Font: Yet Another Tutorial
과제 목록

Nova de Hi*

2020/05/30 †

이 문서에 관하여

공주대학교 문서작성 워크숍 2020에서 진행하는 "폰트: Yet Another Tutorial" 강좌의 학습자료입니다. Zoom 온라인 강의 형식임을 감안하여 (일방적일 수 있는) 발표자료 대신 요점을 추리고 과제를 제시하는 학습자료를 제공합니다.

1 학습 목표

Windows® 10 운영체제에서 TeX Live를 설치하여 XeTeX으로 LaTeX 작업을 하는 조건에서 자신의 문서에 원하는 폰트를 활용할 수 있게 한다.

(1) LaTeX의 본문 폰트 활용 방법을 이해하고 적용한다.

(2) XeTeX에서 TeX Live fontspec 패키지를 통하여 트루타입 · 오픈타입 글꼴을 활용하는 과정을 이해한다.

(3) XeTeX에서 시스템 글꼴을 활용하는 과정을 이해한다.

(4) 새로운 폰트를 설치하고 이를 문서에 적용한다.

본 강좌는 text font만을 다룰 것이다. math font에 관한 내용은 별도의 강좌를 필요로 한다.

2 레이텍의 폰트 선택 스킴 (복습)

lshort-ko를 통하여 충분히 숙지하였으리라고 본다.

family	rm		sf	tt
series	md		bf	
shape	up	it	sl	sc

- NFSS의 텍스트 폰트 선택 스킴 `encoding/family/series/shape/size` 중에서 XeTeX을 문제삼는 현 상황에서 encoding은 신경쓰지 않아도 되겠다. 어차피 모두 유니코드 폰트이기 때문.
- family, series, shape, size는 각각 독립적이다.

*KTUG

† 공주대학교 문서작성 워크숍 2020

- 각 폰트 선택을 지정하는 선언형 매크로가 있다. `\rmfamily`, `\sffamily`, `\ttfamily`, `\mdseries`, `\bfseries`, `\upshape`, `\itshape`, `\slshape`, `\scshape`. 모든 폰트 가족이 이 속성들을 다 갖추고 있는 것은 아니다. 예컨대 많은 트루타입 글꼴이 scshape가 없다.
- `\normalfont`는 `\rmfamily\mdseries\upshape\selectfont`이고 `\normalsize`는 문서 옵션에 따라 기준 크기로 정해진 사이즈가 된다.
- 인자를 갖는 명령 `\textrm`, `\textsf`, `\texttt`, `\textbf` 등 이른바 `\text...` 류 폰트 명령을 사용하는 것이 더 안전하다. 범위를 명확히 확정할 수 있기 때문이다. 선언형 폰트 선택 명령은 반드시 그 적용 범위를 중괄호로 지정해주어야 한다.
- 사이즈 명령 `\small`, `\large`는 인자를 갖는 명령형이 없다. 그리고 `\fontsize` 명령은 두 개의 인자를 취하여 임의의 사이즈로 바꾼다. 이 명령의 효력이 언제 제대로 발생하느냐를 설명하는 것은 꽤 길어지는 문제인데, 결론만 말하면 원하지 않는 결과를 얻고 싶지 않다면 남용하지 말 일이다.

과제: 적당한 영어 단어 하나를 가장 작은 크기 `\tiny`부터 `\Huge`까지 차례로 커지도록 명령을 선언해보고, documentclass의 사이즈 옵션과 폰트 크기 지정 선언의 차이를 말해보자.

참고: LaTeX 2_ε 이전 명령 `\rm`, `\bf`, `\it` 등은 현재 사용하지 않는 것으로 되어 있다. 이십년 이상 이전에 작성된 소스를 컴파일해야 할 때는 어쩔 수 없이 memoir에 `[oldfontcommands]` 옵션을 부여하여야 하지만 결과가 원하는 대로 나오지 않을 수도 있다.

3 트루타입 · 오픈타입 글꼴과 폰트 선택 명령

rmfamily	mainfont
sffamily	sansfont
ttfamily	monofont
bfseries	BoldFont
itshape	ItalicFont
	BoldItalicFont

트루타입이나 오픈타입 글꼴은 LaTeX의 family, series, shape와 같은 것을 알지 못한다. 예를 들면 Windows의 Palatino Linotype이라는 글꼴은 다음 네 개의 파일로 이루어진 글꼴 가족이다.

```
pala.ttf
palab.ttf
palai.ttf
palabi.ttf
```

이것을 다음과 같이 NFSS 스킴과 연결짓는다.

