

초간단 xoblivoir under Xe_ΛTeX 사용법

x-ob-liv-oir

2008년 9월 8일 ~ 2010년 2월 11일 (열두번째 고쳐쓰)

요약

xoblivoir는 Lua_ΛTeX과 Xe_ΛTeX에서 사용할 수 있는 oblivoir이다. 이 문서는 Xe_ΛTeX에서 xoblivoir를 사용하는 방법을 기술한다. oblivoir와 공통인 부분은 기술에서 제외하였다. 이 문서에서 언급하지 않은 것은 모두 oblivoir의 방식대로 하면 된다.

차례

제 1 절	xoblivoir?	3
제 2 절	oblivoir와 비교	3
제 3 절	폰트 설정 방식에 대한 간단한 설명	5
3.1	xetexko 방식의 한글 폰트 설정	6
3.2	폰트 가족의 지정	7
3.3	디폴트 치환	8
3.4	ExternalLocation	9
3.5	파일 이름 문제: 조금 더	10
3.6	이탤릭, 기울임	11
3.7	따옴표	12
3.8	enumerate	12
3.9	graphicx, xcolor	12
3.10	참조 인용, 자동 조사	12
3.11	방점	13
제 4 절	기타	13

4.1	fontspec 옵션과 수학 글꼴	13
4.2	moreverb	14
4.3	옛한글과 세로쓰기	14
4.4	amssymb	14
4.5	flowfram	15
4.6	preload	15
4.7	microtype	15
제 5 절	첨언	16
제 6 절	이 문서의 폰트사용 설정	16
제 7 절	버전 인포	16

제 1 절 xoblivoir?

xoblivoir는 oblivoir에서 파생된 클래스이다. oblivoir가 \LaTeX , \pdf\LaTeX 을 위한 것이라면, xoblivoir는 \pdf\lua\LaTeX 이나 \X\LaTeX 을 위한 것이다. 이 글은 \X\LaTeX 으로 xoblivoir를 쓰려 하는 경우에 대해서만 기술한다. \Lua\LaTeX 에 대해서는 별도로 특기할 만한 것이 없기도 하려니와 아직 준비가 미흡하여, 차후로 미룬다.¹⁾

xoblivoir는 김도현 교수의 xetexko 패키지를 바탕으로 하고 있다. 이 패키지는 2008년 10월 12일에 처음 발표되었으며 그 이전에 시험되던 xkospace를 확장하고 다듬은 것이다. xetexko-space, xetexko-josa, xetexko-dotemph 및 xetexko-font가 포함되어 있는데 xoblivoir는 이를 바탕으로 하면서 사용자 인터페이스를 조금 확장하고 oblivoir와 호환되게 한 것이다. xoblivoir의 쉬운 인터페이스를 통하여 현재 \TeX 에서의 한글 구현이 어느 단계까지 와 있는지를 일반 사용자도 경험하는 기회가 되기를 바란다. 한편 2010년 학술대회를 전후하여 \X\TeX-ko 는 한글 조판과 식자에 있어 “거의 완전한 단계”에 이르렀다. xoblivoir에서 시도하던 많은 부분이 \X\TeX-ko 자체에 의해 구현되게 된 것도 많으며 실제로 출판 현장에서 이를 활용하는데 부족함이 없을 정도가 되었다. 이제 xoblivoir는 memoir 클래스를 \X\TeX-ko 와 함께 쓰도록 하는 클래스라는데 더 큰 의의가 있게 되었다. 한글 \TeX 개발에 고군분투하시는 김도현 교수께 감사의 말씀을 드린다.

\X\TeX-ko 의 이해 없이 xoblivoir를 사용하기 어렵다. 그러므로 반드시 \X\TeX-ko 매뉴얼을 읽어두는 것은 매우 중요하다. xoblivoir에서 \X\TeX-ko 명령은 원칙적으로 모두 사용할 수 있다. \X\TeX-ko 매뉴얼을 읽으려면,

\$ texdoc xetexko

를 실행한다.

제 2 절 oblivoir와 비교

다음 옵션은 disable되었다. 아무런 작용이 없는 것이므로 에러를 발생하지는 않지만 효과도 없다.

[dvíps] 당연히 더이상 쓸 일이 없으므로.

[romanfixed] 폰트 관련된 옵션은 대부분 비활성화된다.

1) 준비가 미흡하다는 것은 xoblivoir의 입장에서 하는 말이다. 현재도 xoblivoir를 통한 \Lua\LaTeX 은 훌륭하게 사용할 수 있다, 고 생각하고 있다.

[finemath], [nofinemath] 이 옵션은 disable되었지만 xkospace에서 finemath의 일부를 구현하고 있다.

[strictcharcheck] dhucs 특유의 옵션이었다.

