초간단 oblivoir v3.2 사용법

x-ob-liv-oir

2023년 2월

요 약

oblivoir 클래스는 그 동안 별도의 브랜치로 개발되어 오던 xoblivoir와 oblivoir를 통합하여 완전히 동일한 클래스가 되었다. 이 문서는 oblivoir 즉 xoblivoir를 사용하는 방법을 간략히 기술한다.

차 례

제 1 절 oblivoir와 xoblivoir 3
제 2 절 oblivoir와 memhangul 4
제 3 절 한글 드라이버 4
제 4 절 옵션들 4
4.1 memoir 옵션 4
4.2 한글 드라이버의 옵션 5
4.3 oblivoir의 옵션 5
제 5 절 폰트 설정 방식에 대한 간단한 설명 9
5.1 ko.TEX-utf 엔진 9
5.2 XeTEX-ko, LuaTEX-ko 9
5.3 oblivoir의 폰트 설정 명령 (\setko... 명령군) 11
5.4 기정의 폰트 세트 17
5.5 폰트 설정 명령 (\setkor... 명령군) 18
5.6 파일 이름으로 찾기에 관한 첨언 18
5.7 이탤릭, 기울임 19

제 6 절 그밖의 사항들 20
6.1 판면 설정을 위한 fapapersize 20
6.2 enumerate 22
6.3 graphicx, xcolor 22
6.4 참조 인용, 자동 조사 22
6.5 문장부호 23
6.6 방점 23
6.7 chapter styles 24
6.8 한글 pagestyle 24
6.9 crop mark: K style 25
6.10 `\ReleaseMacros` 명령 26
6.11 oblivoirlist 26
6.12 sidefootnote와 footnotesinmargin 26
6.13 [figtabcapt] 그림과 표의 캡션 27
6.14 부록의 조판 27
제 7 절 보조 패키지 28
7.1 chaptertoc 28
7.2 mathleading 28
7.3 oblivoir-misc 29
제 8 절 HTML 제작 29
제 9 절 샘플 문서 30
제 10 절 첨언 30
제 11 절 변경 이력 30

제 1 절 oblivoir와 xoblivoir

ko.TEX 2.0 (2013/09/30)의 등장[1]으로 ko.TEX 패키지군은 이전의 텍 엔진[2]과 새로운 엔진들, pdfLATEX, XeLATEX, LuaLATEX에 모두 일관성있게 대응하도록 변모하였다. 이러한 변화에 발맞추어, 레거시 텍 엔진을 위한 oblivoir와 새로운 텍 엔진(주로 XeLATEX)을 위한 xoblivoir로 나누어져 있던 oblivoir 클래스도 체계를 정비하여 그 구별을 없애고 동작하는 엔진에 따라 동작 방식을 자동으로 대응하도록 고쳐졌다. 그러므로, 현재 oblivoir로 작성하는 문서는 (사용자가 몇 가지 주의깊게 엔진별 동작을 지정하기만 하면) 모든 텍 엔진에서 에러 없이 컴파일되고 유사한 결과를 얻을 수 있게 되었다.[3]

그 동안 oblivoir는 비교적 복잡한 길을 거쳐왔다. 대강 정리하면,

(1) HLATEX (나중의 kotex-euc) 한글을 memoir에서 쓰기 위하여 개발된 memhangul. 이 스타일은 더이상 사용할 수 없다.
(2) dhucs (현재의 kotex-utf) 유니코드 한글을 memoir에서 쓰기 위하여 개발된 memhangul-ucs
(3) memhangul-ucs를 바탕으로 memoir 클래스를 통하여 문서를 만드는 fake-article
(4) fake-article을 oblivoir로 개명
(5) XeTEX을 위한 xoblivoir
(6) xoblivoir에 LuaTEX 지원의 추가
(7) xoblivoir와 oblivoir를 통합

이와 같이 발전하여 온 것이고, 이제 oblivoir와 xoblivoir는 완전히 동일한 클래스가 되었다.

이 문서는 oblivoir의 고유한 옵션과 폰트 설정 방식에 대해서만 설명한다. 실제로 oblivoir를 이용하여 문서를 작성할 때는 다음 세 층위의 명령이 모두 사용가능하다.

(1) memoir 명령
(2) 한글 엔진(ko.TEX, XeTEX-ko, LuaTEX-ko)의 명령
(3) oblivoir 명령

이 각각의 명령에 대한 정보를 얻으려면, memoir 매뉴얼(`texdoc memman`), 한글 패키지 매뉴얼(예컨대, `texdoc kotex`, `texdoc xetexko`)을 읽어야 한다.

1) `texdoc kotex` 명령을 내리면 kotexdoc 문서를 읽을 수 있다.
2) 이른바 "레거시 텍"이라 하는 TEX, ε-TEX, pdfTEX을 가리킨다.
3) 폰트 사용 방식의 차이로 인해 "완전히 동일한" 결과를 보증하지는 않는다.

위의 두 층위의 문서에서 설명하지 않는 oblivoir에 대한 정보를 이 문서에서 얻을 수 있다.

제 2 절 oblivoir와 memhangul

memhangul은 memoir를 한글 문서 작성에 사용할 수 있게 하기 위하여 개발된 스타일 패키지이다. 원래 독립된 스타일로서 개발되고 유지되어 왔지만, 현재는 더이상 독립적인 스타일로 사용되지 않으며 oblivoir의 핵심 기능을 정의한 서브스타일로서만 유지된다. 즉 oblivoir란 memhangul을 이용하는 한글 문서작성 클래스라고 할 수 있다.

제 3 절 한글 드라이버

oblivoir는 현재 실행되는 텍 엔진의 종류에 따라 한글 식자를 위하여 다음과 같은 한글 패키지를 부른다. 이 한글 패키지들을 (편의상) oblivoir의 한글 드라이버라고 한다.

oblivoir 옵션	텍 엔진	한글 식자 패키지
no option	pdfLaTeX	ko.TeX-utf
no option	XeLaTeX	XeTeX-ko
no option	LuaLaTeX	LuaTeX-ko

2020년 이후, 이 가운데 pdfLaTeX 지원 브랜치는 더이상 개발을 진행하지 않고 이전 버전과의 호환성만을 유지한다.

제 4 절 옵션들

4.1 memoir 옵션

원칙적으로 oblivoir는 memoir의 모든 옵션을 동일한 의미로 다 받아들인다. 여기에 해당하는 것으로 다음과 같은 것이 있다.

용지 크기 a6paper, a5paper, a4paper, a3paper, b6paper, b5paper, b4paper, b3paper, mcrownvopaper, mlargecrownvopaper, mdemyvopaper, msmallroyalvopaper, dbillpaper, statementpaper, executivepaper, letterpaper, oldpaper, legalpaper, ledgerpaper, broadsheetpaper, pottvopaper, foolscapvopaper, crownvopaper, postvopaper, largecrownvopaper, largepostvopaper, smalldemyvopaper, demyvopaper, mediumvopaper, smallroyalvopaper, royalvopaper, superroyalvopaper, imperialvopaper.

본문 기본 글자 크기 9pt, 10pt, 11pt, 12pt, 14pt, 17pt, 20pt, 25pt, 30pt, 36pt, 48pt, 60pt, *pt, extrafontsizes

프린팅 옵션 twoside, oneside, onecolumn, twocolumn, openright, openleft, openany, final, draft, ms, showtrims

기타 옵션 leqno, fleqn, openbib, article, oldfontcommands

이상의 옵션의 의미와 효과에 대해서는 memoir 매뉴얼을 읽어보라.

4.2 한글 드라이버의 옵션

한글 식자를 위하여 로드되는 패키지에는 고유한 옵션들이 있다. oblivoir는 다음과 같은 옵션을 해당 한글 드라이버에 넘겨준다.

k*o*.TEX-utf hangul, hanja, nojosa, nonfrench, finemath, strictcharcheck

XHTEX-k*o* hangul, hanja,

LuaTEX-k*o* hangul, hanja, unfonts

cjk-k*o* hangul, hanja, nojosa, usedotemph, usecjkt1font. usecjkt1font 옵션은 uset1font로 입력해도 받아들인다.