1. 이 폰트 전체를 하나의 family에 할당한다. 즉 `\rmfamily`가 이 글꼴을 요청하도록 만든다.
2. 이탤릭 서체인 `palai.ttf`를 이 family의 `\itshape`에 할당한다.
3. 볼드 서체인 `palab.ttf`를 이 family의 `\bfseries`에 할당한다.
4. 볼드-이탤릭 서체인 `palabi.ttf`를 이 family의 `\bfseries\itshape`가 동시에 불린 경우에 할당한다.

이 전체 과정의 코드를 보면 다음과 같다.

```
\setmainfont{pala.ttf}%
     [BoldFont=palab.ttf,
      ItalicFont=palai.ttf,
      BoldItalicFont=palabi.ttf]
```

이 설정을 통해서, 본문에서 `\rmfamily`나 `\textit` 같은 명령을 쓸 때 해당하는 폰트로 식자할 수 있게 된다.

4 폰트를 가족 이름으로 부르기

앞서 본 Palatino Linotype (Windows 10 버전) 폰트는 4개의 글꼴 파일로 이루어져 있었다. 우리가 `\setmainfont` 명령을 쓸 적에 이와 같이 폰트 파일의 이름을 확장자까지 쓰는 것이 허용된다.

> 폰트 파일의 이름을 확장자까지 써서 XeTeX이 이를 인식하게 하려면 그 파일들이 '찾을 수 있는' 위치에 있어야 한다. 파일 이름으로 찾을 수 있는 폴더를 지정하기 위하여 우리는 `texmf.cnf` 파일 안에 `OSFONTDIR`라는 변수를 설정하고 파일 이름으로 찾을 폴더를 적어두었다. 거기에 적을 내용은 다음과 같았다.
>
> ```
> OSFONTDIR = $SystemRoot/Fonts;$localappdata/Microsoft/Windows/Fonts
> ```

그런데 파일 이름만이 아니라 가족 이름으로 부를 수 있다면 일일이 BoldFont니 하는 것을 적어주지 않아도 (같은 가족이니까) 간단히 찾을 수 있지 않을까? 실제로 그러하다. 그러려면 먼저 폰트의 가족 이름을 알아내어야 한다.

```
> otfinfo -a C:\Windows\Fonts\pala.ttf
```

명령줄에서 위의 명령을 내려보자. `Palatino Linotype`이라는 답을 얻었는가? 그리고 이 가족 이름이 대표 이름으로서 설정 > 글꼴 설정에서 보이는 이름이다.

> 한글 윈도우즈에서 작업하는 경우에 글꼴 설정에서 보이는 이름 중에 한글 폰트는 한글 이름으로 보일 것이다. 이 이름을 그대로 사용해도 되지만, 이 경우 다른 시스템, 예컨대 영문 윈도우즈나 리눅스, 맥 오에스 등으로 그 파일을 가져가면 비록 같은 폰트가 있어도 컴파일되지 않을 수 있다.

이제 글꼴 가족 이름으로 다음과 같이 폰트를 지정할 수 있다.

```
\setmainfont{Palatino Linotype}
```

별도로 BoldFont, ItalicFont 등을 지정하지 않아도 알아서 잘 찾아준다.

> 글꼴 가족 이름으로 폰트를 지정할 수 있게 하기 위하여 우리는 `<texlive>/2020/texmf-var/fonts/config` 폴더 아래에 `local.conf` 파일을 생성하는 작업을 행하였다. 이 때 그 파일에 적어넣은 내용은 다음과 같았다.
>
> ```
> <dir>C:/Users/<username>/AppData/Local/Microsoft/Windows/Fonts</dir>
> ```
>
> `<username>`은 사용자의 로그인 계정(즉 홈폴더) 이름이다. 좀더 엄밀히 말하면 이 일은 Windows에서는 fontconfig 라이브러리가 하는 것이고 이 라이브러리와 유틸리티가 TeX Live win32에 함께 포함되어 있다.

글꼴 가족 이름은 띄어쓰기나 대소문자를 엄격하게 존중한다.