다음 옵션이 새로 생겨났다.²⁾

[fontspec] 이 옵션의 동작에 대해서는 4.1을 보라.

[moreverb] 이 옵션의 동작에 대해서는 4.2를 보라.

[amsmath] 이 옵션의 동작에 대해서는 4.4를 보라.

[nowinname] 이 옵션의 동작에 대해서는 3.5를 보라.

[microtype] 이 옵션의 동작에 대해서는 4.7을 보라.

다음 옵션은 그대로 있지만 동작이 달라졌다.

[itemph] emph 또는 itshape에 한글을 기울여서 식자한다.

[gremp] emph 또는 itshape에 한글을 바로 선 폰트로 식자한다.

다음 옵션은 특별한 예외 처리를 위한 것이다. 일반적인 경우에는 쓰이지 않는다. 4.5를 보라.

[famt], [fawd] 문서 전체의 stockheight와 stockwidth를 미리 설정한다.

[preload] 이 옵션의 동작에 대해서는 4.6을 보라.

레거시 ko.TE_X의 폰트 설정 명령들은 전혀 동작하지 않고 에러를 토해낸다는 사실을 알아야 한다. 즉, `\SetAdhocFonts`라든가 `\SetHangulFonts` 같은 명령은 제거되어 있다. X_YLA_TE_X을 쓰는 목적이 주로 폰트 문제와 관련이 있으므로, 종래의 폰트 설정 방식은 모두 X_YLA_TE_X 방식으로 바꾸도록 해야 한다.

그밖의 모든 oblivoir의 기능을 xoblivoir에서 쓸 수 있다. 한글 문서 작성에 있어서 표준이 되고자 하는 oblivoir를 X_YLA_TE_X에서 —LuA_TE_X에서도— 편리하게 사용할 수 있게 되기를 바라 마지 않는다.

2) [latinquote] 옵션은 더이상 아무런 작용을 하지 않는다.

제 3 절 폰트 설정 방식에 대한 간단한 설명

*xoblivoir*는 영문 폰트와 한글 폰트를 분리하는 것을 기본으로 하고 있다. 한글과 영문 글꼴을 분리하지 않으려 할 경우, *X_YTEX-ko* 명령인 `\disablekoreanfonts`를 선언한다. 이렇게 하면 한글 글꼴은 영문 글꼴을 따라가게 된다. 즉, `\setmainfont` 등으로 선언된 글꼴이 한글과 영문에 동시에 식자된다.³⁾

영문 폰트는 *fontspec* 패키지의 방식을 따른다. 설정과 사용에 대해서는 *fontspec* 패키지 문서를 참고하라. 매우 방대하고 훌륭한 문서이다.

```
\setmainfont{<Font Name>}
\setromanfont{...}
\setsansfont{...}
```

*X_YTEXko*에서 한글 폰트는 다음과 같이 설정한다. 자세한 사항은 *X_YTEX-ko* 매뉴얼을 참고하라.

```
\setmainhangulfont[Options]{FontName}
\setsanshangulfont[Options]{FontName}
\setmonohangulfont[Options]{FontName}
\setmainhanjafont[Options]{FontName}
\setsanshanjafont[Options]{FontName}
\setmonohanjafont[Options]{FontName}
```

*xoblivoir*에서도 이 방식의 폰트 정의를 할 수 있다. 이 문법은 *fontspec* 패키지를 따르는 것이다.

한글 폰트만을 이용해서 한글/영문에 모두 식자할 필요가 있을 수 있다. 이 경우 다음과 같이 한다.

```
\disablekoreanfonts
\setmainfont{나눔명조}
```

영문이나 숫자 영역까지도 나눔명조로 식자하게 한다.

한편, 한글 글꼴을 선언하는 데 있어서 *xoblivoir*에서는 약간 다른 방법 (더 편리한?)의 폰트 정의가 가능하다. 이것은 다음 세 명령으로 이루어져 있다.

3) 이외에도 한글-영문 글꼴의 분리에 따르는 여러 가지 옵션 정의에 관련된 *X_YTEX-ko* 매크로가 있으므로 이에 대해서는 *X_YTEX-ko* 매뉴얼을 참고하라.

```
\setkormainfont(<Bold>)(<Italic>){<Regular>}
\setkorsansfont(<Bold>)(<Italic>){<Regular>}
\setkormonofont{<Mono>}
```

이후의 설명은 `\setkormainfont`에 대해서만 한다. 다른 두 명령의 사용법은 동일하다.