이 가운데 [hangul] 옵션은 사실상 지정할 필요가 없다. hangul 옵션이 주어져도 예들들어 절 숫자의 형식 같은 것은 oblivoir 방식이 유지되기 때문이다. 그러므로 밑줄 그은 옵션만이 oblivoir에서 의미가 있다고 할 것이다.

4.3 oblivoir의 옵션

한글 드라이버를 지정하는 옵션

다음 두 옵션은 한글 드라이버를 강제로 지정하는 옵션이다. 이 옵션을 쓸 때는 오직 pdfLaTEX만을 실행한다는 의미임을 기억해두자.

cjk cjk-k*o*패키지로 한글을 표시한다. 이 패키지는 memhangul의 일부 기능과 충돌할 가능성이 있으며 memhangul은 원칙적으로 cjk-k*o*를 지원하지 않는다. 그러나 한글 표현만을 위해서라면 이 옵션으로 문서를 작성할 수 있다.

dhucs k*o*.TEX-utf로 한글을 표현한다. pdfLaTEX에서 디폴트이며 memhangul-ucs는 이 패키지를 의도하고 작성된 것이다.

polyglossia 이 옵션이 주어지면 ko.TEX을 로드하지 않는다. 그 대신 polyglossia를 로드하고 이에 의존한 한글 식자를 준비한다.

다음 옵션들은 해당하는 드라이버에서만 의미를 갖는다.

strictcharcheck ko.TEX-utf. 엄격한 문자 검사.

finemath ko.TEX-utf. 한글 간격 미세 조정.

nofinemath ko.TEX-utf. finemath 기능을 끔.

usedotemph cjk-ko. `\dotemph` 명령 사용 가능.

uset1font cjk-ko. 라틴문자도 nanumtype1으로 찍음.

interwordHWP ko.TEX-utf. 단어 간격을 조금 더 넓게 벌려준다.

interworddefault ko.TEX-utf. 단어 간격을 적당히 벌려준다.

레거시 텍과 관련된 옵션

다음 옵션은 pdfTEX 엔진 또는 ε-TEX 엔진에서만 의미를 갖는다. 이 옵션이 주어지고 새로운 텍 엔진이 운영될 때는 무시된다.

dvips oblivoir로 작성된 문서를 latex → dvips → ps2pdf 순으로 컴파일하려 할 때, 즉 pstricks를 이용할 때 이 옵션을 주어야 한다. 이 옵션을 준 문서에 대하여 pdfLATEX을 실행하면 안 된다.

romanfixed 로마 글자의 크기와 위치를 미세조정하는 옵션으로서 untype1을 쓸 때 유용하다. 현재 상황에서는 의미가 크지 않음.

여러 가지 옵션

다음은 oblivoir의 고유한 옵션들이다.

chapter `\chapter` 명령을 제대로 쓸 수 있게 해준다. 이 옵션이 없으면 oblivoir는 `\section`부터 시작하는 문서라고 간주하고 식자하지만 `\chapter`에서 에러를 내지는 않는다.

kosection `\section`에 대하여 "제"와 "절"을 찍어주도록 하는 옵션이다.

amsmath amsmath.sty와 amssymb.sty를 미리 로드해주는 옵션이다.

mathdisp v2.2 버전에서 디스플레이 수식과 본문의 간격을 oblivoir 식으로 재설정하는 것이 디폴트가 되었다. 이전에는 `adjustmath` 옵션을 부여해야 동작하던 기능이 디폴트가 된 것이다. 이 기능을 배제하고 수식과 본문의 간격을 memoir와 amsmath가 설정하는 그대로 두려면 이 옵션을 부여한다.

arabicfront `\frontmatter` 부분의 페이지 숫자를 아라비아 숫자로 찍는다. 기본값은 로만 숫자.

footnote 각주 번호와 숫자를 한국식으로 식자한다.

figtabcapt 그림과 표에 〈그림 1〉과 같은 방식으로 캡션을 단다.

gremph 글꼴 대체 강조 방식을 쓴다. 이것이 기본값이다.

itemph 기울인 글꼴 강조 방식을 쓴다.

nonfrench nonfrenchspacing.

hangulpagestyle 본문의 페이지 스타일을 hangul 양식으로 한다.

nokorean 사실상 memoir와 거의 같은 상태가 되게 한다. 즉 pdf bookmark도 만들지 않으며 한글도 찍히지 않는다.

pdfbookmark nokorean 옵션을 주면서도 북마크는 만들도록 hyperref을 로드해주는 역할을 한다. nokorean 옵션이 주어지지 않을 때는 무의미함.

10.5pt 본문 활자 크기를 10.5pt로 한다.

quotespacing quote, quotation 환경의 줄간격을 '좁은 줄간격'으로 줄인다.

nanum 나눔명조/나눔고딕 트루타입 글꼴을 기본 글꼴로 사용하도록 설정한다.

hcr 함초롬 LVT 글꼴을 기본 글꼴로 사용하도록 설정한다.

lwarp lwarp를 이용하여 HTML을 제작하는 데 필요한 설정을 활성화하는 옵션이다.

다음 옵션들은 특별한 상황에서 의미를 가지는 것이다.

lyxhyper LyX에서 문서를 작성할 때 LyX이 강제로 hyperref을 로드하는 기능과 oblivoir의 hyperref 로드 기능이 충돌하는 것을 방지하기 위한 것이다.

tocentry chapter 옵션과 같이 쓰여서 toc, lof의 엔트리를 조정해준다.

microtype 이 옵션이 지정되면 pdfTEX과 LuaTEX에서 microtype 패키지를 불러온다. 다만 XETEX에서는 문장부호 끌어내기를 위한 xetexko-hanging 스타일을 로드한다.

subfigure subfig 패키지 대신 subfigure 패키지를 쓰기 위해서 충돌이 있는 코드 하나를 수정해준다.

manualfontspec fontspec 패키지를 자동으로 로드하지 않고 사용자가 직접 설정하고자 할 때

fontspec fontspec 패키지에 넘겨줄 옵션을 지정한다.

xcolor xcolor 패키지에 넘겨줄 옵션을 지정한다.

hyperref hyperref 패키지에 넘겨줄 옵션을 지정한다.

moreverb moreverb 패키지를 사용하려 할 때. 약간의 충돌을 해결해준다.

preload `\documentclass`가 시작되기 전에 로드해야 할 패키지를 지정한다.

preloadoption preload할 때 함께 넘겨줄 옵션을 쓴다.

faht, fawd 특별히 pdf 사이즈를 조절할 필요가 있을 때 사용한다. 특히 flowfram 패키지를 위해서 필요하다.

noreserveinserts ε-TEX의 reserveinserts 확장 코드를 억제한다. 일반적으로 사용할 필요 없다.

moreroom pdfTEX에서 용량 부족으로 에러가 발생할 때 특별히 지정한다.

다음 옵션들은 현재 큰 의미를 지니지 않는 것들이다. 대부분 ko.TEX의 발전과 더불어 oblivoir에서 특별히 지정할 필요가 없어졌다.

latinquote 따옴표를 라틴 글꼴에서 찍도록 강제하는 옵션이었다. 현재는 아무런 작용도 하지 않는다.

oldhangul 옛한글 식자를 위한 옵션이었다. 현재는 이 옵션이 없어도 옛한글을 잘 처리한다.

nowinname 은글꼴을 위해서 마련된 옵션이었으나 현재는 무의미하다.

제 5 절 폰트 설정 방식에 대한 간단한 설명

5.1 ko.TEX-utf 엔진

글꼴 선택 명령 `\SetHangulFonts`, `\SetHanjaFonts`, `\SetAdhocFonts`와 더불어, gremph 옵션이 주어졌을 때 `\SetGremphFonts` 등을 사용한다. 이 명령의 의미와 용법에 대해서는 ko.TEX 사용설명서를 참고하라.