과제: 적절한 본문 로만, 산세리프 글꼴을 선택하여 mainfont와 sansfont를 구성하여 보아라.

5 Opentype Feature란 무엇인가?

요즘 폰트는 단순히 그림과 선의 집합이 아니다. 폰트 자체가 하나의 프로그램처럼 동작하게 되어 있다. 해당 폰트가 어떤 기능을 지원하는가를 알면 그 feature를 (필요하다면) 활성화할 수 있다. 영어권 폰트에서 대표적인 것으로는 small caps, alternate figures, ligatures, ordinals 등의 기능을 부가적으로 갖추어 요청이 있으면 특정한 방식으로 출력하게 하는 것이다. 예를 들면 Palatino Linotype의 경우에 Small Capital은 다음과 같이 그 feature를 켜면 얻을 수 있다.

```
\fontspec{pala.ttf}[RawFeature=+smcp] Let me show you Small Capitals.
```

Let me show you Small Capitals.

이것을 NFSS에서처럼 `\scshape`로 얻게 하려면 옵션을 주의깊게 주면 되지만 여기서는 더 다루지 않겠다.

한글 폰트에서 특히 중요한 feature는 적어도 다음 몇 가지가 있다고 본다. 첫째는 `Script=Hangul` 옵션을 주었을 때 동작하여야 하는 ccmp (자모에서 한글을 조합하는 기능이다), vertical writing (세로쓰기)와 vertical half metrics (세로쓰기에서 문장부호 간격) 등. 현재로서 이런 feature를 가진 폰트는 실상 많지가 않아서, 유감이다.

몇 년 전에 이주호 님께서 공주대학교 문서작성 워크숍에서 Fontspec에 관한 강연을 하셨다. 그 때의 자료를 찾아보면 유명한 영문 폰트에서 opentype feature를 이용하여 멋진 식자를 하는 예를 볼 수 있다.

특정 폰트에서 사용 가능한 오픈타입 feature는 `otfinfo -f`로 확인할 수 있고, 이에 대응하는 `\fontspec` 옵션은 fontspec 매뉴얼 p. 37의 table 4와 그 다음 페이지 table 5를 참고할 수 있다.

6 영문 글꼴과 한글 글꼴을 분리하는 이유

```
\setmainfont{바탕}
```

이렇게 지정하고 문서를 작성해보자.

한글 LaTeX은 역사적으로 영문자와 한글 문자를 서로 다른 폰트로 식자하는 전통을 발전시켜 왔다. 그런데 생각해보면 만약 한 개의 폰트가 품위도 충분하고 feature도 모자람이 없다면 굳이 이를 분리할 필요가 있을까? 당연한 의문이다.

이에 대한 답은, "그런 글꼴이 없다"는 것이다. 우선 당장 한글 글꼴 중에서 Palatino Linotype처럼 regular, italic, bold, bolditalic이라는 네 종류를 제공하는 것이 전혀 없다. 게다가, 소위 11172자의 현대 한글 (음절문자)을 모두 지원하는 글꼴이 당연시된 것도 얼마 되지 않는다. 그 전에는 몇 천자 정도의 한글만 있는 글꼴이 많았다. 아는 사람이 있을지도 모르지만 "똠방각하"를 자연스럽게 식자할 수 있게 된 지도 얼마 지나지 않은 것이다.

대부분의 한글 글꼴은 1개의 파일이 1개의 가족으로 묶여 있어서 가족이랄 것도 없고, 그나마 요즘은 regular와 bold 두 개를 하나의 가족으로 배포하는 것이 조금 있는 정도이다. Source Han Serif와 같이 여러 종류의 두께(weight)를 가진 글꼴도 몇 종류 있기는 하나 이것은 그냥 두께의 차이일 뿐이고 italic과 같은 의미를 갖는 것은 존재하지 않는다.

어쨌든 이 전통에 충실하게 xetexko는 영문자 글꼴과 한글 글꼴, 심지어 한자 글꼴을 모두 분리하도록 해두고 있다. 이를 위한 명령이

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

들이며, 그 사용법은 `\setmainfont`류와 크게 다를 것이 없다. 다만 한글만을 위한 몇 가지 옵션을 추가하고 있다. 이에 대하여는 xetexko 매뉴얼에 자세하다.