3.1 xetexko 방식의 한글 폰트 설정

```
\setkormainfont{Font_Family_Name}
```

한글 글꼴 가운데 어떤 것은 — 모든 한글 글꼴이 그렇지는 않다는 데 주의하라 — 글꼴 가족을 이루고 있는 것이 있다. 예를 들면 “맑은 고딕”은 맥락에 따라 “맑은 고딕 Bold”를 글꼴 가족으로 인식한다. 이런 경우 예를 들어

```
\setsansfont{맑은 고딕}
```

이렇게만 지정해도 굵은 글꼴이 요구되는 곳에서는 “맑은 고딕 Bold”를 식자해 준다. 이렇게 잘 만들어진 글꼴을 이용하는 경우, 간단히 하나의 옵션으로 글꼴을 지정해 주는 것만으로 굵은 글꼴을 사용하게 할 수 있다.

```
\setkormainfont[Options]{Font_Name}
```

만약 각괄호 ([])가 첫 인자로 나오면 이 명령의 사용법은 `\setmainhangulfont`와 동일하다. 즉 `fontspec` 명령을 그대로 쓸 수 있다.

```
\setkormainfont[Options]{Font_Name}[hnj_Options]{hnj_Font_Name}
```

다른 점은 위와 같이 한자 글꼴을 잇대어 정의할 수 있다는 것이다. 이 때 반드시 한자 글꼴 추가 정의가 시작된다는 뜻으로 각괄호 []를 써주어야 한다. 아무런 옵션을 지정하지 않더라도 이 부호는 있어야 한다.

3.2 폰트 가족의 지정

한글 폰트의 폰트 가족 (font family) 은 인위적으로 지정해주어야 할 때가 있다. 여러 한글 폰트를 임의로 묶어서 하나의 폰트 가족처럼 다루는 경우이다.

한글의 폰트 가족은 *Regular/Bold/Italic* 으로 이루어진다고 가정하고 있다.⁴⁾ 폰트 가족을 지정하여 글꼴 설정을 할 때는 다음과 같이 한다.

```
\setkormainfont(<hgBd>)(<hgIt>){<Hangul>}
\setkorsansfont(<hgBd>)(<hgIt>){<Hangul>}
```

첫 두 인자를 둘러싼 괄호가 각괄호 [] 가 아니라 괄호 () 임에 주의하라.⁵⁾ 아무튼 괄호 () 로 둘러싸인 부분은 생략가능하다. 첫 괄호 인자만 있고 이탤릭 글꼴을 지정하지 않으면 기본 글꼴로 치환된다.

한글 폰트와 별도의 한자 폰트를 설정할 필요가 있을 때는 다음과 같이 한다.

```
\setkormainfont(<hgBd>)(<hgIt>){<Hangul>}(<hjBd>)(<hjIt>){<Hanja>}
\setkorsansfont(<hgBd>)(<hgIt>){<Hangul>}(<hjBd>)(<hjIt>){<Hanja>}
```

한자를 한글과 별도의 폰트로 식자하도록 하는 것은 그 번거로움을 피할 수 없지만, 한글 폰트 가운데 한자 영역이 거의 없거나 있더라도 부실한 것이 너무 많기 때문에 현재의 한글 폰트의 사정을 돌아볼 때 부득이하고 불가피하다.

두 번째 선택 인자는 이탤릭체를 나타낸다. 이 인자를 지정하지 않으면 기본 글꼴인 은 글꼴로 치환된다. 그런데 가끔, 은 글꼴이 아니라 *Regular* 폰트를 그대로 쓰고 싶을 때가 있다. 이럴 경우 간단히

```
\setkormainfont(BOLD)(*){REGULAR}
```

와 같이 (*) 로 줄여 쓸 수 있다. 즉, 아예 두번째 인자가 없으면 은 바탕 또는 은 돌음으로 치환되고, (*) 로 지시되면 *REGULAR* 글꼴을 이탤릭으로 식자한다. 만약 *itemph* 옵션이 지시되어 있다면 *REGULAR*의 *FakeSlanted* 글자가 식자될 것이다. 이 줄인 옵션은 두 번째 선택 인자, 즉 이탤릭의 선언에만 동작하며, `\setkormainfont` 와 `\setkorsansfont`에서 쓸 수 있다.

4) 원한다면 *BoldItalic*까지 포함할 수 있으나, *xoblivoir*는 *BoldItalic*을 별도로 설정하는 것을 기본으로 삼지 않았다. 지나치게 번거로울 뿐 아니라 *BoldItalic*에 특별한 의미를 부여하기 힘들었기 때문이다. 반면, *X_YTEX-ko*는 *BoldItalic*을 별도로 설정할 수 있게 하고 있으므로 혹시라도 원한다면 이것은 사용자에게 맡겨져 있다.

5) 이것은 2008/11/09 버전의 변경사항이다. 하위호환성을 제공하지 않으므로 이전의 모든 소스를 수정해야 한다.

