

LyX으로 한글 문서 작성 (1) — 문서의 기본 틀

제 1 절 시작하기 전에

이 글은 LyX이라는 문서 처리기(Document Processor)를 처음 사용하는 사람을 위하여 일종의 처음 사용 안내서로 작성하는 것이다. 그러나 글의 초점은 “설치하기”에 있지 않고 “문서 작성하기”에 놓인다. 설치와 설정에 대해서는 최소한 필요한 언급에 그쳤다.

LyX은 초보자가 L^AT_EX 문서 작성에 익숙해지는 매우 훌륭한 입문 도구라고 생각한다. 또한 숙련자에게도 문서 작성을 훨씬 편리하게 만들어준다. 이 글이 L^AT_EX과 LyX을 좋아하는 사람이나 처음 대하는 사람에게 약간의 도움이 되기만을 바란다.

이 문서는 매킨토시 상의 LyX 1.5.5에서 작성하였다. 아마 다른 운영체제 아래서도 잘 열리고 처리될 것이다.

다음 몇 가지 용어를 확인해두자.

LyX “릭”이라고 읽는다. “릭스”라고 발음하는 사람도 있지만 T_EX (텍) 또는 L^AT_EX (레이텍, 라텍)과 운을 맞추기 위해서 “릭”이라는 발음을 채택하기로 한다. 특히 라텍은 라텍스라고 읽히면 모욕감을 느끼지만 릭은 릭이든 릭스든 크게 상관없을 듯하다. 릭에 대한 전반적인 사항은 홈페이지(<http://www.lyx.org>) 참고. 한글을 원활하게 쓰기 위해서 약간의 추가적인 작업을 해주어야 하는데 이에 대해서는 <http://faq.ktug.or.kr/faq/Karnes/LyX>을 참고한다.¹⁾

T_EX, L^AT_EX 별도의 설명은 생략. LyX은 프론트엔드 L^AT_EX 에디터라고 생각해도 된다. 물론 “L^AT_EX을 백엔드로 하는 WYSIWYM 워드 프로세서”라고 하는 사람도 있다. LyX을 운영하기 위해서는 T_EX 시스템이 반드시 필요하다. 더 자세한 것은 <http://www.ktug.or.kr>을 방문해보라. 여기서는 KTUG에서 배포하는 KC2007이라는 시스템을 사용할 것을 권장한다.

WYSIWYM “화면에 나타나는 것은 저자의 의도 (What you see is what you mean)”라는 의미이다. LyX의 철학을 압축적으로 보여주는 문구이므로 이것을 잘 이해하는 것이 중요하다. 원래 이 말은 매킨토시 그림 운영체제에서 시작되어 디자인 툴 및 워드 프로세서에 이르기까지 화면상에 보이는 것이 출력물과 동일(유사)하다는 것을 의미하는 “WYSIWYG (What you see is what you get)”의 아날로지로서 만들어진 말이다. LyX은 WYSIWYG가 아니다. 그러나 시각적 요소를 이용하여 “출력물의 형태”가 아닌 “저자가 하고자 하는 것”을 표현하도록 설계되어 있다. 예를 들면 LyX에서

1) 유니코드 지원 LyX은 1.5 이후 버전이다. 그 전에는 유니코드를 사용할 수 없었던 관계로 CJK-LyX이라는 변형 LyX이 별도로 있었는데 컴파일도 어렵고 윈도나 맥에서 이 프로그램을 잘 사용할 수 없어서 사실상 한글 사용이 대단히 어렵던 시절이 있었다. 지금도 LyX 관련 검색을 해보면 “한글을 사용하려면 CJK-LyX을 쓰라”는 말을 찾을 수 있다. 그러나 현재 1.5 이후 버전에서 유니코드 한글 사용은 매우 쉽고 잘 된다. 이것은 한글 L^AT_EX이 유니코드 베이스로 이행했기 때문에 손쉬워진 측면도 있다.