아무런 지정도 없을 경우 nanumtype1으로 식자하고, 이것은 ko.TEX-utf의 디폴트 상황과 동일하다. gremph는 ko.TEX-utf에서 `\usepackage{dhucs-gremph}`를 선언한 경우와 동일하게 동작하므로 별도로 이 스타일을 얹을 필요는 없다.

5.2 XETEX-ko, LuaTEX-ko

한글 드라이버들은 라틴 문자 폰트와 별도로 한글/한자 폰트를 지정할 수 있게 하고 있다. 한글과 라틴 문자 글꼴을 분리하지 않으려 할 경우, XETEX-ko 명령인 `\disablekoreanfonts`를 선언한다. 이렇게 하면 한글 글꼴은 라틴 문자 글꼴을 따라가게 된다. 즉, `\setmainfont` 등으로 선언된 글꼴이 한글과 라틴 문자에 동시에 식자된다.[4)]

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

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

이 명령의 옵션으로 `[Ligatures=TeX]`을 주면 ``, `---`와 같은 입력이 “, —와 같이 나타난다. 대부분의 텍 소스에는 이런 이른바 텍 리거쳐를 사용하는 것이 일반적이므로 이 옵션을 지정하는 것이 좋은데, 최신 버전의 fontspec은 이 옵션을 자동으로 붙여주며, 아래 설명할 `\setkomainfont` 등의 명령을 써도 역시 이 옵션은 자동으로 붙기 때문에 별다른 고려를 하지 않아도 좋다. 만약 `TeX Gyre Termes`를 라틴 문자 세리프 글꼴로 쓰려 한다면,

```
\setmainfont{TeX Gyre Termes}
```

으로 충분하다.

4) 이외에도 한글-라틴 문자 글꼴의 분리에 따르는 여러 가지 옵션 정의에 관련된 XETEX-ko, LuaTEX-ko 매크로가 있으므로 이에 대해서는 XETEX-ko, LuaTEX-ko 매뉴얼을 참고하라.

한글과 한자 한글 폰트는 다음 명령을 사용하여 설정한다. 자세한 사항은 XeTeX-k*o*, LuaTeX-k*o* 매뉴얼을 참고하라.

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

이밖에 특별한 명령으로 LuaTeX-k*o*에는 fallbackfont를 위한 명령이 있다.

```
\setmainfallbackfont
\setsansfallbackfont
\setmonofallbackfont
```

XeTeX-k*o*에는 이 명령이 없는 대신 `\newfontfamily\fallbackhanjafont`를 이용하여 fallback 폰트를 정의할 수 있다. 이 기능은 한자(또는 한글/한자) 글꼴에서도 찾을 수 없는 한자를 식자하기 위해서 사용된다.

여기 언급하지 못한 다양한 명령들이 제공되므로 해당 매뉴얼을 반드시 읽어보기 바란다. oblivoir에서도 XeTeX-k*o*, LuaTeX-k*o*의 폰트 설정 방식을 그대로 활용할 수 있다.

임시 폰트 교체 문장을 작성 중에 일시적으로 폰트를 교체하기 위한 명령으로

```
\adhochangulfont
\adhochanjafont
```

이 정의되어 있다. 이 명령들은 `\setmainhangulfont` 명령과 같은 방법으로 사용한다. 또한,

```
\hangulfontspec
\hanjafontspec
```

명령도 정의되어 있으며, `\fontspec` 명령이 적용되지 않는 한글과 한자 영역에 효력을 발휘하므로 필요한 대로 쓸 수 있다.

5.3 oblivoir의 폰트 설정 명령 (\setko... 명령군)

글꼴의 이름

fontspec을 이용하여 글꼴을 지정하는 데 있어서 사용자가 곤란을 겪는 문제 중의 하나가 글꼴 이름을 지정하는 것이다. 먼저 특정 폰트(여기서는 함초롬바탕LVT)의 글꼴 이름을 알아보기 위해 `otfinfo`를 사용해보자. 글꼴에 대하여 `otfinfo -i`를 실행하면 다음과 같은 결과가 나온다.

```
$ otfinfo -i HANBatang-LVT.ttf
Family:              HCR Batang LVT
Subfamily:           Regular
Full name:           HCR Batang LVT
PostScript name:     HCRBatangLVT
Mac font menu name:  HCR Batang LVT
Version:             Version 1.940; KTS Build 20140401
Unique ID:           YoonDesign: HCR Batang LVT: KTS 20140401
Description:         The Korean TeX Society has added GSUB/GPOS/vhea/vmtx tables
                     chiefly for old hangul rendering.
                     Please contact http://www.ktug.org for these issues.
Designer URL:        http://yoonfont.co.kr/
Manufacturer:        YoonDesign; The Korean TeX Society
Vendor URL:          http://yoonfont.co.kr/
Trademark:           HCR Batang is a trademark of YoonDesign.
Copyright:           Copyright (c) 2010-2013 Hancom INC(HNC). All rights reserved.
License URL:         http://yoonfont.co.kr/
License Description: YoonDesign Inc.
Vendor ID:           YDI
```

이 가운데, "Full name"과 "Postscript name"을 사용하면 된다.[5] 여기서 "Family"는 같은 글꼴 가족들이 똑같은 이름을 가진다. 예를 들어 함초롬바탕LVT Bold체는 Family가 "HCR Batang LVT"이고 Subfamily가 "Bold"로 되어 있다. 이렇게 글꼴 자체가 글꼴 가족에 대한 정보를 가지고 있으면 Family만 지정해도 자동으로 Bold 글꼴을 글꼴가족에서 찾는다 (모든 한글 글꼴이 이렇지는 않다).

예를 들면 "맑은 고딕"은 맥락에 따라 "맑은 고딕 Bold"를 글꼴 가족으로 인식한다. 맑은 고딕의 폰트 정보 중에서 이름(name)과 가족에 해당하는 부분을 보면,

```
$ otfinfo -i Malgun.ttf | grep -e "name" -e "amily"
Family:              Malgun Gothic
Subfamily:           Regular
Full name:           Malgun Gothic
```

5) 2014년 6월 현재, 대체로 XeTeX은 "Full name"을 쓰면 거의 오류없이 동작하고 LuaTeX은 Postscript name을 선호하는 듯하다. 이 사정은 `luaotfload`나 XeTeX 엔진의 개선에 따라 달라질 수 있다.

```
PostScript name:     MalgunGothicRegular

$ otfinfo -i Malgunbd.ttf | grep -e "name" -e "amily"
Family:              Malgun Gothic
Subfamily:           Bold
Full name:           Malgun Gothic Bold
PostScript name:     MalgunGothicBold
```

그러므로

```
\setkosansfont(Malgun Gothic)
```

이렇게만 지정해도 굵은 글꼴이 요구되는 곳에서는 "맑은 고딕 Bold"를 식자해준다.

폰트 파일 이름 자체를 쓸 수도 있다. 다만 이 경우는 글꼴 가족을 지정하는 것이 아니므로 Bold나 Italic subfamily를 자동으로 인식하지 못할 수 있다. 파일 이름으로 사용하려면 확장자를 붙여주면 된다. 예컨대 함초롬바탕 LVT라면 `HANBatang-LVT.ttf`를 그대로 지정할 수 있다. 굳이 `[ExternalLocation]` 옵션을 주지 않아도 확장자를 붙이는 것만으로 ExternalLocation과 동일한 결과를 얻는다.

간단한 폰트 지정

XeTeX-k*o*, LuaTeX-k*o*의 기본 한글/한자 폰트 지정 명령인

```
\setmainhangulfont, \setsanshangulfont, \setmonohangulfont
\setmainhanjafont,  \setsanshanjafont,  \setmonohanjafont
```

들이 있으나, xoblivoir에서는 약간 다른 방법(더 편리한?)의 폰트 정의가 가능하다. 이것은 다음 세 명령으로 이루어져 있다.

```
\setkomainfont[<basename>](<Regular>)(<Bold>)(<Italic>)
\setkosansfont[<basename>](<Regular>)(<Bold>)(<Italic>)
\setkomonofont[<basename>](<Regular>)(<Bold>)(<Italic>)
```

이 세 명령을 사용할 때 주의할 점은 중괄호 { }가 전혀 쓰이지 않는다는 것이다. 입력상 실수하기 쉬우므로 주의를 요한다.

