저장)

Configparser

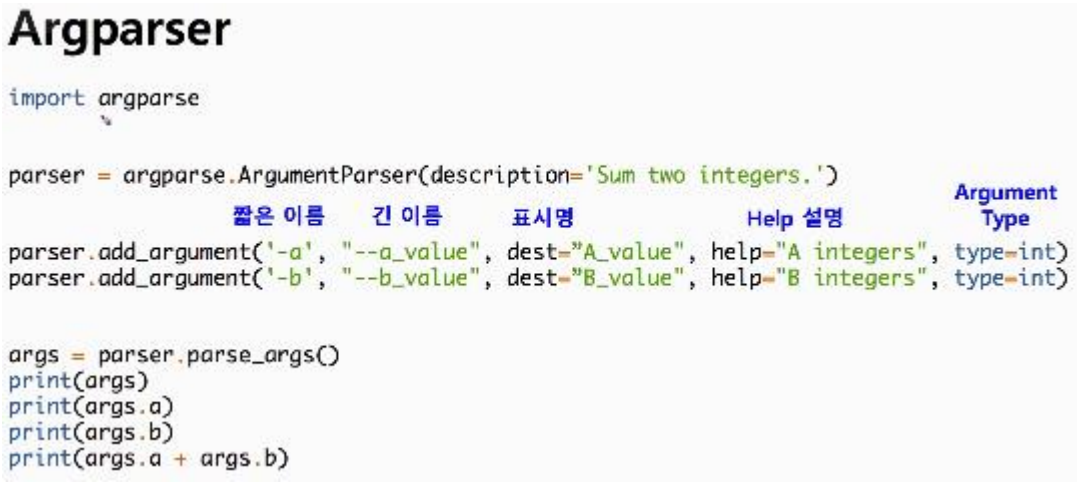
-프로그램의 실행 설정을 file에 저장함, Section, key, Value 값의 형태로 설정된 파일 사용

-설정파일은 dict type으로 설정

Argparser

-Console 창에서 프로그램 실행 시 Setting 정보를 저장함

-**Command-Line Option** 이라고 부름



Logging formatter

-Log의 결과값의 format을 지정해줄 수 있음