Section 문단은 그것이 Section임을 분명히 보여준다. 그러나 $\text{L}_\text{Y}\text{X}$ 화면에 표시된 것이 실제 출력물의 인쇄 상태를 보여주지는 않는 것이다. 이를 통하여 $\text{T}_\text{E}\text{X}$ 과 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 의 강력한 매크로 및 조판 기능과 “내용에만 집중할 수 있게” 해주는 글쓰기의 편의성을 결합할 수 있게 되었다. 지금 입력하는 내용이 어떻게 인쇄될까에는 관심을 갖지 말고 오직 그것이 구조적으로 어떤 의미를 갖는 것인가에만 집중하는 것이 최선이다.²⁾

제 2 절 준비, 설치

“시작이 반”이듯이, 설치 (installation)가 반이다. 설치에 성공했다면 반은 성공한 것이나 다름없다. 다행히 설치 과정은 매뉴얼이 잘 마련되어 있으므로 이것을 그대로 따라하면 된다. 중요한 것은 매뉴얼을 잘 읽고 이를 충실히 따르는 것이다. 각 단계가 무슨 의미인지 모르겠다면 일단 그대로 따라하라. 텍 시스템과 릭에 익숙해진 후에는 자신만의 환경을 구축하는 것도 가능하다. 설치를 위해서 참고할 사이트를 지적하는 것으로 이 절을 대신하려 한다.

설치 과정에서 문제가 생기거나 의도와 다르게 동작한다면 주저하지 말고 질문하는 것이 좋다. 텍 또는 릭 설치에 관련된 질문은 KTUG의 설치운영게시판(<http://www.ktug.or.kr>)을 이용한다.

- (1) 텍 시스템을 설치한다. <http://faq.ktug.or.kr/faq/%BC%B3%C4%A1>.
- (2) $\text{L}_\text{Y}\text{X}$ 을 설치한다. <http://www.lyx.org>.
- (3) $\text{L}_\text{Y}\text{X}$ 을 한글화한다. <http://faq.ktug.or.kr/faq/Karnes/LyX>.

위의 과정을 그대로 따르면 될 것이다. 윈도우즈, 리눅스, 매킨토시에 모두 동일한 환경을 구축할 수 있다. 다만 한글 사용을 위해서 리눅스나 매킨토시에서는 ko. $\text{T}_\text{E}\text{X}$ 을 반드시 설치하고 운영할 수 있게 준비해두어야 할 것이다.

제 3 절 $\text{L}_\text{Y}\text{X}$ 가지고 놀기

준비가 끝났으면 간단한 문서를 만들면서 $\text{L}_\text{Y}\text{X}$ 의 특징을 익혀보도록 하자.

3.1 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 과 $\text{L}_\text{Y}\text{X}$

그림 1은 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 문서 작성과 $\text{L}_\text{Y}\text{X}$ 문서 작성을 직관적으로 비교한 것이다. 그림에서 보듯이, $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 문서 작성은 내용을 제외한 모든 ‘명령’을 mark-up하여야 한다. 이것은 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 의 장점인 동시에 번거로운 점이기도 하다. $\text{L}_\text{Y}\text{X}$ 의 편집 화면은 이러한 ‘명령 마크업’을 시각적 요소로 바꾸어준다. 어느 쪽이 문서의 내용에 더 집중할 수 있을지 상상해보라! 실제 이 두 문서는 완전히 동일한 결과를 얻기 때문에 $\text{L}_\text{Y}\text{X}$ 이 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 에디터 역할을 할 수 있는 것이다. $\text{L}_\text{Y}\text{X}$ 은 이 파일의 (임시 또는 영구적) pdf 결과를 만들 때 일단 임시 폴더에 .tex 파일을 만들어서 $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ 으로 컴파일하여 결과를 얻으며, 이 임시 파일과 같은 내용을 사용자가 원하는

2) 그러나 원한다면 출력 형태를 제어하거나 항상 그 인쇄 상태를 확인하면서 작업하는 것도 가능하다.