이러한 폰트 정의 방식은 xoblivoir 클래스의 원래 의도인 '되도록 간단하게 필요한 것만' 사용하자는 데서 나온 것이다.

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

```
\setkomainfont(Font Name)
```

가장 기본적인 사용법은 위와 같다. 중괄호 { }가 아니라 괄호 ( )를 쓰고 있음에 주의하라.

긴 명령 이 명령의 완전한 형태는 다음과 같다.

```
\setkomainfont[<namebase>]%
              (regular)%
              (bold)%
              (italic)%
              [hangul-feature]%
              [<hanja namebase>]%
              (hanja-regular)%
              (hanja-bold)%
              (hanja-italic)%
              [hanja-feature]
```

모두 열 개의 옵션 인자가 올 수 있는데 그 가운데 적어도 하나의 괄호 옵션 인자는 반드시 있어야 한다. 그러므로 그것(첫번째 regular)은 "옵션" 인자가 아니라 그냥 인자이지만 하나만을 중괄호로 묶는 것이 오히려 코딩 실수를 증가시킬 것으로 보아서 모두 괄호 인자를 사용하도록 했다. 괄호 옵션 인자를 하나만 준다면 그것은 한글 regular 글꼴 이름으로 받아들일 것이다. 나머지 아홉 개의 옵션 인자는 생략 가능하다. 각 옵션 인자가 생략가능하기 때문에 괄호나 꺾쇠괄호 사이에 스페이스를 남기지 않도록 주의해야 한다. 예를 들어 `(fontname)␣[feature]`와 같이 적으면 스페이스 때문에 옵션 인자의 파싱에 실패할 수 있다.

기본적으로 폰트 이름은 ( ) 안에 들어간다. 그리고 미리 오는 [ ]는 이름의 공통 부분을 축약하기 위한 것이고 끝에 오는 [ ]는 속성을 추가하기 위한 것이다. 이 규칙이 두 번 반복된다고 생각하면 되겠다.

한글과 한자 이 가운데 앞의 다섯 개는 한글, 뒤의 다섯 개는 한자관련 설정이다. 그런데 예컨대

```
\setkomainfont(Fontname A)(Fontname B)
```

이렇게 코딩해서 Fontname B가 한자 글꼴이 되게 하려 해도 두 번째 괄호 옵션 인자는 한자 regular로 받아들이는 것이 아니라 한글 bold 이름으로 인식할 것이다. 따라서 한글과 한자 사이에 다음과 같이 경계를 주면

`\setkomainfont(한글 글꼴)[](한자 글꼴)`

이제 의도대로 동작한다. 다음 보기는 한글을 함초롬바탕LVT, 한자를 맑은 고딕으로 선택하는 예가 된다.

```
\setkomainfont(HCR Batang LVT)[](Malgun Gothic)
```

한자 설정은 한글 설정 이후에 잇대어 쓰는 것으로 한글 설정 방식과 완전히 똑같다. 실제로 `\setkomainfont(A)[](B)` 명령은

```
\setmainhangulfont{A}
\setmainhanjafont{B}
```

와 동일하기 때문에 굳이 한자 글꼴을 따로 지정하지 않아도 한자 자면을 가진 한글 글꼴을 설정하면 한자가 잘 표시된다. 그러나 NanumMyeongjo와 같이 한자 자면이 없는 글꼴이라면 한자 글꼴을 별도로 선언해주어야 할 것이다.

한자 폰트 부분을 선언하면 LuaTEX-k*o*의 `\hanjabyhanjafont`를 1로 만들어서 선언된 한자 글꼴이 우선적으로 사용되도록 하는 효과가 있다. 만약 이를 원하지 않는다면 `\hanjabyhanjafont 0`을 직접 선언하도록 하라.

regular, bold, italic, bolditalic 글꼴 가족이 잘 설정된 폰트를 사용한다면 bold를 굳이 지정할 필요는 없다. bold를 지정하지 않는 것은 이 인자를 주지 않는 것이다. `()`와 같이 비운 인자를 주면 bold 글꼴에 regular 글꼴이름을 사용하므로 오히려 bold 효과가 사라진다. bold 글꼴을 아예 별도로 지정할 때는 글꼴 이름을 다 적어준다. 한편, 예컨대 HCR Batang LVT에 대하여 HCR Batang LVT Bold가 볼드체 이름임을 이용하여 `\setkomainfont(HCR Batang LVT)(* Bold)`와 같이 지정하는 방법도 통한다. 함초롬바탕의 full name을 쓰지 않고 Postscript name을 쓴다면,

```
\setkomainfont(HCRBatangLVT)(*-Bold)
```

와 같이 하면 될 것이다.

한글은 이탤릭이 없지만 우사체를 쓰는 관행이 있다. oblivoir 클래스 옵션으로 `[itemph]`를 설정하면 italic을 써야 할 곳에서 기울어진 서체를 쓸 수 있다. 반면 `[gremph]`로 하면 바로 선 서체를 사용한다. `[gremph]`가 디폴트이며, 이 경우 이탤릭을 써야 할 곳에 다른 폰트를 사용하게 할 수 있다. 예를 들면

```
\setkomainfont(HCR Batang LVT)(* Bold)(NanumGothic)
```

이렇게 하면 이탤릭을 쓸 자리에 나눔고딕이 사용된다. 만약 클래스 옵션 [itemph]가 주어져 있다면 나눔고딕도 기울어진다. 이탤릭 글꼴을 아예 지정하지 않으면 regular에 지정된 글꼴을 그대로 쓴다. [itemph]라면 이탤릭 글꼴을 지정하지 않아도 될 것이고 [gremph]라면 적당한 글꼴을 적어주면 될 것이다.

bolditalic의 경우 볼드체 글꼴과 이탤릭 속성을 사용한다. 즉 [gremph]이면 bold와 bolditalic이 동일할 것이고, [itemph]라면 bold 글꼴이 기울어진 모양으로 나온다. bolditalic 폰트를 별도로 지정하려 한다면 아래에서 설명할 feature 추가 방식으로 다음과 같이 하여야 한다.

```
\setkomainfont(Regular)(Bold)(Italic)[BoldItalicFont={Fontname}]
```

base name을 사용하여 공통 부분 줄여쓰기 KoPubWorldBatang체 같은 경우, 글꼴 가족이 서로 다른 다음 세 폰트가 있다.

```
KoPubWorldBatangLight.ttf  KoPubWorldBatangLight  KoPubWorldBatang Light
KoPubWorldBatangMedium.ttf KoPubWorldBatangMedium KoPubWorldBatang Medium
KoPubWorldBatangBold.ttf   KoPubWorldBatangBold   KoPubWorldBatang Bold
```

이 글꼴은 앞부분 이름은 같지만 Regular/Bold에 대응하는 글꼴로 설정되어 있지 않다. 이러한 상황에서 앞의 같은 부분을 base name으로 지정하고 나머지 부분을 각각 써넣는 방식으로 글꼴을 지정할 수 있다.

```
\setkomainfont[KoPubWorldBatang](Light)(Bold)
```

이것은

```
\setkomainfont(KoPubWorldBatangLight)(KoPubWorldBatangBold)
```

를 줄여쓴 것으로 이해하면 된다. 이탤릭 폰트는 지정하지 않은 보기인데, 원한다면 세번째 괄호 옵션에 써넣을 수 있다. 단, 이 방법을 쓸 때는 별표(*)를 사용할 수 없다. 왜냐하면 별표를 사용하였을 때 KoPubWorldBatang*이라는 이름의 폰트를 찾으려 할 것이기 때문이다.

