라텍 최소 예제 작성법

Nicola L C Talbot

2014년 12월 4일

요 약

라텍 오류(errors)를 디버깅하기 위해 최소 예제(minimal example)를 작성해야 할 때가 있다. 특히 이것이 중요한 것은 버그 리포트를 하거나 도움을 요청할 때 문제가 되는 부분을 찾아낼 수 있도록 하기 위해서이다. 최소 예제를 작성하다 보면 스스로 문제가 무엇인지를 알게 되어 질문을 하여 남을 괴롭히거나 답변을 기다리느라 시간을 보내지 않고도 문제를 해결하게 되는 경우가 많다. 이 문서는 그 최소 예제를 작성하는 법을 적어보려 한다. Need More Help?를 보라.

이 문서의 홈페이지는 http://www.dickimaw-books.com/latex/minexample/이다. 문서의 소스 코드를 Zip 압축파일로 다운받을 수 있다.

한국어 번역은 TeX Live에 포함된 `dickimaw-minexample.pdf` (2014년 1월 17일 (버전 1.2))를 대본으로 하였고 번역자는 Nova De Hi이다. 번역 문서의 라이선스는 원본과 같다. 원문에는 마지막 절로 GNU 자유 문서 라이선스 전문이 포함되어 있으나 번역본에서는 생략하였다. 라이선스 전문을 보려면 원문을 참고하라.

차 례

1 들어가기

최소 예제란 문제점이 무엇인지를 보여주는 완전한 형식을 갖춘 최소한의 문서를 말한다. 최소 예제 파일은 문제점과 무관한 패키지나 코드를 포함하지 않아야 하지만 도큐멘트 클래스(`\documentclass`)와 도큐멘트 환경(`\begin{document}`…`\end{document}`)은 반드시 포함해야 한다.

최소 예제를 작성하는 데는 두 가지 접근방법이 있다. 하나는 처음부터 만드는 "상향식(building up)" 방법이고 다른 하나는 전체로부터 지워나가는 "하향식(hacking down)"이다. 이 글에서는 두 방법을 모두 보이겠다. 최소 예제를 작성하다 보면 문제를 바로 해결할 수도 있다. 그러나 여전히 문제가 지속된다면 이 최소 예제를 포스팅하여 도움을 청할 수 있다. (패키지 문서를 읽거나(패키지 문서를 찾아 읽는 방법 (section 6)) 뉴스그룹 아카이브, 포럼, 질문답변 사이트 등 인터넷을 먼저 찾아보는 것을 잊지 말자.)

(나를 포함해서) 많은 패키지 저자들은 The LaTeX Community, StackExchange, 뉴스그룹 comp.text.tex 같은 데서 메시지를 읽는다. 그러므로 자신이 해결할 수 없는 문제가 발생하였다면 이 사이트 중 하나에 문의하는 글을 게시하는 것(이 메시지에 최소 예제 파일을 첨부하여)이 좋다. 코드에 실수가 있다면 누군가 그것을 지적해줄 것이므로 패키지 저자에게 직접 요청하는 것보다 더 빨리 답변을 얻을 수도 있고 이러한 질문과 답변을 보고 배우는 사람도 있다. 그런데 명심할 것은 이와 같이 답변을 해주는 사람은 돈 받고 하는 일이 아니며 답변을 강요할 수도 없다는 점이다. 그러므로 자신의 요청이나 질문이 마치 권리행사 식의 요구나 비난처럼 보이지 않도록 주의해야 한다.

질문 글을 올릴 때 문제점을 간단히 기술하여야 한다. 그리고 문제를 해결하기 위해 자신이 시도해본 방법을 적는다. 자신이 작성하는 문서에 대하여 길고 장황하게 설명하지 말자. 그렇게 하는 것이 문제를 명확히하는데 별 도움이 안 되는 수가 많다. 너무 글이 길고 요구사항이 많으면 사람들이 읽지 않는다. 또한, 누군가 비슷한 문제를 겪고 질문답변이 이루어진 적이 없는지를 앞서 언급한 사이트에서 검색해보는 것이 좋다. 자주 묻는 질문을 뒤져보면 이미 비슷한 문제에 대한 간략한 답변을 찾을지도 모른다. 먼저 확인하자.