3.3 디폴트 치환

필요한 정보가 충분히 주어지지 않으면 폰트 설정은 기본값으로 치환된다. 기본 글꼴은 은 글꼴이다. 아래 몇 가지 사용례를 보기로 하자.

```
\setkormainfont{-윤명조120}
```

위의 명령을 실행하면 Regular와 Bold, Italic이 지정한 폰트로 바뀐다. 여기 예시한 글꼴은 font family를 이루고 있지 않은 개별 글꼴이므로 굵은 글꼴에도 같은 폰트가 식자된다.

```
\setkormainfont[BoldFont=-윤명조150,ItalicFont=-윤명조120,%  
ItalicFeatures={FakeSlant={.17}}]{-윤명조120}
```

위와 같이 fontspec 문법의 옵션을 지정할 수 있다.

```
\setkormainfont(-윤명조150){-윤명조120}
```

이 명령은 Regular와 Bold만 지정된다. Italic은 디폴트인 은 바탕으로 바뀐다.

```
\setkormainfont(-윤명조150)(*){-윤명조120}
```

이 명령은 Italic에 윤명조120을 식자하게 한다. gremphe에서는 upshape와 itsshape가 똑같이 찍히게 되지만 itemph에서는 같은 글꼴이 기울어져 식자될 것이다.

```
\setkormainfont(-윤명조150)(산돌량수 L){-윤명조120}
```

이것은 Regular, Bold, Italic을 한 가족으로 묶는다. BoldItalic이 만약 필요해지면 Bold체와 동일하게 설정된다.⁶⁾

한자 글꼴을 별도로 설정해야 할 때가 있다. 한자 글꼴을 설정하기 위해서는 위의 명령을 확장하여 다음과 같이 지정한다.

```
\setkormainfont(한글돋움)(한글이탤릭){한글레귤러}%  
(한자돋움)(한자이탤릭){한자레귤러}
```

⁶⁾ BoldItalic은 위에서 설명한 (*) 줄인 옵션의 영향을 받지 않는다.

()로 둘러싸인 부분은 모두 생략가능하지만, 이것이 한자 폰트를 설정하는 것이라는 점을 알리기 위해 한자 시작 부분에 (문자를 지정해야 한다. 즉,

```
\setkormainfont(){-윤명조120}(){한컴바탕}
```

위와 같이 하면 한글 영역에 윤명조120, 한자 영역에 한컴바탕을 식자할 수 있다. 이 때 볼드를 지정하지 않지만 최소한 ()를 써주어야 한다는 점에 주의한다. 한글 폰트 정의에서 괄호 ()를 사용했다면 한자 정의 확장에도 같은 괄호 ()를 써야 한다. 한글 폰트 정의에서 각괄호 []를 사용했다면 한자 정의 확장에도 같은 괄호 []를 써야 한다. 참고로, 일부 한글 기호문자는 한자 폰트로 식자된다.

```
\setmonoscale{<scale>}
```

한글 mono 폰트를 기본값 (은 타자) 이 아닌 글꼴로 지정할 경우 *scaling*이 필요할 때가 있다. 이를 위하여 위의 명령이 정의되어 있다. 이것은 한글 및 한자 mono 글꼴에만 영향을 미친다. 영문자는 *fontspec* 옵션으로 지정하면 될 것이다.

문장 중간에서 한글 및 한자 폰트를 일부 바꾸어 써야 할 때가 있다. 이 경우에는 \TeX -ko의 `\hangulfontspec` 명령을 이용하면 매우 간편하게 이를 실현할 수 있다.

3.4 ExternalLocation

*ExternalLocation*으로 폰트를 찾으려 하는 설정은 특히 폰트 정보가 이상한 한글 폰트를 사용할 때를 위해서도 매우 중요하다. 굳이 폰트 캐싱을 할 필요가 줄어들기도 하므로, 이 방식을 선호하는 경우도 있다. *xoblivoir*에서는 “별표붙인 명령”으로 지정한다. 참고로 별표붙인 명령을 쓸 때는 *fontspec* 옵션을 지정하는 명령형식을 쓸 수 없다. 그 이유는 *fontspec* 명령 형식에서는 *ExternalLocation* 옵션을 줄 수 있기 때문이다.

```
\setkormainfont*(BOLD)(ITALIC){REGULAR}
```

\TeX Live를 포함하여 대부분의 \TeX 임플리멘테이션에서 시스템의 폰트 폴더를 *kpathsearch*로 찾을 수 있으므로 예컨대 Windows 폰트 폴더의 폰트들도 이 방식으로 호출할 수 있다. 한편, 한겨레결체의 경우 폰트 이름을 영문으로 바꾸어두고 *ExternalLocation*으로 찾아야만 사용할 수 있다.⁷⁾

⁷⁾ 즉, 폰트 이름 정보가 잘못되어 있는 글꼴이다.