예를 들어, 서울시체를 본문 글꼴로 하고 바탕 글꼴용 한자는 한양해서를 쓰고 싶다면(실제 해보면 그다지 어울리지 않지만) 어떻게 할 수 있을까?

```
\setkomainfont[SeoulHangang](L)(B)[](HYhaeseo)
\setkosansfont[SeoulNamsan](L)(B)
```

이 방식에서 주의할 것은 예컨대 이탤릭 글꼴로 완전히 이름이 다른 것을 쓰고자 한다면 이런 식으로 할 수 없다는 점이다.

feature의 추가 네 번째 옵션 인자는 한글 폰트를 설정할 때 넘겨줄 feature를 지정한다.

```
\setkomainfont(HCRBatangLVT)[FakeStretch=0.95,InterHangul=-0.05em]
```

이것은 장평을 95%로 하고 자간을 5% 줄이는 예이다. 여기에 쓸 수 있는 feature에 관해서는 fontspec 설명서와 XeTeX-k*o*, LuaTeX-k*o* 설명서를 참고하라.

옛한글을 식자하려면 (지원되는 폰트에 대하여) 다음과 같이 feature 옵션을 추가하여야 할 수 있다.

```
\setkomainfont(Noto Serif CJK KR)[Script=Hangul,Renderer=OpenType]
```

다섯번째 옵션 인자는 한자의 base name이다. 네번째 없이 다섯번째만 올 수 없으므로 이것을 설정하려면

```
\setkomainfont(HCRBatangLVT)[][KoPubBatang](Light)
```

와 같이 적어도 네번째 옵션 인자를 (비우더라도) 지정해야 할 것이다.

파일 이름으로 찾기 폰트를 그 이름(full name 또는 Postscript name)으로 지정하지 않고 파일 이름으로 찾으려 할 때, 다음과 같은 방법이 있다.

먼저 옵션 인자로 `ExternalLocation`을 선언하는 방법이다. 이 때는 확장자를 지정하지 않아도 된다. 현재 fontspec은 이 방법을 지원하기는 하지만 권장하지는 않는다.

```
\setkomainfont(HANBatang-LVT)[ExternalLocation]
```

파일 이름을 그대로 적어주는 방법이 있다. 즉 확장자를 붙여서 파일 이름 자체를 지정하는 방법이다. 이것이 권장하는 방법이 되었다.

```
\setkomainfont(HANBatang-LVT.ttf)
```

파일 이름으로 호출하는 경우 별표(*)를 이용하여 이름의 공통부분을 줄여쓰는 방법을 사용할 수 없다. 그리고 하나의 글꼴군 세트에서 “이름으로 찾기”와 “파일이름으로 찾기”를 둘 다 사용할 수 없다. 즉 다음과 같이 하는 것은 오류이다.

```
\setkomainfont(HCRBatangLVT)(HANDotum-LVT.ttf)
```

그러나 한글군/한자군에 대해서는 따로 사용해도 상관없다. 다음은 한글은 Postscript 이름으로, 한자는 파일 이름으로 찾게 설정한 경우이다. 오류없이 동작한다. 그렇지만 되도록 일관성있게 쓰는 것이 좋을 것이다.

```
\setkomainfont(HCRBatangLVT)(*-Bold)[](UNI_HSR.ttf)
```

XeTeX에서만 되는 FakeBold를 한자에만 적용해보자면,

```
\setkomainfont(HCRBatangLVT)(*-Bold)[](UNI_HSR.ttf)[AutoFakeBold]
```

이렇게 하면 되는데, FakeBold는 가급적 사용하지 않는 것이 좋겠다.

`\setob... 명령` 라틴 문자 영역의 폰트를 설정하는 데는 `\setmainfont` 등이 그다지 불편하지 않기 때문에 그냥 쓰면 된다. 그러나 `\setkomainfont`의 괄호 `( )`를 이용한 폰트 지정 방식과 유사하게 쓰려 하거나, base name을 이용한 축약 기능을 라틴 문자 정의에도 쓰고 싶다면 다음처럼 해도 상관없다.

```
\setobmainfont[<base name>](<rm>)(<bf>)(<it>)[features]
```

일반적으로는 거의 의미없는 명령이나 반은 재미로 마련해두었다.

5.4 기정의 폰트 세트

기정의 글꼴 옵션은 `[nanum]`과 `[hcr]`이 있다. 나눔 글꼴과 함초롬 글꼴 자체는 자신이 스스로 설치하여야 한다.

클래스 옵션으로 `[nanum]`을 지정하면 본문이 나눔명조와 나눔고딕으로 식자된다.[6)]

클래스 옵션 `[hcr]`은 함초롬 LVT 바탕, 함초롬 LVT 돋움 글꼴을 본문 글꼴로 사용하게 한다. `[Script=Hangul]`을 지정하여 옛한글도 잘 처리하게 한다.[7)]

문서의 Preamble에 `\setkomainfont` 또는 `\setmainhangulfont` 명령을 쓴 적이 없고 `[hcr]`이나 `[nanum]` 클래스 옵션도 부여하지 않는다면 문서는 은 글꼴로 식자한다. oblivoir는 은 글꼴의 자간과 어간을 조금 조절하는 것을 기본값으로 한다. 만약 은 글꼴에 아무런 조작을 가하지 않은 상태로 쓰고 싶다면 `\setkomainfont` 등의 명령으로 은 글꼴을 지정하라.

6) XeTeX이나 LuaTeX에서는 은 바탕을 기본 글꼴로 하고 있다. pdfTeX을 위한 kotex-utf의 경우는 여전히 nanumtype1이다.

7) `\setkomainfont` 명령의 인자로 HCR 계열이 지정되면 이 옵션이 자동으로 추가된다. 그밖의 폰트에서 옛한글을 처리하게 하려면(예를 들어 Malgun Gothic, UnBatang 등) feature 추가 옵션으로 `[Script=Hangul]`을 지정할 수 있다.

은 글꼴을 기본 글꼴로서 다루는 가장 중요한 이유는 이것이 TeX Live에 포함되어 배포되기 때문에 별도의 폰트 설치를 요구하지 않기 때문이며, 이 글꼴은 한글 LaTeX의 발전사와 깊이 연관되어 있는 중요한 글꼴이기 때문이기도 하다.

5.5 폰트 설정 명령 (\setkor... 명령군)

이전 버전의 oblivoir에서는 다음 세 명령을 제공하였다. 3.0버전부터 이 명령들은 사용하지 않는 것으로 본다. 다만 이전에 작성된 문서와의 호환성을 위해서 에러가 발생하지는 않게 해두었다. 이 명령군에 대해서는 더 설명하지 않는다.

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

5.6 파일 이름으로 찾기에 관한 첨언

폰트를 호출하는 이름은 앞서 설명한 `otfinfo -i`를 사용하여 확인할 수 있는 full name이나 Postscript name을 사용하는 것이 가장 좋다. 그러나 부득이한 경우 파일 이름으로 사용하는 것도 가능하다.

이 방식은 특히 폰트 정보가 이상한 한글 폰트를 사용할 때를 위해서도 필요하다. 굳이 폰트 캐싱을 할 필요가 줄어들기도 하므로, 이 방식을 선호하는 경우도 있다.

TeX Live를 포함하여 대부분의 TeX 임플리멘테이션에서 시스템의 폰트 폴더를 kpathsearch로 찾을 수 있으므로 예컨대 Windows 폰트 폴더의 폰트들도 이 방식으로 호출할 수 있다. 휴먼명조와 같은 글꼴은 ExternalLocation 방식으로 `HMKMM.TTF`를 직접 지시하는 것이 가장 안전하다.

이 때 몇 가지 이슈가 있다.

(가) Windows에서는 한글 폰트 파일 이름을 부를 수 없다. 모든 폰트 파일 이름은 라틴 문자이어야만 한다. 그 이유는 한글 폰트 파일 이름이 윈도우즈 시스템 인코딩인 CP949로 불려져야 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 폰트 파일 이름도 ExternalLocation으로 호출할 수 있을 것이다. 리눅스에서는 모르겠으나 매킨토시에서는 이것이 가능했다.