2 상향식 접근

먼저 문서를 다음과 같이 시작한다.

```
\documentclass{article}
\begin{document}
\end{document}
```

그런 다음에 문제가 발생할 때까지 여기에 내용을 추가해간다. `\chapter`가 있는 문서라면 `article` 대신 `book`이나 `report`를 쓰자. 이렇게 최소 예제 파일을 처음부터 작성하는 것을 '상향식 접근'이라고 한다.

예를 들어가며 설명하겠다. 문제가 발생한 문서가 다음과 같이 되어 있다고 하자.

```
\documentclass{myuniversityscustomclass}

\usepackage[french,USenglish]{babel}
\usepackage[mmddyyyy]{datetime}
\usepackage{nonstandardpackage}
\usepackage{anothernonstandardpackage}
% lots of other packages that may or may not be standard

% lots of your own definitions

\author{John Doe}
\title{Astounding Discoveries}

\begin{document}
\maketitle
\tableofcontents
\listoffigures
\listoftables

% 300 or so pages of text, graphics, tables, bibliography and
% sundry other stuff

\end{document}
```

문제가 되는 것은 표제지 상에 나타나는 날짜가 11/14/2008로 나타났으면 좋겠는데 여기서는 November 14, 2008로 나온다는 것이라고 하자. datetime 패키지를 로드하면서 `mmddyyyy` 옵션을 주었음을 이미 확인하였다. 뭐가 잘못된 걸까?

`\date` 명령을 사용하지 않았기 때문에 표제지의 날짜는 `\today` 명령으로 만들어질 것이다. 따라서 이 문제는 `\today` 명령의 정의와 관련된 것이 틀림없다. 이것은 datetime 패키지의 버그로 보인다. 어떻게 해야 하나? 하필 이 패키지의 저자가 나(필자)인데, 만약 이 문제를 발견한 사람이 나에게 자신의 300쪽이 넘는 모든 문서와 수많은 그림 파일 그리고 엄청난 문헌목록을 내게 보낸다면? 곤란하다. 나의 메일 수신함의 용량 문제는 차치하고라도 내게 `youruniversityscustomclass.cls`가 있을 리가 없고 비표준 패키지들도 설치되어 있지 않기 때문에 이 문서를 테스트해볼 수가 없는 것이다. 그러면 나는

바로 최소 예제 파일을 보내달라고 하거나, 아니면 "나중에 봐야지" 하고서 잊어버리고 있다가 하루 이틀, 몇 주나 지나서 최소 예제 파일을 보내달라고 하게 될 것이다.[1]

이미 문제가 `\today` 명령에 있다는 것을 파악했으므로 최소 예제를 만들 필요가 있다. 이 명령을 재정의하는 datetime 패키지를 사용하고자 하므로 최소 예제에서 이 패키지를 로드해야 한다. 패키지 옵션은 원래 문서에서 지정된 것과 같이 한다.

```
\documentclass{article}
\usepackage[mmddyyyy]{datetime}
\begin{document}
\today
\end{document}
```

이 파일의 이름을 `test.tex`이라고 하고 LaTeX을 실행하여 결과를 확인해본다. 그런데 결과를 보니 문제가 없다. 따라서 문제를 일으킨 것은 datetime이 아니라 함께 로드한 다른 패키지 때문일 것이다. 원래 문서에서 로드했던 패키지들을 하나씩 차례로 순서대로 넣어본다. 패키지를 추가했는데 결과에 영향이 없다면 그 패키지는 테스트 파일에서 제외한다. 그리고 그 다음 패키지를 시험한다. 예를 들어 문제가 된 문서가 babel 패키지를 로드하고 있으므로 테스트 파일에 이 패키지를 원래 문서에서와 동일한 옵션으로 추가하여 보자. 그러면 최소 예제 파일은 다음과 같이 된다.

```
\documentclass{article}
\usepackage[french,USenglish]{babel}
\usepackage[mmddyyyy]{datetime}
\begin{document}
\today
\end{document}
```

LaTeX을 실행하고 결과를 확인하자. 과연 November 14, 2008이라고 날짜 표기가 바뀌어 있다. 이제 이 파일은 오류를 재현하게 되었다. 수백 줄 수천 줄의 파일이 아니라 여섯 줄만으로 문제가 무엇인지 파악할 수 있게 된 것이다.