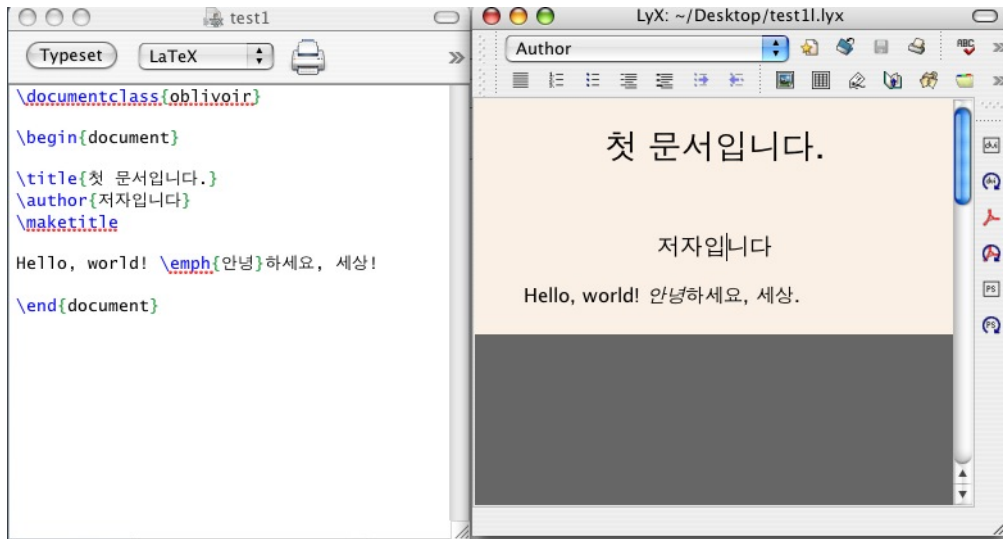


그림 1: L^AT_EX과 LyX

폴더에 .tex 파일로 export 하여 주는 기능이 있다. 즉, 처음에는 LyX으로 대략의 문서를 작성한 후, .tex 파일을 얻어서 이를 직접 수정하는 방법도 있다는 것이다.

한편, LyX이 아무리 잘 만들어진 L^AT_EX 프론트엔드라 하더라도 전세계 수많은 사용자가 각자 자기 식으로 정의한 모든 명령들을 전부 이해하게 할 수는 없는 일이다. 그래서 LyX에는, 필요하다면 문서 작성자가 직접 문서의 일부에 L^AT_EX 마크업을 써넣을 수 있도록 해 두고 있는데, 그래도 본문과 이 “써넣은 L^AT_EX 명령” —이런 명령이 들어간 부분을 ERT 박스라고 한다—은 시각적으로 구분이 잘 되기 때문에 본문의 흐름을 놓치게 만들지 않는 장점이 있다. 만약 L^AT_EX 숙련자라면 ERT 박스를 잘 활용하여 문서를 얼마든지 풍부하게 작성할 수도 있는 것이다.

문장의 흐름을 놓치지 않게 한다는 것 이외에도 LyX은 L^AT_EX을 직접 사용할 때 신경 써야 할 자잘한 문제들을 거의 대부분 알아서해준다. 대표적인 것이 그림 포맷인데, .jpg니, .png니 하는 그림 형식을 사용자는 거의 의식할 필요가 없다. LyX은 자신이 이해하는 그림이기만 하면(여기에는 심지어 .bmp, .gif 등도 포함된다) 문서에 그림을 포함하는 것을 알아서 다 해결해주는 것이다. 게다가 그 그림을 화면으로 보면서 문서 작업을 할 수도 있다. 수식의 입력이 마치 WYSIWYG 수식 편집기와 비슷한 방식으로 이루어질 뿐 아니라 TeX/L^AT_EX 명령 입력 방식도 똑같이 동작하는 것이라든가, LyX의 수식을 복사해서 에디터로 가져오면 TeX 수식 형식으로 붙일 수 있다든가 하는 것은 어쩌면 당연한 듯이 여겨지기도 하지만 실제로 매우 편리하다. 또한 문서 클래스 자체의 문자 인코딩 세트를 미리 설정해놓으면 한글의 인코딩 변환도 자동적으로 이루어지므로 클래스 선택만으로 인코딩 혼선을 피할 수 있다. 윈도우즈 사용자의 경우, 미리보기에서 Adobe Reader를 LyX이 자동으로 달아주는 것이 대단히 편리하게 여겨지는 경우도 많을 것이다.³⁾