(다) texmf.cnf의 OSFONTDIR 변수를 수작업으로 수정해야 하는 경우가 있다. 이것은 폰트를 "파일 이름으로" 찾게 하기 위해 필요하다.

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

5.7 이탤릭, 기울임

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

```
\setkomainfont(Malgun Gothic)(* Bold)(Gungsuh)
```

으로 지정했을 때, 궁서체가 이탤릭에 해당하는 폰트로 설정된다. gremph가 디폴트이므로 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 `\hangulfontspec`이나 `\hanjafontspec`, 즉 `\adhochangulfont`를 이용하거나 `\addhangulfontfeature` 명령을 써서 조작할 수 있으므로 별도로 명령을 만들어두거나 하지 않았다.

제 6 절 그밖의 사항들

6.1 판면 설정을 위한 fapapersize

memoir는 geometry 패키지와는 다른 방식의 자체 판면 설정 명령을 가지고 있다. oblivoir에서도 기본적으로 memoir의 판면 설정 방식을 사용할 수 있다. 이와 더불어, fapapersize라는 패키지가 oblivoir에서 제공된다.

```
\usepackage{fapapersize}
\usefapapersize{*,*,1in,*,1in,*}
```

`\usefapapersize`는 여섯 개의 콤마로 연결된 인자를 취하는데, 첫번째, 두번째, 네번째, 여섯번째 인자를 별표(*)로 대용할 수 있다. 이 각각은

- paperwidth
- paperheight
- left margin
- right margin
- upper margin
- lower margin

이다. 만약 `\setheadfoot`이라든가 marginnote 설정이 필요하다면 `\usefapapersize` 명령 앞에 둔다. 적어도 left margin과 upper margin은 반드시 주어야 한다는 점과, 콤마 사이에 공백이 없도록 해야 한다는 점에 주의하여야 한다.

용지(stock)를 설정하려면 다음과 같이 한다.

```
\usepackage[stock]{fapapersize}
\usefastocksize{210mm,297mm}
\usefapapersize{190mm,260mm,1in,*,1in,*}
```

`[showtrims]` 옵션이 주어져 있다면 crop 선이 함께 나타날 것이다.

fapapersize에 기정의 용지 설정이 몇 가지 있다.

mum 국판. (148mm×210mm). 여백 25mm.

newmum 신국판. (154mm×225mm). 여백 25mm.

1in plain TEX에서처럼 1in 오프셋을 설정한 판면. 여백 1in.

dbl4x6 4×6 배판. (190mm×260mm). 여백 30mm.
a4 a4paper 용지에 [1in] 옵션과 같은 여백.

v3.1

geometry 패키지의 \newgeometry, \restoregeometry 명령과 유사하게 문서의 중간에 페이지 레이아웃을 변경하는 명령이 있다. 다음과 같이 사용한다.

\definefageometry 페이지 레이아웃을 정의하고 저장한다.

```
\definefageometry{default}{20mm,*,25mm,*}
```

첫 번째 인자는 이 페이지 레이아웃의 unique한 이름으로서 letter로 이루어진다.

두 번째 mandatory 인자는 \fapapersize 명령에서 페이지 사이즈에 해당하는 부분을 제외한 네 개의 길이를 쉼표로 구분하여 제시한다.

이 명령은 다음과 같이 쓸 수 있다.

```
\definefageometry{default}[before]{<layout>}[after]
```

여기 before와 after 위치에는 레이아웃을 바꾸기 전과 후에 실행할 토큰을 둘 수 있다.

\selectfageometry 정의된 페이지 레이아웃을 적용한다. 레이아웃 변경의 효과는 반드시 새로운 페이지가 시작되어야 나타나기 때문에 \newpage 또는 \clearpage를 직접 지정하여야 하나, 이 명령에 별표를 붙이면 페이지 레이아웃을 정의하기 전에 \clearpage를 실행한다. 페이지 레이아웃이 define되어 있지 않으면 안 된다.

다음 한 페이지의 레이아웃을 바꾸어본다. 다음과 같이 정의한 레이아웃이다

```
\definefageometry{test}{15mm,25mm,35mm,*}[\pagecolor{cyan!15}]
```

\evenmarginsameasodd는 짝수쪽의 마진을 홀수쪽 마진과 동일하게 설정하라는 명령이다. 페이지 레이아웃 디자인시에 필요할 수 있어서 별도로 정의해두었다.

6.2 enumerate

enumerate 패키지의 enumerate 아이템 항목 머리 설정은 다음과 같이 한다. 엔진과 무관하게 동작한다.

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

(㉠) 첫째 항목
(㉡) 둘째 항목

paralist에서 위와 같은 방식으로 항목 머리를 설정하려면 xob-paralist를 로드한다.[8)]

6.3 graphicx, xcolor

pdfLaTeX 문서는 dvi 드라이버로 어떤 것이 실행될지 모르기 때문에 graphicx 패키지의 로딩에 주의를 기울여야 한다. 즉 pdftex

```
\usepackage[<driver>]{graphicx}
```

과 같이 하는 것이 안전하다. 그러나 대부분의 경우 `<driver>`를 지정하지 않아도 된다. xe(lua)tex

XeTeX과 LuaTeX의 경우는 graphicx 패키지에 대한 명시적인 호출이 없어도 png, jpg, pdf 그림을 잘 불러온다.

6.4 참조 인용, 자동 조사

자동 조사는 ko.TeX에서와 동일하다. 한글 label은 XeTeX, LuaTeX에서 사용할 수 있다. 레거시 텍에서 label 자리에는 한글을 쓸 수 없다. pdftex

pdfTeX:

“소절 `\ref{sec:font}`를 보라.” “소절 제 5 절을 보라.”

XeTeX, LuaTeX:

“소절 `\ref{sec:폰트}`를 보라.” “소절 제 5 절을 보라.”

8) 물론 paralist 자체는 그 이전에 부르거나 xob-paralist가 스스로 부르도록 할 것이고 xob-paralist 뒤에 paralist만 별도로 다시 부르면 안 된다.

6.5 문장부호

이 패키지가 제공하는 문장부호는 다음과 같은 것이 있다.

- `\bnm`, `\snm`, `\cnm`, `\ccnm`. 각각 『제목』, 「제목」, 〈제목〉, 《제목》과 같이 식자된다.[9)]
- `\obldots`. `\obellipsis`. 각각 . . . , ……와 같이 식자된다. 한글 패키지 ko.TEX이 로드되면 `\ldots`의 모양이 …와 같이 바뀌기 때문에 영문서의 `\ldots`는 `\obldots`로 식자할 . . . 수 있다. `\obellipsis`는 행이 나뉘어지지 않는 여섯 개의 점으로 이루어진 말출임표이다.
- `\cntrdot`, `\cntrdots`. 각각 ·, ⋯로 식자한다.
- `\expldash`, `\explpunc`. 이것은 설명을 위하여 삽입하는 괘선——을 표현하기 위한 매크로이다. `\explpunc`의 사용법이 조금 특별하므로 주의하라. `\explpunc.some␣text.\␣` ——some text——와 같이 입력한다. 이 사용법이 복잡하다면 `\expldash`를 두 번 쓰는 방법이 있다.

6.6 방점

ko.TEX에서는 `\dotemph` 명령의 방점[10)]을 지원했다. XETEX-ko와 LuaTEX-ko에서도 이 명령을 사용할 수 있으며, oblivoir에서 조금 확장했다. 기본인 `\dotemph` 외에 `\circemph` `\useremph` 두 개의 명령을 더 쓸 수 있다.

`\dotemph`와 `\circemph`의 결과는 다음과 같다.