과제: 영문자를 기본 글꼴(Latin Modern Roman)로 하고 한글을 나눔명조로 하여 짧은 글을 작성하여라.

7 어떤 한글 폰트가 있는가?

1. TEX Live에 두 종류의 한글 글꼴이 있다. 하나는 백묵 글꼴이고 다른 하나는 은 글꼴이다. TEX Live를 설치하면 바로 사용할 수 있다.
2. Windows의 시스템 글꼴이 있다. 바탕/굴림/돋움/궁서와 맑은 고딕. 글꼴의 품위로는 맑은 고딕이 가장 훌륭하다.
3. 인스톨러를 통해 설치하거나 다른 프로그램에서 설치한 폰트가 있을 수 있다. 한컴오피스를 깔면 시스템에 추가되는 글꼴들이 그런 것이다.
4. 인스톨러 없이 사용자가 개별적으로 추가한 폰트가 있다. Noto Serif, Noto Sans 같은 것은 이렇게 설치할 수밖에 없다.

과제: Noto Serif CJK와 Noto Sans CJK 폰트를 설치하여라. LATEX 문서에서 이 폰트를 기본 글꼴로 하여 짧은 문서를 작성하여라.

자유롭게 다운로드하여 설치 사용할 수 있는 폰트 중에, 신뢰할 만한, LATEX이 원하는 기능을 모두 갖춘 것은 다음 정도이다.

- 은 바탕. (은 글꼴 전부가 아니라 은 바탕만)
- 함초롬 LVT.
- 맑은 고딕 (바탕 등의 글꼴은 제외)
- KoPub 폰트 (한국출판인회의에서 무상으로 제공하는 폰트)
- Noto CJK 폰트 또는 Source Han 폰트

위의 글꼴들 정도가 한글 식자에 관한 필요한 조건을 거의 갖추고 있다. 나눔 글꼴은 옛한글 식자가 곤란하다는 점 때문에 제외되었고, 옛한글은 되는 나눔명조옛한글의 경우 굵은 글꼴이 없어서 탈락.

그러나 기준을 이렇게 높게 잡지 않으면 쓸 만한 글꼴이 충분히 많을 것이다.

8 Fake, Fake, Fake

한글 글꼴에서 처음 부딪친 문제는 "이탤릭체"였다. 초창기 한글 LATEX은 영문의 경우에 `\emph`하면 이탤릭으로 식자되는 것이 부러운 나머지 강제로 오른쪽으로 기울인 우사체(기울임꼴)를 `\itshape`에 대응하여 쓰는 전통이 있었다. 후에 이것은 합당하지 못하다 하여 폐기되었지만 지금도 그 잔재가 남아 있다.

아무튼 이탤릭체를 써야할 곳에 기울어진 글자나 나타나게 해달라는 것이 요구였고, 이것은 FakeSlant라는 방식으로 지원되었다. 나 자신은 싫어하지만 좌우간 *기울어지는 글자*가 찍히게 하는 것은 어려운 일이 아니다. `[AutoFakeSlant]`라는 옵션을 주는 것으로 충분하다.

강제로 두꺼운 글자를 만들기 위해 한때는 세 번 겹쳐찍기까지 시도했다. 지금은 fontspec이 FakeBold 기능을 제공하고 있다. XETEX애서는 `[AutoFakeBold]`로 될 것이다. 아마 LuaTEX에서는 안 될지도 모르지만.

솔직히 말하면 이 둘 다 보기 싫다. 웬만하면 피해야 할 일이다. 변변한 폰트가 없던 시절의 유물이다.

장평 마지막으로 소개할 fake는 FakeStretch라는 것인데 이것은 소위 '장평'이라는 것을 구현하기 위해 필요하다. 예컨대 은 신문 글꼴은 가로쓰기용 글자로는 가로가 너무 길다.

100%
유구한 역사와 전통에 빛나는 우리 대한국민

92% + scale=1.05
유구한 역사와 전통에 빛나는 우리 대한국민

자간 백묵 글꼴로 한글 문장을 만들면 다음과 같이 (바보같아) 보인다.

> 유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며,

자간을 적당히 당겨줘야겠다.

> 유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며,

어간 볼 만해졌는데 단어 사이가 너무 벙벙해서 역시 바보같다.