3) LyX에 장점만 있다고 주장하고 싶지는 않다. 이런 편리한 점이 있는 대신 또 불편한 점도 없지 않다. 그러나 크게 보면 이런 편리한 점이 소구하는 바가 더 큰 경우가 많을 것이라고 믿는다. 아무튼 그런 단점 중에는

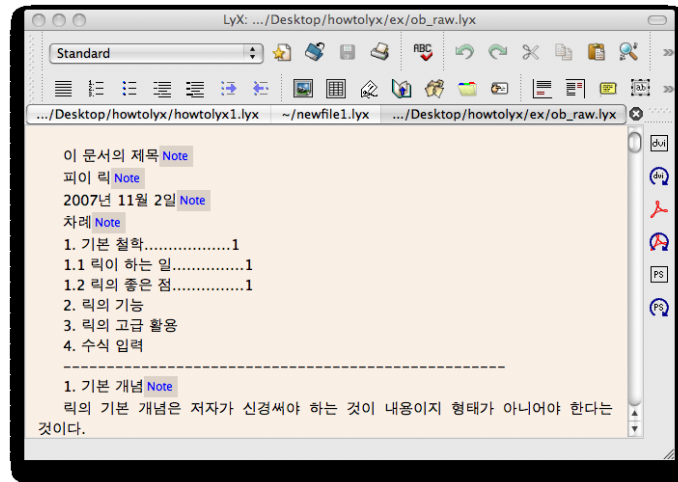


그림 2: 샘플 예제 파일

3.2 첫 문서 편집

LyX 맛보기용 파일이 준비되어 있다. 이 파일은 거의 아무런 형식적 요소가 없는 플레인 텍스트 파일이다. 이것을 가지고 제대로 된 문서로 바꾸면서 LyX으로 할 수 있는 일을 테스트해보는 것이다. 연습용 파일은 각각 `ob_raw.lyx`과 `ob_lyxified.lyx`이라는 이름으로 되어 있고, 이 글의 소스와 함께 압축되어 있다.

그림 2는 `ob_raw.lyx`을 연 모습이다. 이것을 한 줄씩 지시대로 (지시사항 또는 힌트는 Note 박스를 클릭하면 볼 수 있다) 수정하면 `ob_lyxified.lyx`과 같은 결과를 얻을 수 있다. 이 파일을 수정하는 것 자체가 훌륭한 연습이 될 것이다.

예를 들면, 첫 행에 커서를 놓고 상단의 문단 모양 박스를 끌어내려 열어서 ‘Title’을 고르고 이를 적용한다(그림 3).

3.3 한글 문서를 위한 설정

`ob_lyxified.lyx`을 열어보자. 이 문서는 한글 문서에 제일 먼저 적용하는 몇 가지 설정이 이루어져 있다. 이 사항을 확인한다. 만약 한글 문서를 작성하려 한다면 반드시 고려하여야 할 점들이다.

문서 클래스 LyX의 메뉴에서 문서(Document) → 구성(Setting)을 선택하면, 문서 형식(Document Class)이라는 항목을 볼 수 있다. 여기서는 `oblivoir`로 되어 있을 것이다. 한글 문서를 작성하는 데 있어 일반적으로 가장 쉽고 권장되는 것이 `oblivoir` 클래스 문서로 작성하는 것이다. 선택사항이란 클래스 옵션을 의미하는 데, 필요에 따라 여러 옵션을 추가할 수 있다. 이 클래스 옵션은 `oblivoir.cls`의 옵션에 의하는데, 여기 직접 써넣지 않고 별도로 선택하는 사항이 있으므로 중복되지 않도록 한다.

에러가 발생했을 때 초보자가 대처하기 더 어렵다는 문제도 있다.

