Ch3. 파일과 예외: 에러 다루기

- summarized by 전승일(2014/9/27)
- source code: http://www.headfirstlabs.com/books/hfpython/
- support site: http://python.itcarlow.ie

데이터는 프로그램 바깥에 있습니다

- 일반적인 프로그램 모델

```
입력 -> 처리 -> 출력
```

- 주요 입력 소스

```
DB
File
Internet(URL)
Console입력
```

파일은 단지 여러 라인으로 구성된 텍스트

- 파일을 열고 한번에 한줄씩 읽어서 처리
- 실습하기: file input method

데이터 자세히 살펴보기

- sketch.txt 파일 확인
- split method

```
(role, line_spoken) = each_line.split(':')
```

- 실습하기: line split

데이터 분석

- 런타임 에러 발생

```
ValueError: too many values to unpack (expected 2)
```

method를 알아보기 위해 help를 사용하세요

- 실습하기: split method 인자 수정하기

```
help(str.split)

(role, line_spoken) = each_line.split(':', 1)
```

- 런타임 에러 발생

```
ValueError: need more than 1 value to unpack
```

데이터를 (더 잘) 파악하세요

- 콜론(:)이 없는 경우

전혀 다른 두 가지 방법

- split method를 호출할 수 있는지 미리 확인해서 프로그램 논리를 추가하는 방법 (방법1)
- 일단 에러가 발생하도록 놔둔 다음에 문제가 생기면 문제를 찾아서 해결하는 방법 (방법2)

프로그램 논리 추가 (방법1)

- 문자열에 콜론(:)이 있는지 먼저 확인하는 방법

```
help(str.find)
```

- 실습하기: line split with find
- 이 방법의 문제점?

```
데이터의 형식이 바뀌면 조건문도 바뀌어야 한다
또 다른 예외 조건이 발생하면 또다시 에러가 발생하므로 이 코드는 다소 취약하다
```

예외 처리하기 (방법2)

- 예외가 발생했을 때 프로그램이 예외를 무시하면 -> crash 발생
- 예외가 발생했을 때 적어도 프로그램이 crash 되지 않도록 예외를 잡을(catch) 수 있다 -> 프로그램이 가능한 한 꿋꿋하게 작동

먼저 실행하고, 나중에 복구하기

- try/except 메커니즘

```
try:
    # 런타임 에러를 발생시킬 수도 있는 코드
except:
    # 여러분이 정의한 에러 복구 코드
```

보호할 코드 알아내기

- split 코드 및 그 이하 세줄

```
split method 호출에서 에러가 발생하면 그 이하 세줄도 실행하지 않아야 하므로
```

에러가 발생하면 지나가세요

- 아무런 수행도 하지 않는데 코드를 넣어야 하는 경우

```
help("pass")
```

- 실습하기: try
- 난롯가 담소: 131쪽

또 다른 에러는 어떻게 해야 하나요?

- 데이터 파일이 지워져 버린 예외 발생 시

```
방법1 코드와 방법2 코드에서 모두 예외 발생하고 crash 됨
```

- 실습하기: sketch.txt 파일명을 바꾸고 코드1과 코드2에서 모두 예외 발생하는지 확인하기

```
FileNotFoundError: [Errno 2] No such file or directory: 'sketch.txt'
```

에러 확인 코드 추가하기...

- 프로그램 논리 추가 방법

```
파일이 존재하는지 확인하는 코드 추가
help(os.path.exists)
```

- 실습하기: more error checking

... 또는 예외 처리에 한 단계 추가하기

- 예외 처리하기 방법 사용
- 실습하기: more exception handling

그러면 어떤 방법이 좋은가?

- 코드가 해야 할 일에 집중하세요

이제 다 됐습니다... 사소한 거 하나만 빼고요

- 가장 흔한 에러: IOError, ValueError
- 예외 처리는 너무 광범위하지 않도록 가능하면 특정 유형의 에러 범위로 제한할 필요가 있음

특정 예외만 처리하도록 하세요

- 실습 하기: exception handling

Additional Information

- Builtin Exception: Built-in Execeptions hierarchy

  ```
  IOError는 OSError와 같으며 하위 호환성을 위해 유지됨
  ```

- Another good IDE: PyCharm - JetBrains : Community Edition

  ```
  Install 방법: Download 탭에 가면 각 OS 별로 Installation Instruction 링크가 있음
  ```