`\dotemph{우리나라} \circemph{대한민국}` 우리나라 대한민국

`\useremph` 명령은 이전 버전과 사용법이 달라졌다. 2014년 6월 이전 oblivoir에서 `\useremph`는 pdfTEX에서와 XETEX, LuaTEX에서 사용법이 서로 달랐고 하나의 선택인자와 두 개의 인자를 요구하는 등 사용법이 복잡했으나 이것이 모두 다음과 같이 간단하게 바뀌었다.

```
\useremph[<raise>][<char>]{text}
```

9) 만약 LuaTEX에서 이 부호들이 전각문자로 나타난다면 `\compressbnms`를 선언하라.

10) 가로쓰기에서는 점을 글자 위에 찍으므로 傍점이 아니라 上점이 맞겠지만 관행적으로 방점이라 불러왔다. 이 문장부호의 정확한 명칭은 "드러냄표"이다.

한 번 raise 값과 char를 주고난 후에는 그 설정이 보존된다. 즉 \useremph를 인자 없이 쓰면 이전에 설정된 것을 따라가고, 이전에 설정된 것이 없으면 \useremph와 \circemph가 동일하다.

```
\useremph{드러냄표}
\useremph[.3ex][\tiny★]{드러냄표}
다시한번 \useremph{드러냄표}
이번에는 \useremph[][\tiny+]{드러냄표}
높이조절 \useremph[0ex]{드러냄표}
```

이 코드의 결과는 다음과 같다:
드러냄표 드러냄표 다시한번 드러냄표 이번에는 드러냄표 높이조절 드러냄표

편의를 위하여 \useremphstarwhite(☆)와 \useremphstarblack(★) 명령을 준비해두었다. 이것은 현재 식자하는 폰트에 해당 글자가 있으면 잘 나타난다.

```
\useremph[.5ex][\useremphstarwhite]{드러냄표}
```
드러냄표

6.7 chapter styles

최근의 memoir는 상당히 많은 chapter style을 정의하고 있다. oblivoir의 chapter style 정의 방식은 기본적으로 memoir와 동일하지만 다음 몇 가지가 다르다.

- \printchaptername 명령이 무의미하다.
- \prechapternum 명령과 \postchapternum 명령이 추가되었다.
- \hchaptertitlehead라는 명령이 사용된다. 이것은 특히 running heading과 관련이 있다.

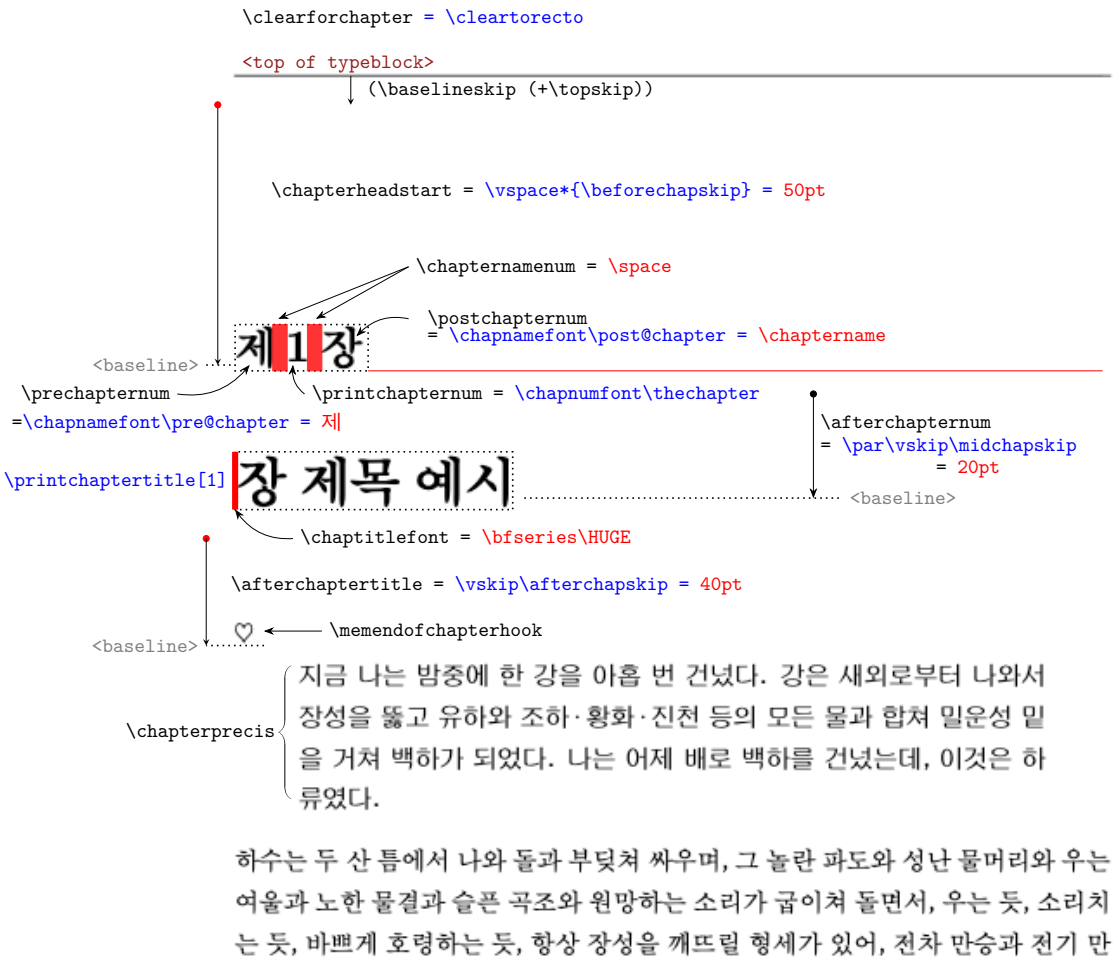
그림 1은 oblivoir의 장 표제 스타일에서 사용되는 매크로를 도시한 것이다.

KTUG 사설 저장소를 통하여 설치할 수 있는[11] ob-chapstyles라는 패키지에는 몇 가지 memoir chapter 스타일을 oblivoir화해둔 것이 있다. 이 자체를 그대로 써도 좋고 이를 자신만의 스타일을 만드는 데 참고하여도 좋을 것이다.

6.8 한글 pagestyle

oblivoir가 추가적으로 제공하는 페이지 스타일로 hangul이 있다.

11) 사설 저장소를 등록할 수 없는 상황이라면 직접 다운로드하라. http://ftp.ktug.org/KTUG/texlive/tlnet/archive/



〈그림 1〉 oblivoir의 장 표제 스타일

```
\pagestyle{hangul}
```

6.9 crop mark: K style

출판 현장에서 oblivoir를 이용하여 단행본을 제작하려 할 적에 memoir의 기본 crop mark가 너무 길어서 불평하는 경우가 있었다. 우리나라의 출판 현장에 알맞도록 조금 짧은 crop mark를 다음 명령으로 그릴 수 있게 하였다.

```
\trimKmark
```

6.10 \ReleaseMacros 명령

여러 LaTeX 패키지를 로드하여 쓰다 보면 어떤 명령이 이미 정의되었다는 에러를 만날 때가 있다. 이럴 때는 해당 패키지를 로딩하기 전에

```
\ReleaseMacros{\XeTeX,\XeLaTeX}
```

과 같이 선언하여 이미 정의된 매크로를 무력화하는 시도를 해볼 수 있다. 둘 이상의 명령을 쉼표로 구분하여 지정할 수 있다. 이 명령은 preamble에서만 쓰게 되어 있다.

6.11 oblivoirlist

나열 환경의 아이템 간 간격을 제어하기 위하여 `\oblivoirlist`와 `\oblivoirlists` 명령을 마련하였다. `\oblivoirlists`는 해당 선언 이후 모든 나열환경을 `\oblivoirlist` 간격으로 만든다.

- 배
- 사과
- 복숭아

다음 보기와 비교하여 보아라.

- 배
- 사과
- 복숭아

memoir의 `\firmlist`와 `\tightlist`는 여전히 동작한다.

12) 이 각주는 마진에 놓인다.