그 다음에 해야 할 일은? datetime 패키지 문서에서 babel 패키지를 언급하는 부분이 없는지 찾아보아야 한다. datetime 패키지 문서는 PDF나 HTML 형식으로 제공된다. PDF 뷰어나 HTML 뷰어에서 "babel"이라는 단어를 검색하여 찾아보자. 그러면 babel을 datetime보다 먼저 로드하라고 언급하는 문장을 찾을 수 있을 것이다. 테스트 파일에서 로딩 순서를 바꾸어서 테스트해본다.

[1] 사실 요새 나는 버그 리포트 폼(http://www.dickimaw-books.com/bug-report.html)으로 버그 리포트를 해달라고 하고 있다.

지금 문제삼은 것은 datetime 패키지의 FAQ (http://www.dickimaw-books.com/faqs/datetimefaq.html)가 있으므로 그곳을 찾아 보자. 이 FAQ는 내가 작성한 것 중 가장 널리 쓰이는 패키지를 망라하고 있다.[2] 목차에서 datetime 섹션을 찾아보면 "The date is in another language or in the wrong format" 이라는 항목을 발견할 수 있다. 클릭해서 답변을 읽어보자. 이 답변에서는 datetime 이전 버전에 babel 패키지와 함께 쓸 때 버그가 있었고 지금은 수정되었다고 하고 있다. 따라서 다음에 확인하여야 할 것은 사용 중인 패키지의 버전일 것이다. 테스트 파일에 `\listfiles` 명령을 추가하자.

```
\listfiles
\documentclass{article}
\usepackage[french,USenglish]{babel}
\usepackage[mmddyyyy]{datetime}
\begin{document}
\today
\end{document}
```

log 파일의 마지막에 로드 된 모든 파일의 목록이 버전 번호와 작성일과 함께 열거된다. datetime 패키지의 버전을 확인하자. 최신 버전인가? 만약 그렇지 않다면 최신 버전을 다운로드하여 다시 시도한다. 만약 최신 버전이라면 저자(이 경우에는 즉 나)에게 테스트 파일과 로그 파일을 보내자. 패키지 문서를 보면 패키지 저자에게 연락할 수 있는 방법이나 버그 리포팅 방법이 적혀 있다.

공개적으로 이용할 수 없는 패키지와 충돌하는 경우(예를 들면 "우리 대학 리포트 스타일"이나 비공개적으로 배포되는 패키지)라면 그 패키지도 함께 보내어야 저자가 검토할 수 있다. 공개 패키지이기는 하지만 CTAN에 등록되지 않은 것이라면 그것을 다운로드받을 수 있는 방법을 알려주어야 한다.

3 하향식 접근

앞선 상향식 접근 (section 2) 절에서 최소 예제 파일을 처음부터 작성하는 방법을 보였다. 이 절에서는 "지워가면서 만드는 방법"을 보이겠다. 앞서와 마찬가지로 300페이지에 이르고 많은 그림과 표, 문헌목록을 포함한 문서를 생각해보자.

```
\documentclass{myuniversityscustomclass}

\usepackage{nonstandardpackage}
```

[2] 더 정확하게 말하자면 내가 가장 많이 포스팅한 패키지들을 다루고 있다.

```
\usepackage{anothernonstandardpackage}
% lots of other packages

\usepackage{glossaries}

% lots of your own command and environment definitions

\newglossaryentry{minex}{name={Minimal Example},
description={A small document illustrating failing behaviour},
text={minimal example}}

% lots more glossary definitions

\author{John Doe}
\title{Astounding Discoveries}

\begin{document}
\maketitle
\tableofcontents
\listoffigures
\listoftables

% 300 or so pages of text, graphics, tables and
% sundry other stuff

% Somewhere in the document is the following:

A \gls{minex is essential when encountering a \TeX\ or \LaTeX\
error you don't understand.

% Lots more text, figures, tables and a bibliography
\end{document}
```

이 문서는 다음 에러를 보인다.

```
Runaway argument?
{minexam is essential when encountering a \TeX\ or \LaTeX\ ^^Merror
\ETC.
! Paragraph ended before \@gls was complete.
<to be read again>
           \par
```

이러한 에러가 어디서 왜 발생했는지 알지 못하겠다고 하자.[3] 어떤 명령에서 문제가 생긴 건지 모르기 때문에 앞선 절에서 보인 것과 같은 방법을 쓸 수가 없다. 여기서 필요한 것이 "지워나가는" 접근방법이다.