> 유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로 건립된 대한민국임시정부의 법통과 불의에 항거한 4·19민주이념을 계승하고, 조국의 민주개혁과 평화적 통일의 사명에 입각하여 정의·인도와 동포애로써 민족의 단결을 공고히 하고, 모든 사회적 폐습과 불의를 타파하며,

드디어 읽을 만하게 되었다. 이와 같이 폰트의 성격에 따라 장평, 자간, 어간 등을 강제로 조정해야 하는 경우가 있다. (이것은 좋은 폰트가 아닌 것이다.) 기본값을 그대로 써도 좋은 문단으로 만들어지는 폰트가 가장 바람직하다. 여기서는 다만 이런 조절이 가능하기도 하다는 것을 보였다.

과제: 위의 자간, 어간을 어떻게 구현했는지 알아내어라.

9 ad hoc 폰트

특정 위치에서 임의로 특정 폰트를 적용하려면, xetexko의 경우 라틴 문자 및 기타 문자에 대하여는 `\fontspec` 명령, 한글에 대하여는 `\adhochangulfont` 또는 `\hangulfontspec` 명령, 한자에 대하여는 `\adhochanjafont` 또는 `\hanjafontspec` 명령을 쓴다.

```
\hangulfontspec{Baekmuk Batang}
텍스트
```

과제: 필기 계열의 폰트 하나를 골라서 짧은 문장을 식자하여라.

만약 같은 폰트를 한 문서에서 두 번 이상 사용한다면 이 명령을 사용해서는 안된다. 그 때에는 main/sans/mono의 폰트 설계 속에 맞는 곳을 찾아넣든가, 그런 곳이 없는 특별한 장식 글꼴이라면 `\newfontfamily`나 `\newhangulfontfamily` 또는 임의의 사용자 명령으로 폰트를 지정하여 본문에서는 오로지 이 폰트 지정 매크로를 사용하여야 한다. 그것이 LaTeX에 적합한 폰트 지정 방법이다.

따라서, 같은 폰트를 임의 글꼴 명령으로 두 번 이상 지정한 문서를 제출하면 감점.

10 옛한글 관련 약간의 보충

현재 옛한글 식자 방법은 간단히 말하면, (1) 한글 자모로 입력하고 (2) 자모 처리의 feature를 갖춘 폰트로 이를 출력한다는 것이다.

입력의 문제는 윈도우즈의 옛한글 두벌식 키보드를 이용하면 되니까 어려울 것이 없고, 아예 알파벳으로 입력하게 하는 pmhanguljamo라는 패키지도 있다. 문제는 폰트인데, `[Script=Hangul,Renderer=OpenType]`이라는 feature 명령을 받아들이는 폰트가 있어야 하는 것이다.

앞서 언급한 함초롬 LVT, Noto Serif/Sans, 맑은 고딕, KoPub 폰트 등이 가능한 글꼴이라서 예전에 비하면 한결 사정이 나아졌다.

그런데, 예전부터 "옛한글 글꼴"이라고 우기고 있는 새바탕, 새굴림, 한컴바탕 따위는 어떻게 된 것인가?

이런 폰트의 사용자 영역에 있는 옛한글 음절 글자들이 아쉽다면 어떻게든 그것을 사용할 길이 없는 것은 아니나, 자연스럽지 않고 번거롭다.

11 과제

1. oblivoir는 `\setkomainfont`, `\setkosansfont`라는 조금 다른 방식의 폰트 지정 명령을 가지고 있다. 그러나 본질적으로 `\setmainfont`와 `\sethangulmainfont`를 부르는 것이므로 그 효과는 동일하다. xetexko의 폰트 지정 명령으로 작성한 문서를 oblivoir 폰트 지정방식으로 바꾸어보라.
2. 흔히 사용되는 fontspec 옵션인 Color, Numbers, Scale, Ligatures, Language, Script 등이 어떤 의미를 가지는지 알아보아라. 특히 `Scale=MatchLowercase`가 가져오는 효과가 무엇인가?
3. 한글과 한자 글꼴에 대하여는 CharRaise라는 옵션이 있다. 무슨 일을 하는지 알아보자.
4. 오늘 배운 내용을 바탕으로 아름다운 문서 하나를 작성하여 KTUG Wiki에 업로드하여라.