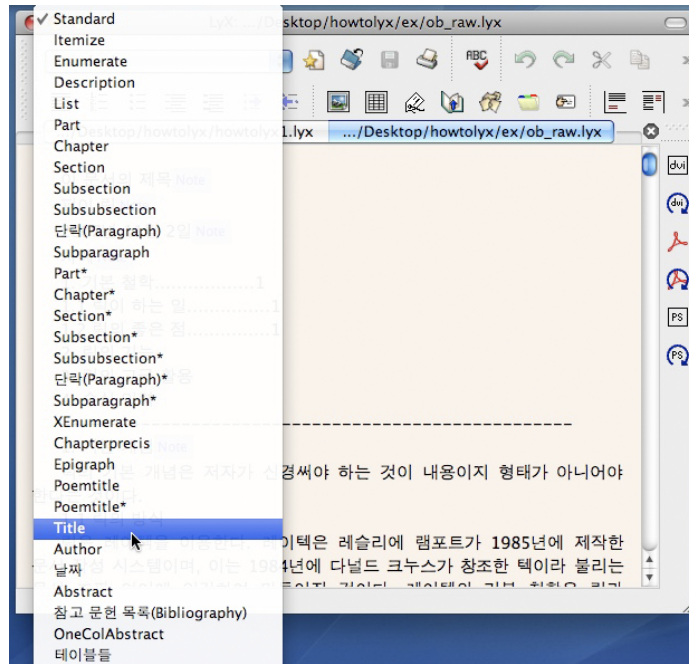


그림 3: 샘플 예제 파일 : 문단 스타일 선택

이 항목에서 써넣을 수 있는 옵션 중에서 예를 들면, `microtype`, `nobookmarks`, `chapter` 등이 있는데, 이들은 `oblivoir` 문서의 설정을 의미한다. 반면 `a4paper`, `10pt` 와 같은 옵션은 아래 설명을 참고하도록 한다.

폰트 Base Size 항목을 선택하면 문서의 기본 폰트 크기를 고를 수 있다. 기본 설정은 10pt 이다. 만약 12pt를 선택하면 문서의 기본 폰트 크기가 변한다. 대부분 이 페이지의 옵션을 바꾸지 않는 것이 좋다.

본문 모양새 (text layout) 여기서는 문단 들여쓰기로 문단 구분을 하는 것이 디폴트로 선택되어 있다. 일부 특별한 문서에서 문단 구분을 문단 간격으로 하고 `parindent` 를 0으로 하는 방법이 있다. `oblivoir` 문서에서 행간격은 주의를 요한다. `oblivoir` 자체가 행 간격을 특별한 방법으로 제어하고 있기 때문에 이 부분은 디폴트 상태로 두는 것이 좋다.⁴⁾ 페이지 여백의 경우도 마찬가지인데, `oblivoir` 의 경우 `fapapersize` 패키지를 이용하는 편이 좋다. 그러므로 이 항목의 경우에도 기본설정 여백을 사용한다.

언어 언어는 `KoreanUCS`로 설정되어 있다. 만약 다른 언어로 바뀌어 있다면 `KoreanUCS` 를 선택하고 `Use language's default encoding` 을 체크한다.

Numbering & TOC 이 부분은 자신의 취향에 맞게 조절하도록 한다.

4) 예를 들면 `\SetHangulspace` 등의 명령을 이용한다.