다른 작업을 진행하기 전에 문제가 된 문서를 **복사**한다. 원래의 문서를 이를테면 `test.tex`으로 복사하여 이 파일만을 수정하자. 문제가 해결될 때까지 원래 문서는 건드리지 않아야 한다. 그렇지 않으면 작업을 날릴지도 모른다.

문제가 생긴 부분을 발견하는 한 가지 방법은 반분 탐색(binary search)법이다. 이 파일 소스 코드가 1000줄로 되어 있다고 하자. 그러면 500줄 근처에서(즉, 반 정도 되는 곳에서) 다음을 삽입한다.[4]

```
\end{document}
```

(이 명령이 다른 환경이나 그룹(중괄호로 둘러싸인 부분) 안에 오지 않도록 주의하라.)

그리고 LaTeX을 실행하자. 문서의 반을 없앴기 때문에 몇 가지 경고가 보일 수 있지만 여기서는 무시한다.

- 여전히 에러가 발생한다면 문제는 문서의 전반부에 있는 것이다. 그러므로 테스트 파일의 `\end{document}`를 삽입한 이후 부분을 지워버린 후에 이 과정을 반복한다.
- 에러가 발생하지 않는다면 문제는 문서의 후반부에 있는 것이다. 이럴 경우 테스트 파일의 `\begin{document}` 이후부터 방금 삽입한 `\end{document}`까지를 지운다. 그리고 이 과정을 반복한다.

이 과정을 반복하여 테스트 파일에 문제가 일어나는 단 하나의 문단만을 남긴다. 문제가 발생한 곳이 `\input`이나 `\include` 명령이 쓰인 부분이라면, 이 명령을 삭제(또는 주석처리)하라. 에러가 나오지 않는다면 문제는 그 파일에 있는 것이므로 `\input`한 파일의 내용을 그대로 테스트 파일로 가져와서 이 과정을 반복한다. `\input`한 파일 자체를 수정하지 말고 테스트 파일로 복사해와서 검토하도록 하라. 완료되면 `\listfiles`를 추가하는 것이 좋다.

그러면 테스트 파일이 다음과 같은 모양이 된다.

```
\listfiles
\documentclass{myuniversityscustomclass}

\usepackage{nonstandardpackage}
```

[3] 사실 이 예제에서 `\gls`가 짧은 명령이기 때문에 에러 메시지에 행 번호가 표시된다. 그러나 에러 메시지의 행 번호가 항상 도움이 되는 것은 아니기 때문에 그것이 없다고 치고 계속한다.

[4] LaTeX은 처음으로 만나는 `\end{document}`를 문서의 끝으로 인식하여 그 이후에 오는 것은 모두 무시한다.

```
\usepackage{anothernonstandardpackage}
% lots of other packages

\usepackage{glossaries}

% lots of your own command and environment definitions

\newglossaryentry{minex}{name={Minimal Example},
description={A small document illustrating failing behaviour},
text={minimal example}}

% lots more glossary definitions

\begin{document}

A \gls{minex is essential when encountering a \TeX\ or \LaTeX\
error you don't understand.

\end{document}
```

이제 문제가 어디에서 온 것인지 확인할 수 있을 것이다. 그러나 왜 에러가 발생하는지는 여전히 알지 못한다. 그 다음에 할 일은 프리앰블에 남아 있는 불필요한 정보를 제거하는 것이다. 프리앰블에 정의한 명령이나 환경이 문제가 된 문단에서 쓰이고 있지 않다면 그 정의 부분을 삭제한다. 새로운 theorem이나 glossary 등등. 이 보기에서 문제가 되는 문단에 용어 엔트리가 포함되어 있으므로 그 부분의 정의는 유지하고 나머지는 지운다.

```
\listfiles
\documentclass{myuniversityscustomclass}

\usepackage{nonstandardpackage}
\usepackage{anothernonstandardpackage}
% lots of other packages

\usepackage{glossaries}

\newglossaryentry{minex}{name={Minimal Example},
description={A small document illustrating failing behaviour},
text={minimal example}}
```

```
\begin{document}

A \gls{minex is essential when encountering a \TeX\ or \LaTeX\
error you don't understand.

\end{document}
```