한글 폰트 이름은 폰트 자체의 특성에 따라서 매우 복잡한 양상을 보인다. 이 글에서는 Windows에서 확인할 수 있는 이름을 기준으로 하고 있으나, 플랫폼에 따라서는 이름으로 폰트를 호출하기 위해서 다른 방법을 사용해야 할 수도 있다. 가장 안전한 방법은 ExternalLocation으로 폰트파일의 이름을 부르는 것이다. 이때 몇 가지 이슈가 있다.

- (가) Windows에서는 한글 폰트 파일 이름을 부를 수 없다. 모든 폰트파일 이름은 영문이어야만 한다. 그 이유는 한글 폰트파일 이름이 윈도우즈 시스템 인코딩인 CP949로 불러져야 하는데 우리가 작성하는 tex 원본 파일은 UTF-8 인코딩이므로 한글 파일 이름을 호출하는 것이 윈도우즈에서 원천적으로 불가능하기 때문이다.
- (나) 반면, 시스템 로케일이 utf-8인 매킨토시나 리눅스에서는 한글 폰트파일 이름도 ExternalLocation으로 호출할 수 있을 것이다. 리눅스에서는 모르겠으나 매킨토시에서는 이것이 가능했다.
- (다) texmf.cnf의 OSFONTDIR 변수를 수작업으로 수정해야 하는 경우가 있다. Windows의 KC2008이나 TeXLive 자체가 이것을 잘 설정하여 주므로 사용자가 신경 쓸 것이 없으나 경우에 따라 이 변수를 설정해야 할 필요가 있을 수 있다. 맥의 경우, /usr/local/texlive/2008/texmf.cnf 파일을 열어서⁸⁾ 다음 내용을 자신의 환경에 맞게 써넣으면 될 것이다.

OSFONTDIR = /Library/Fonts;~/Library/Fonts

- (라) ExternalLocation으로 폰트를 호출한다는 것은 kpathsearch를 이용한다는 것이다. 그러므로 texmf 트리 아래에 해당 폰트를 가져다두고 mktexlsr해도 그 폰트에 접근할 수 있다.

3.5 파일 이름 문제: 조금 더

폰트를 이름으로 호출하는 데 있어 결정적으로 중요한 것은 시스템 로케일이다. 즉, 한글 로케일이면 한글 폰트 이름을 호출할 수 있지만 그렇지 않으면 폰트 이름을 한글로 부를 수 없다. xoblivoir는 한글 로케일의 시스템 —한글 Windows, Mac OS X 한글 상위 등— 일 것을 기본값으로 하고 있다. 또한 각 운영체제가 인식하는 파일 이름들이 조금씩 다르다. 이것은 폰트 자체의 특성에 따르는 경우가 많으므로 폰트의 이름이나 특성을 사용자가 잘 이해하고 사용하는 도리밖에 없다.

8) /usr/local/texlive/2008/texmf/web2c/texmf.cnf를 수정해서는 안된다.

한글이 아닌 로케일의 시스템에서는 `[nowinname]` 옵션을 지정하고 폰트를 영문 이름으로 호출하면 대부분의 문제가 해결된다.

```
\documentclass[nowinname]{xoblivoir}
\setkormainfont(UnBatangBold){UnBatang}
```

예컨대 영문 상위의 매킨토시에서 다음과 같이 한글 폰트를 정의한 경우,

```
\setkormainfont(NanumMyeongjoOTF-Bold){NanumMyeongjoOTF-Regular}
```

두 번째 선택인자가 없기 때문에 이탤릭 글꼴은 기본 글꼴인 은바탕으로 치환된다. 이때, 운영체제가 한글 글꼴 이름 “은바탕”을 인식하지 못할 수가 있다.⁹⁾

해결책은 앞서 설명한 `ExternalLocation`으로 폰트를 호출하든가, 아니면 이탤릭 옵션까지 모두 채워서 선언하든가 하면 물론 된다. 그러나 좀 번거로울 수도 있는 것이다. 이런 경우에 대비하여 `[nowinname]` 옵션을 마련해 두었다. 은 글꼴의 호출명을 영문으로 하게 함으로써 한글 글꼴 이름을 인식 못하는 운영체제의 문제를 피하게 하였다.¹⁰⁾ 영문 상위의 매킨토시, 그리고 한글 로케일을 사용하지 않는 리눅스 등에서 이 옵션이 활용될 수 있을 것이다.

이 옵션은 `xetexko-xobfont`, `xetexko-var`, `memhangul-x` 등에 부여할 수도 있으므로 Beamer 문서를 작성하기 위해 `xetexko-var`를 쓰는 경우에는 `xetexko-var`의 옵션으로 지시하면 된다.

3.6 이탤릭, 기울임

한글 글꼴에 이탤릭은 없다. 그러므로 강조를 위해 기울임으로 이탤릭을 대응하는 것은 그다지 권장하지 않는다. 예를 들어

```
\setkormainfont(얇은 고딕 Bold)(궁서체){얇은 고딕}
```

으로 지정했을 때, 궁서체가 이탤릭에 해당하는 폰트로 설정된다. 궁서체는 곧은 모양으로 찍힌다.