6.12 sidefootnote와 footnotesinmargin

oblivoir 2.0까지 `\footnotesinmargin`이 동작하지 않던 문제를 고쳤다.[12)]
`\sidefootnote`에서 발생하던 문제점도 해결하였다.[1]
memoir 설명서에 설명된 것과 동일하게 동작한다.[13)]

[1] 이 각주는 사이드 풋노트이다.

13) 상세한 것은 memoir manual을 볼 것.

6.13 [figtabcapt] 그림과 표의 캡션

클래스 옵션으로 [figtabcapt]를 지시하면 기본적으로 다음과 같은 모양으로 캡션이 바뀐다.

```
\caption{그림과 표의 캡션}
```

〈그림 1〉 그림과 표의 캡션

"그림 1" 부분을 둘러싸는 delimiter를 다음 명령으로 바꿀 수 있다. 이 가운데 \obCaptionFont는 캡션 텍스트에 영향을 미치는 명령이 아니고 라벨의 delimiter를 식자하는 데만 사용되는 폰트 명령이다.

```
\renewcommand*\obCaptionnameOpen{[}
\renewcommand*\obCaptionnameClose{]}
\obCaptionFont{\sffamily\bfseries}
```

[그림 2] 그림과 표의 캡션

이를 제외한 다른 부분, 이를테면 label과 caption text의 폰트를 바꾸는 것 등은 memoir의 해당 명령을 이용한다. 즉, \captionnamefont, \captiontitlefont 등을 이용하라는 것이다.

6.14 부록의 조판

oblivoir의 부록 만들기와 관련하여 알아두어야 하는 매크로는 다음과 같다.

\AppendixTitleToToc '부록'이라는 표제명을 toc에도 붙인다.

\AttachAppendixTitleToSecnum '부록'이라는 표제명을 section number에 붙인다.

\AppendixTitle \appendixname을 식자한다.

부록은 보통 chapter가 있는 문서에서 만드는 것이므로 장 표제(chapter 옵션의 유무와 관련없이)처럼 식자한다. 이 동작을 바꾸고 싶다면 chapterstyle과 section heading을 자신에게 필요한 대로 설정할 수 있을 것이다. 위의 몇 가지 매크로는 다른 설정 없이 default 스타일에서 나타나게 하기 위한 것이다.

부록과 관련하여 스타일이나 페이지를 디자인할 때 유념해야 할 것은, `\appendix` 또는 `appendices` 환경의 선언 이후로는 chapter와 section 등 장절표제 명령의 카운터에 `APP`가 붙는다는 점이다. 부록 chapter의 카운터는 `APPchapter`, section 카운터는 `APPsection`이다.[14)]

제 7 절 보조 패키지

7.1 chaptertoc

v3.0

chaptertoc란 장 표제면에 그 chapter에 해당하는 절(section) 이하의 목록을 만드는 것을 말한다. 이 목적을 위한 별도의 패키지가 있고 oblivoir에서 해당 패키지를 활용하는 것도 가능하다. 한편 oblivoir v3.0은 obchaptertoc라는 부수 패키지를 제공하는데 이것은 memoir의 기능만을 이용하고 다른 패키지에 의존하지 않으면서 chaptertoc를 제작하게 한 것이다. 이 기능은 오직 `\chapter`보다 높은 수준의 문서구분명령에서만 동작하며 `\section` 이하 수준의 명령에 대해서는 고려하지 않았다. 따라서 `[chapter]` 옵션이 주어진 경우에 유효하다고 하겠다.

```
\usepackage{obchaptertoc}
%%
\chaptertoc
```

이 패키지는 원래 독자적으로 개발되었던 것으로 안내 문서를 따로 가지고 있다(한국어). 문서를 읽으려면

```
# texdoc obchaptertoc
```

7.2 mathleading

oblivoir는 한국어 문서에 대하여 기본 행간을 넓혀서 조판하기 때문에 여러 줄 수식의 경우에도 그 영향을 받아서 행간격이 늘어지는 경우가 있었다. amsmath의 여러 줄 수식에 대하여 이 문제를 조절할 수 있게 하는 ob-mathleading 패키지를 포함하였다. 따로 문서가 마련되어 있으므로 이를 참조하라.

```
\usepackage{ob-mathleading}
```

14) `[nokorean]` 옵션이 주어진 경우는 그러하지 않다.

문서를 읽으려면,

```
# texdoc ob-mathleading
```

7.3 oblivoir-misc

v3.1

시험적인 기능이나 그다지 중요하지 않은 명령을 포함하고 있는 부수 패키지이다. 현재 다음 두 가지 명령이 정의되어 있다.

tikz pagenode `tikz`를 로드했을 때 `current page` 노드가 memoir의 레이아웃과 미묘하게 엇갈리는 것을 보정해준다. 이것은 oblivoir-misc를 로드하면 (tikz가 불렸을 때) 자동으로 처리한다.

tikzpagenodes tikzpagenodes 패키지가 로드되었을 때에 일부 마진 길이를 잘못 측정하는 문제를 해결해준다.

`\texthl` 한글 문자에 하이라이트해준다. 실험적인 기능이다. 하이라이트할 색상은 `\obhlcolor`, 높이와 위치는 `\obhlheight`, `\obhlraisedim`을 재정의하여 설정할 수 있다.

제 8 절 HTML 제작

lwarp를 이용하여 HTML을 제작하려면 문서에 lwarp 옵션을 주고 작성한다. 그 후,

```
lwarpmk html <file>.tex
```

명령을 실행한다. MathJax 수식을 위해서는

```
\documentclass[lwarp,lwarpoption={mathjax}]{oblivoir}
```

와 같이 옵션을 줄 수 있다.

모든 문서가 원하는 모양대로 HTML로 만들어지지 않을 것이다. 특히 사용자화하여 복잡한 환경이나 명령을 만들어 쓴 경우에는 의도하지 않은 결과가 나올 수도 있음에 주의하라. 이 클래스의 `lwarp` 옵션은 단지 `lwarpmk` 명령을 부르기 위한 준비를 도와주는 것뿐이다.

제 9 절 샘플 문서

이 작은 안내서에 더하여 간단한 oblivoir 샘플 문서를 하나 제공한다. 이 문서에서 여러 가지 memoir와 oblivoir의 기능을 살펴볼 수 있을 것이다. oblivoir-test.tex을 컴파일해 보고 소스를 검토해보시기 바란다.

제 10 절 첨언

xoblivoir 사용이 어느 정도로 쉬운가 하면, 나는 맨처음 이 문서를 LyX에서 작성하여 export한 다음, 두 줄 정도를 지우고 폰트 설정명령만을 써넣었다. 그래도 훌륭한 XeLaTeX 문서가 만들어졌던 것이다.

이 글을 쓰기 시작할 때만 해도 XeTeX-ko와 xoblivoir는 완성되어 있지 않았다. 그러나 지금은 일반적인 문서를 작성함에 있어서 불편이 없을 정도가 되었다.

돌이켜보면, 한글을 TeX 문서에 사용할 수 있다는 사실 자체가 신기했던 그 때로부터 20여년이 흘렀다. 본격적인 한글LaTeX 시스템들이 나오기 시작했던 1990년대 중반으로부터 헤아려도 십수 년, 이 기간 동안 한글이라는 문자 체계를 식자하기 위해 지불해야 했던 엄청난 노력과 자원을 생각하면 금석지감이 없지 않다.

LuaTeX과 XeTeX이라는 유니코드 텍 엔진의 등장은, 이러한 모든 노력들을 일시에 해결해버렸다. 이제 한글 문자의 식자는 더이상 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 11 절 변경 이력

2023년의 3.2 버전은 부분적인 수정에 그쳤다. 3.1에서 발생했던 대부분의 문제들은 해결되었다.
2022년의 3.1 버전은 fapapersize에 새로운 명령을 추가하고 약간의 개선된 기능을 포함하였다.
2021년의 3.0 버전은 상당히 많은 버그와 의도와 다른 동작을 수정하고 새로운 기능을 추가하였다.
2020년의 2.2 버전은 그 동안 알려진 몇 가지 버그를 수정하고 약간의 기능을 추가하는데 그쳤다.