그런 다음 문제에 영향을 끼치지 않는 패키지들을 하나씩 삭제한다. 패키지를 삭제할 때는 테스트 파일에 LaTeX을 실행하여 에러가 없어지지는 않는지 확인하라. 만약 어떤 패키지를 삭제했을 때 에러가 사라진다면 그 패키지를 되돌린다. 패키지를 제거하자 "Undefined control sequence"라는 메시지가 나온다면 정의되지 않은 명령도 삭제한다. 이렇게 하여서 문제가 사라진다면 명령을 다시 추가하고 패키지를 되돌린다. 예를 들어 다음 행을 삭제하였을 때 `\newglossaryentry`나 `\gls`가 정의되지 않았다는 에러가 나온다.

```
\usepackage{glossaries}
```

이 명령을 지워버리면 원래의 에러 메시지가 더이상 나오지 않는다. 그러므로 이 명령을 테스트 파일에 남겨두어야 하고 glossaries 패키지도 그대로 두어야 한다.

그 다음 단계는 클래스 파일을 article이나 report로 바꾸어보는 것이다. 이렇게 했을 때 에러가 사라진다면 문제는 원래의 클래스 파일에 있는 것이므로 그 클래스를 쓰도록 되돌려야 한다. 만약 클래스 파일이 공개적으로 사용할 수 없는 것(예를 들어 대학교 논문 양식과 같이 비공개의 자작 클래스)이라면 클래스 파일의 저자에게 연락하여 테스트 파일과 로그 파일을 보내주어야 한다. (물론 먼저 문서를 찾아보는 것을 잊어서는 안 된다.)

이 절차를 끝내면 테스트 파일은 다음과 같이 된다.

```
\listfiles
\documentclass{article}

\usepackage{glossaries}

\newglossaryentry{minex}{name={Minimal Example},
description={A small document illustrating failing behaviour},
text={minimal example}}
```

```
\begin{document}

A \gls{minex is essential when encountering a \TeX\ or \LaTeX\
error you don't understand.

\end{document}
```

이렇게 하면 `\gls` 명령의 인자를 지정할 닫는 중괄호가 빠져 있다는 것을 금방 알게 된다. 그러나 여전히 뭐가 문제인지 알지 못하겠다면 (문서를 찾아 읽고 포럼이나 뉴스그룹 등을 충분히 찾아보았다고 가정할 때) 테스트 파일의 내용을 첨부하여 StackExchange나 The LaTeX Community나 comp.text.tex에 게시물을 포스팅할 수 있다.

4 외부 파일

상향식 접근 (section 2)과 하향식 접근 (section 3)에 대해서 알아보았다. 그런데 문제가 최소 예제 파일로 복사/붙이기할 수 없는 부수파일에서 발생한다면 어떻게 할 것인가?

그 파일이 그림 파일이라면 그 명령을 같은 크기를 갖는 rule로 대치한다. 예컨대 그림이 4×3 크기라면 `\includegraphics{myImage}`를

```
\rule{4in}{3in}
```

로 바꾼다. 또는 mwe 패키지에 대신 사용할 수 있는 몇 가지 샘플 이미지가 포함되어 있다. 그러므로 `\includegraphics{myImage}`를

```
\includegraphics[height=3in]{example-image}
```

와 같이 수정할 수 있다. (이 패키지가 제공하는 다른 그림도 있으므로 mwe 패키지 문서를 참고하라.)

만약 문제가 되는 파일을 BibTeX 파일이라면 그 파일을 일단 복사한 후에 에러를 일으키는 엔트리 하나가 남을 때까지 모든 엔트리를 하나씩 삭제한다. CSV 파일이 문제라면 그 파일을 복사한 후에 문제가 되는 행만 남을 때까지 모든 행을 (헤더 행이 있으면 이것은 남겨둔다) 하나씩 삭제한다. 이렇게 만들어진 임시 파일을 최소 예제와 함께 보내거나 filecontents 또는 filecontents* 환경을 이용하여 최소 예제 파일에 포함한다.[5] 이 환경은 외부에 쓸 파일 이름에 해당하는 인자 하나를 취한다.