그러나 디자인 상의 효과를 위해서나 다른 이유에서 이 서체를 기울이고 싶은 경우가 있을 것이다. 이 경우 `[itemph]` 클래스 옵션을 지정한다. 그 반대의 경우는 `[gremph]`이고 이것이 디폴트이다. 부분부분 기울이려 한다면 `fontspec` 명령을 써서 조작할 수 있으므로 별도로 명령을 만들어두거나 하지 않았다.

9) 김영욱 교수께서 이 문제를 알려주셨다.

10) 다행히, 은 글꼴은 한글 윈도우즈에서도 영문 이름으로 호출할 수도 있다. 즉, 한글 윈도우즈에서는 영문 이름과 한글 이름을 모두 사용할 수 있는 것이다. 그러나 모든 글꼴이 그렇지 않다.

3.7 따옴표

2010년 초, $\text{\texttt{X}\TeX-ko}$ 의 급격한 발전과 더불어 *xoblivoir*가 시도하던 문장부호에 관한 사항은 사실상 $\text{\texttt{X}\TeX-ko}$ 를 따르는 것으로 충분해졌다. 문장부호를 식자할 때 영문 폰트의 것, 한글 폰트의 것을 자유롭게 선택할 수 있으며 간격 문제도 글꼴 선택의 옵션으로 가능하여졌다. $\text{\texttt{X}\TeX-ko}$ 매뉴얼을 잘 읽어보기 바란다.

이 절은 더 이상 의미가 없게 되어 내용을 삭제한다.

3.8 enumerate

enumerate 패키지의 *enumerate* 아이템 항목 머리 설정은 *oblivoir*에서와 같다.

```
\begin{enumerate}[(㉠)] \tightlist
\item 첫째 항목
\item 둘째 항목
\end{enumerate}
```

(㉠) 첫째 항목

(㉡) 둘째 항목

*paralist*에서 위와 같은 방식으로 항목 머리를 설정하려면 *xob-paralist*를 로드한다.¹¹⁾

3.9 graphicx, xcolor

$\text{\texttt{X}\TeX}$ 은 *graphicx* 패키지에 대한 명시적인 호출이 없어도 *png*, *jpg*, *pdf* 그림을 잘 불러온다. *pdf* 그림을 불러오거나 색상 관련 명령을 쓰기 위해 *color* 또는 *xcolor*를 로드하려면, 옵션 설정을 *graphicx*와 똑같이 하면 된다.

```
\usepackage{graphicx}
\usepackage{xcolor}
```

3.10 참조 인용, 자동 조사

자동 조사는 $\text{\texttt{ko}\TeX}$ 에서와 동일하다. $\text{\texttt{ko}\TeX}$ 에서는 쓸 수 없었던 한글 *label*을 사용할 수 있다.

“소적 \ref{sec:폰트}를 보라.”

“소절 제 3 절을 보라.”

¹¹⁾ 물론 *paralist* 자체는 그 이전에 부르거나 *xob-paralist*가 스스로 부르도록 할 것이고 *xob-paralist* 뒤에 *paralist*만 별도로 다시 부르면 안 된다.

3.11 방점

ko.TEX에서는 `\dotemph` 명령의 방점¹²⁾을 지원했다. XeTeX-ko에서도 이 명령을 사용할 수 있게 되었고, xoblivoir에서 조금 확장했다. 기본인 `\dotemph` 외에 `\circemph` `\useremph` 두 개의 명령을 더 쓸 수 있고 마지막의 `\useremph` 명령은 하나의 선택 인자와 두 개의 인자를 요구한다. 즉,

```
\useremph[<height>]{<dotchar>}{text}
```

height는 방점으로 사용할 문자가 올라갈 길이이고 dotchar는 방점 문자이다. text는 방점을 찍을 텍스트.

제 4 절 기타

4.1 fontspec 옵션과 수학 글꼴

fontspec 패키지와 이를 확장한 mathspec 패키지를 이용하여 수학 글꼴 일부를 바꾸고자 하거나 mathpazo와 같은 수학 글꼴 세트를 적용하고자 할 경우가 있다. 이때는 다음 두 가지 조치를 해야 한다.

- (1) 클래스 옵션으로 [fontspec]을 선언한다. 이 선언으로 사용자는 fontspec을 자신의 책임 하에 로드할 수 있다. 심지어 xltextra나 mathspec과 같이 fontspec을 부르는 패키지를 별도로 로드할 수 있다.
- (2) 윗항의 fontspec 패키지 로드 후에 xetexko-xobfont 패키지를 부른다. 이 설정 이후에야 \setkormainfont와 같은 명령을 쓸 수 있게 될 것이다.