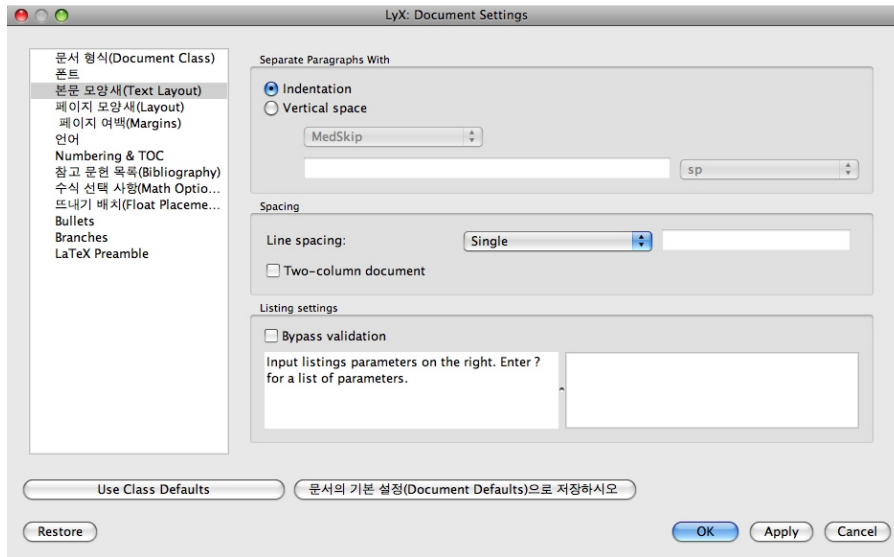


그림 4: 문서 기본 설정

참고 문헌 목록 기본값(default)이 기본이지만 natbib이나 jurabib을 사용할 수 있게 되어 있다. 최근 각광받고 있는 biblatex을 쓰려 한다면 default로 설정해두고 \LaTeX Preamble에서 설정을 해야 할 것이다.

수식 선택 사항 AMS 수식꾸러미 사용 또는 자동 사용을 선택하고 esint는 disable할 것을 권장한다. KC2007의 경우 esint는 별도로 패키지를 설치해야 하는 불편이 있다. 이것은 적분 기호의 표시에 해당하는 것이지만 ams 적분기호에 만족한다면 굳이 esint를 쓸 필요는 없다.

뜨내기 배치 디폴트로 한다.

Bullets itemize 환경을 사용할 때 수준별 불릿의 모양을 선택할 수 있다. 만약 필요하다면 문서에서 직접 바꾸면 되므로 지금 이 부분을 수정할 필요는 없지 않을까?

Branches 지금은 그냥 둔다.

\LaTeX Preamble 위에서 언급된 문서 설정 선택(용지 크기, 언어, 폰트 등)을 제외한 일반적인 설정을 위하여 사용자가 preamble에 직접 \LaTeX 코드를 써넣을 수 있다. 여기에 입력하는 모든 내용은 export하는 .tex 파일에 포함되며 문서의 모양에 영향을 끼친다. 예를 들어 multirow 패키지를 포함하는 등의 설정은 여기에서 직접 `\usepackage{multirow}`라고 써넣는다.

템플릿으로 저장하기

위의 설정을 문서 작업을 시작할 때마다 해야 하는 것은 피곤하다. 그러므로 이것을 템플릿으로 저장해두면 매우 편리하게 유사한 문서를 작성할 수 있을 것이다. LyX의 별도의 템플릿 포맷은 없고 평범한 문서를 Templates 폴더에 넣어두면 되는 것인데, 윈도우즈 LyX의 경우 템플릿 폴더는 C:\Program Files\LyX*\Resource\Template이다. 꼭 이 폴더에 저장하지 않아도 적당한 곳에 두고 템플릿으로 이용하는 것으로 충분하다.

makeindex와 bibtex

KC2007의 makeindex를 쓰려면 runkomkindex.bat 라는 배치파일을 부르는 것이 가장 좋다. makeindex의 설정을 바꾸거나 별도의 스타일을 적요하려면 이 배치파일을 수정하도록 되어 있다. makeindex 명령은 Preferences -> Outputs -> LaTeX 에서 수정할 수 있다.

3.4 패키지가 없는 경우

KC2007은 kcpm이라는 유틸리티를 통하여 필요한 패키지를 계속 업데이트하면서 작업하도록 설계되어 있다. 그런데 LyX은 시스템에 설치된 패키지를 미리 인식하고 작업해야 하기 때문에 이따금 LyX 문서를 만들 때 패키지가 없어서 곤란을 겪는 경우가 있다. 물론 kcpm을 이용하면 되지만 좀 번거롭다.

KC2007을 사용한다고 전제했을 때 이 문제를 피해가는 방법으로 L^AT_EX 또는 pdfL^AT_EX 컴파일러를 아예 kcclogmpm으로 해두는 방법이 있다. Preferences의 Converters 항목에서 LaTeX (pdflatex) -> PDF (pdflatex) 이라고 되어 있는 부분을 다음과 같이 수정한다.

```
kcclogmpm --pdflatex $$i
```

이제 latex 대신 kcclogmpm이 돌면서 필요한 패키지가 있다면 자동으로 설치해줄 것이다. 다만 이 방법을 쓰게 되면 컴파일 타임 에러를 검출하는 데는 약간 지장이 있으므로 자신의 선택에 따라 쓰도록 하는 것이 좋겠다.