[5] 별표붙은 명령은 추가된 주석을 파일에 쓰지 않는다.

```
\documentclass{article}

\begin{filecontents*}{test.bib}
@article{sample,
  author={Ann Other},
  title={Sample Title},
  journal={Journal of Something},
  year=2014
}
\end{filecontents*}

\begin{document}
\cite{sample}

\bibliography{test}
\end{document}
```

5 채우기 텍스트

이따금 문제가 어떤 특정 위치나 어떤 페이지에서 일어나는 수가 있다. 이럴 경우 예제 파일에 채우기 텍스트로 페이지를 채울 필요가 있다. lipsum 패키지가 유용하다. 이 패키지는 `\lipsum`이라는 명령을 제공하는데 옵션을 주어서 특정 문단 또는 몇 개 범위의 문단을 식자하게 한다.

예를 들면 book 클래스를 사용하고 있고 왜 두 번째 페이지는 페이지 번호가 상단에 오는데 chapter의 첫 페이지에서는 하단에 페이지 번호가 찍히는지 이해하지 못하겠다고 하자. 이 경우를 나타내는 최소 예제는 다음과 같이 된다.

```
\documentclass{book}

\usepackage{lipsum}

\begin{document}
\chapter{Sample}

\lipsum[1-4]
\end{document}
```

이렇게 하면 두 페이지를 조판하기에 충분한 텍스트를 적을 수 있다.

다른 채우기 텍스트 패키지로 blindtext라는 것이 있는데 이 패키지가 제공하는 명령은 \blindtext(짧은 문장)와 \Blindtext(긴 문장)이다. 예를 들어본다.

```
\documentclass{book}

\usepackage{blindtext}

\begin{document}
\chapter{Sample}

\Blindtext
\end{document}
```

blindtext 패키지는 무작위 문서, 더미 리스트 등과 같은 다른 명령도 함께 제공한다. 자세한 사항은 패키지 문서를 참조하라.

6 패키지 문서를 찾아 읽는 방법

요즘 대부분의 패키지 문서는 PDF 파일로 제공된다. 시스템에 패키지를 설치하였다면 texdoc 프로그램을 이용하여 문서를 읽을 수 있다. 터미널 또는 명령행에서 texdoc 다음에 패키지 이름을 써주면 된다. datetime 패키지의 문서를 읽으려면 다음과 같이 한다.

```
texdoc datetime
```

가끔 패키지 이름과 패키지 문서 이름이 같지 않은 때가 있다.

```
texdoc flowfram
```

이 명령은 소스 코드 문서를 보여줄 것이다. 그런데 이 문서의 처음 부분을 잘 보면 ffuserguide.pdf라는 사용자 설명서 이름을 알려주고 있을 것이다. 그러므로

```
texdoc ffuserguide
```

라고 하여 사용자 설명서를 읽을 수 있다.

어떤 경우 (특히 오래된 패키지) 문서가 README 파일로 존재하거나 .sty 또는 .cls 파일 시작 부분이나 끝에 주석으로만 되어 있는 경우도 있다.

한편, 문서가 자신의 시스템에 설치되어 있지 않을 때, CTAN에서 다운로드받아서 볼 수도 있다. CTAN 홈페이지의 찾아보기 박스를 이용하거나, 아니면 직접 http://

ctan.org/pkg/<*name*> (여기서 <*name*>은 패키지의 이름) URL로 접속하여 보아도 된다. 예를 들어 glossaries 패키지에 대한 정보를 얻으려면 http://ctan.org/pkg/glossaries로 접속한다. 해당 페이지에 문서에 대하여 링크가 제공될 것이다.

7 에러 메시지 이해하기

TEX과 LATEX의 에러 메시지는 영문모를 소리처럼 보이기도 하지만 가끔 메시지를 분석해 보면 적어도 어디서 뭐가 잘못된 건지를 알아낼 수 있다.