이것은 mathfont를 조절하려면 fontspec의 옵션을 별도로 정의하여 상세한 설정을 해야 할 경우가 있기 때문이다. 예컨대 mathpazo를 수학 기본 글꼴로 쓰려 한다면 다음과 같이 하는 것이 가능하다.

```
\documentclass[fontspec]{xoblivoir}
\usepackage{mathpazo}
\usepackage[math,quiet]{fontspec}
\usepackage[math,quiet,MnSymbol]{mathspec}
\setmathsf{Asana-Math}
```

¹²⁾ 가로쓰기에서는 점을 글자 위에 찍으므로 傍점(보통)이 아니라 上점(상점)이 맞겠지만 관행적으로 방점이라 불러왔다. 이 문장부호의 정확한 명칭은 “드러범표”이다.

```
\setmainfont[Ligatures=Common]{Palatino Linotype}
\usepackage{xetexko-xobfont}
```

이 예보다 간단하게 할 수 있는 것도 많다. 이 예는 설명을 위하여 보인 것일 따름이다. 위의 `mathspec` 대신 `fontspec` 패키지 문서에 나와 있듯이 `\setmathrm` 등의 명령으로 수학 폰트를 조절할 수 있다.

4.2 moreverb

이 옵션은 `pstricks`를 사용하기 위하여 `pdfmtricks`를 이용하려 할 때 필요하다. `pdfmtricks`는 `moreverb`, `graphicx`, `(x)color` 패키지가 미리 로드되어야 동작하는데, 이 중 `graphicx`와 `xcolor`는 문제가 없지만 `oblivoir(memoir)`에서 `moreverb`는 `\usepackage`로 로드하면 `memoir`의 일부 명령과 충돌한다. 이 충돌을 해결해주는 옵션이며, 이 옵션을 준 후에 `moreverb`를 별도로 로드할 필요 없다. `moreverb`는 15에서 설명하는 `preload`로 미리 로드하는 방법도 있다.

4.3 옛 한글과 세로쓰기

2010년 2월, `XYTEX-ko`의 옛 한글 조판은 거의 완전한 수준에 이르렀다. 입력을 표준에 맞는 소위 ‘첫가끝’ 코드로 하면서도 사용자 영역(PUA)에 옛 한글이 들어 있는 글꼴의 옛 한글로 mapping이 가능해졌으며, `GSUB` 옛 한글 글꼴(현재 알려진 것으로는 은바탕과 Microsoft Office 2002 Plus Pack의 옛 한글 글꼴밖에 없다)을 그대로 이용하여 옛 한글 식자가 가능하다.

이 역시 `XYTEX-ko`의 기능으로서 `xoblivoir`는 더 이상 할 일이 없어 이 절의 내용을 삭제한다.

4.4 amssymb

`amssymb` 패키지를 로드하려 시도하면 몇 가지 명령이 이미 정의되어 있다는 메시지가 나온다. 이 메시지를 줄이려면 `amssymb` 대신 `xob-amssymb`를 `usepackage` 하도록 한다.

한편, `LyX`에는 `amsmath`와 `amssymb` 패키지를 자동으로 로드하는 기능이 있다. 이 때문에 사용자가 `xob-amssymb`를 로드하려 해도 그보다 이전에 `amssymb`가 `LyX`에 의해 로드되어 의도하는 결과를 얻지 못하는 경우가 있다. 이 때를 위하여 `[amsmath]` 옵션을 마련해두었다. 이 옵션이 활성화되면 `amsmath`와 `xob-amssymb`를 `xoblivoir`가 로드해 준다.

4.5 flowfram

`fapapersize` 및 `flowfram` 패키지와 함께 쓸 때, 첫 페이지의 pdf 사이즈만이 `fapapersize` 로 지정한 것을 따라가지 않는 문제점이 있다.¹³⁾

`xoblivoir`는 `memoir`의 페이지 출력 루틴을 조금 수정하여 대부분의 경우 pdf 사이즈 충돌 문제가 해결되도록 해두었다. 그러나 어떤 경우 pdf 파일의 첫 페이지와 이후 페이지의 사이즈가 불일치하는 문제가 여전히 발생할 가능성이 있어 다음 옵션을 없애지 않았다.

`[la4paper]` 등 미리 정의된 페이지의 경우는 아무런 문제가 생기지 않는다. 그러나 `memoir` 옵션으로 지정할 수 없는 사이즈, 예컨대 `190mm×260mm` pdf를 만들고 싶을 때는 어떻게 하는가? `xoblivoir`에게 페이지 사이즈를 강제로 알려주는 방법이 있다.

```
\documentclass[<other options>,fawd=190mm,faht=260mm]{xoblivoir}
\usepackage{fapapersize}
\usefapapersize{190mm,260mm,30mm,*,40mm,*}
\usepackage{flowfram}
```

이제 첫 페이지의 사이즈도 두번째 이후의 것과 같아졌을 것이다.