다음과 같은 문서를 생각해보자.

```
\documentclass{article}

\newcommand{\example}[1]{#1}

\begin{document}
This is a sample document that contains a long
command \example{with an error.

This is the next paragraph
\end{document}
```

이것을 컴파일하면 다음 에러 메시지를 만난다.

```
Runaway argument?
{with an error. \par This is the next paragraph \end {document}
! File ended while scanning use of \example.
<inserted text>
         \par
```

첫 줄("Runaway argument?")은 에러의 유형을 가리킨다. runaway argument(인자 없음) 에러는 보통 닫는 중괄호를 빠뜨렸을 때 나타난다. 그 다음 줄은 에러가 발생하기 직전에 TEX이 처리하던 일을 보여준다. 이 에러 메시지 예시에는 행 번호가 빠져 있지만 보통 에러 발생 위치를 찾기 쉽도록 행 번호를 함께 보여준다. 이 행의 첫 부분 (즉 `{with an error`)을 복사하여 에디터의 찾기 기능의 찾을 텍스트로 붙여넣는다. 이렇게 하여 문제가 발생한 행을 찾아보면 닫는 괄호가 빠져 있는 것을 볼 수 있을 것이다.

이번에는 문서가 다음과 같다고 하자.

```
\documentclass{article}

\newcommand*{\example}[1]{#1}

\begin{document}
This is a sample document that contains a short
command \example{with an error.

This is the next paragraph
\end{document}
```

에러 메시지는 다음과 같다.

```
Runaway argument?
{with an error.
! Paragraph ended before \example was complete.
<to be read again>
          \par
l.8
```

이 에러 메시지에는 뭔가가 잘못되기 시작한 곳의 행 번호(l.8)가 나와 있다. 그러므로 에디터에게 "행으로 가기"를 실행하게 할 수 있다.

때로 에러 메시지에서 주어진 행 번호가 실제 에러가 발생한 행을 가리키지 않을 때가 있다. 다음 예를 보자.

```
\documentclass{report}

\author{A.N. Author}
\title{A sample document with a \badcommand}
\date{14th November, 2008}

\begin{document}
\maketitle
\end{document}
```

이 문서에서 에러는 명령(\badcommand)이 정의되지 않은 것으로 4행에 있는 것이다. 그러나 에러 메시지에서는

```
! Undefined control sequence.
```

```
\@title ->A sample document with a \badcommand

l.8 \maketitle
```

이와 같이 8행에서 문제가 생긴 것으로 나온다. 그 이유는 8행의 표제지를 조판하는 명령 `\maketitle` 명령을 만나기 전까지 TEX은 실제로 `\badcommand`를 해석하지 않았기 때문이다.

이런 종류의 상황에 맞닥뜨리면 문제를 추적하기 위해 약간 고심을 해야 한다. 위의 경우에는 두 가지 방법이 있다.

1. 에러 메시지의 첫 줄이 에러의 종류를 보여주고 (명령이 정의되지 않았음) 두 번째 줄은 `\badcommand`가 정의되지 않은 명령임을 가리키고 있다. 그러므로 에디터에서 `\badcommand`나 나오는 곳을 모두 찾아서 그것을 적당한 명령으로 바꿀 수 있다. 아니면 해당 명령을 정의하는 패키지를 로드하지 않았거나, 스스로 작성한 명령이라면 그 명령의 정의를 잊었거나 한 것이다.
2. 에러 메시지의 마지막 줄은 8번 행에서 문제가 발생했음을 보여준다. 그리고 그 행에 `\maketitle`이라는 명령이 있다. `\maketitle`에 영향을 주는 명령은 무엇인가? report, article 같은 일반적인 클래스에서 `\author`, `\title`, `\date`가 그러하다. 따라서 이 명령들이 사용된 곳의 코드를 검토한다. 여전히 에러가 사라지지 않으면 하나만 남겨두고 나머지는 주석처리해보자. 만약 title과 date가 포함된 행을 주석처리하였는데 에러가 없어졌고, author와 date를 주석처리하니까 에러가 남아있다면, 이 에러는 title에서 온 것임을 알 수 있다.

에러 메시지를 더 많이 이해하려면 UK TEX Faq의 How to approach errors 문서를 보라. 이 글과 같은 사이트에서 찾을 수 있는 LATEX for Complete Novices 문서의 일반적인 에러 메시지 목록도 참고할 수 있다.

8 역자의 말

한국인이라면 다음 사항을 고려할 수 있다.

1. StackExchange 등 질문을 할 수 있는 사이트로 KTUG이 있다.
2. 문서 클래스로 oblivoir가 있다.
3. 채우기 텍스트를 만드는 패키지로 jiwonlipsum이 있다.

oblivoir나 jiwonlipsum을 사용하여 만들어진 한글 최소 예제 문서라면 KTUG에 질문하는 것이 가장 확실한 방법이다.