4.6 preload

일부 패키지 중에는 이따금 `memoir` 클래스 자체보다 미리 로드되어야 하는 것이 있다. 대표적인 예가 아랍어를 식자할 때 빼놓을 수 없는 `bidi` 패키지이다. 이와 같이 `memoir`보다 먼저 로드해야 하는 패키지를 쓸 때 다음과 같이 한다.

```
\documentclass[preload={bidi}]{xoblivoir}
```

`moreverb`의 경우도 이렇게 하면 되므로 이제 의미가 없어진 옵션이 되었지만 종래 작성된 문서와의 호환을 위해서 없애지는 않았다.

4.7 microtype

현재까지 \LaTeX 은 \pdfTeX 과 \LuaTeX 의 `microtype` 기능을 엔진 수준에서 지원하지 않는다. 그러나 \XeTeX -ko는 `xetexko-hanging`이라는 기법을 이용하여 온점과 반점을

13) 이주호 님이 알려주셨음.

판면 밖으로 밀어냄으로써 *margin kerning*과 비슷한 효과를 줄 수 있게 하는 재미 있는 기능을 제공한다. 이 옵션은 이름은 *microtype*이지만 실은 *xetexko-hanging*을 활성화하는 역할을 한다. 이 문서가 이 기능을 활성화하여 작성되었다.

제 5 절 첨언

xoblivoir 사용이 어느 정도로 쉬운가 하면, 나는 맨처음 이 문서를 *LyX*에서 작성하여 *export*한 다음, 두 줄 정도를 지우고 폰트 설정 명령만을 써넣었다. 그래도 훌륭한 *X_YTEX* 문서가 만들어졌던 것이다.

이 글을 쓰기 시작할 때만 해도 *X_YTEX-ko*와 *xoblivoir*는 완성되어 있지 않았다. 그러나 지금은 일반적인 문서를 작성함에 있어서 불편이 없을 정도가 되었다.

돌이켜보면, 한글을 *TEX* 문서에 사용할 수 있다는 사실 자체가 신기했던 그때로부터 20여년이 흘렀다. 본격적인 한글 *TEX* 시스템들이 나오기 시작했던 1990년대 중반으로부터 헤아려도 십수년, 이 기간 동안 한글이라는 문자 체계를 식자하기 위해 지불해야 했던 엄청난 노력과 자원을 생각하면 금석지감이 없지 않다.

*LUATEX*과 *X_YTEX*이라는 유니코드 텍 엔진의 등장은, 이러한 모든 노력들을 일시에 해결해버렸다. 이제 한글 문자의 식자는 더 이상 문제가 되지 않는다. 그러나 한글 문서다운 한글 문서, 한글 문서의 타이포그래피의 완성을 위한 길은 아직도 요원하다. 단순히 “글자를 찍는” 문제가 해결되었다고 해서 모든 일이 끝난 것은 아닌 것이다. 단지 더 생산적인 문제를 더 잘 구현할 수 있는 바탕이 갖추어진 것일 뿐이라고 생각한다.

제 6 절 이 글서의 폰트 사용 설정

이 문서의 폰트 사용 설정은 다음과 같다.

```
\setmainfont[Mapping=tex-text]{Bradley Hand ITC}
\setmonofont[Scale=.85]{Consolas}
\setkormainfont(문화 궁서 Std L){문화 궁서 흐림 Std L}(){}{네이버사전}
\setkormonofont{은 펍기}
\setmonoscale{0.9}
```

제 7 절 버전 정보

1. 이 초간단 매뉴얼은 *xoblivoir 2010/02/10* 버전에 일치한다.

2. 이 초간단 매뉴얼은 xoblivoir 2008/12/03 버전에 일치한다.
3. 이 초간단 매뉴얼은 xoblivoir 2008/11/24 버전에 일치한다.
4. 이 초간단 매뉴얼은 xoblivoir 2008/11/09 버전에 일치한다.
5. 이 초간단 매뉴얼은 xoblivoir 2008/10/23 버전에 일치한다.
6. 이 초간단 매뉴얼은 xoblivoir 2008/10/22 버전에 일치한다.
7. 이 초간단 매뉴얼은 xoblivoir 2008/10/12 버전에 일치한다.
8. 이 초간단 매뉴얼은 xoblivoir 2008/10/11 버전에 일치한다.

이 매뉴얼은 Notepad++로編輯하였다. 다 좋은데 Notepad++의 KCMenuplug-in에 xelatex 實行 命令 短縮키가 없어서 不便했다. 그러던 것이 최근 새로운 단축키가 생김으로써 훨씬 편하게 작업할 수 있게 되었다.