

# ksmisc

Nova De Hi

2020년 2월 5일

## 차 례

1	ksjosaref.sty . . . . .	1
2	kstextsks.sty . . . . .	1
3	kslinematters.sty . . . . .	1
4	preparefont.sty . . . . .	3
5	ksmisc.sty . . . . .	3

## 1 ksjosaref.sty

ko<sub>TeX</sub>을 사용하지 아니하는 상황에서 간이 자동조사가 동작하는 ??류 명령을 정의한다.

```
$ texdoc ksjosaref-doc.pdf
```

## 2 kstextsks.sty

KS X 1001에 정의된 기호문자를 입력할 수 있는 매크로와 한글 수학책에 가끔 쓰이는 닳음 기호를 정의한다.

```
$ texdoc kstextks-doc.pdf
```

## 3 kslinematters.sty

space, carriage return, tab 문자의 catcode를 조작하여 “보이는 대로” 출력하도록 하는 명령과 환경을 제공한다.

```
\obeythem, \disobeythem
\vobeyspaces, \vobeytabs, \vobeylines
\begin{ksobeys} ... \end{ksobeys}
```

다음은 예시이다.

```
\obeythem
가 나다
가

가나다 라
\disobeythem
```

```
가 나다
가

가나다 라
```

`\obeythem... \disobeythem`을 `ksobeys` 환경으로 쓸 수 있다. 다만 이 명령/환경은 평문단에서 정상 동작한다.

다음 패키지 옵션은 `ksobeys`의 동작을 제어한다.

**noblankline** 여러 개의 빈 줄을 하나의 빈 줄로 처리하게 한다.

**tabtosp** `tab` 문자를 `tabstop` 위치까지 이동하는 것이 아니라 4개의 스페이스 문자로 치환한다.

**extcmd** `\section` 명령과 `\subsection` 명령이 `ksobeys` 환경 안에서 동작하게 한다.

단 이 명령들은 오직 하나의 인자만을 가지는 것으로 간주한다.

```
\newlinecommand, \renewlinecommand
```

`line`의 끝까지를 인자로 받는 명령을 정의한다. 인자는 #1.

```
\newlinecommand\testline{\uline{#1}}
```

이 명령은 현재 위치에서 `\testline` 줄 끝까지를 인자로 받는다.  
여기서 줄이 바뀌었다.

이 명령은 현재 위치에서 줄 끝까지를 인자로 받는다. 여기서 줄이 바뀌었다.

## 4 preparefont.sty

```
\preparefontfile{Font_File_Name}{url}[local_saved_file_name]
```

--shell-escape을 주어야 한다. 먼저 Font\_File\_Name으로 폰트를 체크하고 만약 없으면 <url>로부터 다운로드받는다. zip이거나 직접 폰트 파일 자체를 가리키는 url이어야 한다. 폰트의 존재 유무를 체크하기 위하여 luaotfload-tool을 부르므로 이것이 실행가능해야 한다.

Windows에서는 KTUG 사설 저장소의 ktugbin 패키지 설치를 요구한다. Mac이나 linux에서는 curl과 unzip이 필요하다.

첨부된 testpreparefont.tex와 게시물 “preparefont.sty”<sup>1</sup>을 참고하라.

## 5 ksmisc.sty

```
\ks_pop_left:NN <tlvar1> <tlvar2>
\ks_pop_right:NN <tlvar1> <tlvar2>
\ks_clist_set_from_tl:NN <clistvar> <tlvar>
\ks_seq_set_from_tl:NN <seqvar> <tlvar>
```

\ks\_pop\_left:NN과 \ks\_pop\_right:NN은 <tlvar1>의 왼쪽(오른쪽) 끝의 item을 하나 꺼내어 <tlvar2>에 넣고 <tlvar1>에서 삭제한다. Expl3에는 \seq\_pop\_left:NN이 있지만 tl(token list)에 대해서는 \tl\_head:N와 \tl\_tail:N이 있는 대신 pop 명령이 없어서 만든 것이다. 이를 이용하여 tl을 간단한 stack처럼 활용할 수 있다. 그러나 이 명령은 \ks\_seq\_set\_from\_tl:NN과 \ks\_clist\_set\_from\_tl:NN을 작성하기 위한 것이기도 하다. clist와 seq의 막강한 데이터 조작 능력을 tl에 대하여 얻고자 하는 것이 이 명령의 목적이다.

다음 예를 참고하라. (\ks\_for\_loop:nnn 명령은 ksforloop.sty를 요구한다. 이것은 \ksforloop와 달리 루프 코드를 그루핑하지 않는 대신 nesting이 불가능하다.)

```
\ExplSyntaxOn
\NewDocumentCommand \test { m }
{
  \tl_set:Nn \l_tst { #1 }
  \ks_for_loop:nnn {}{\tl_count:N \l_tst }
  {
    \ks_pop_right:NN \l_tst \l_tmp
  }
}
```

---

<sup>1</sup><http://doeun.blogspot.kr/2014/07/preparefontsty-dropbox.html>.

```

        \fbox{\l_tmp}
    }
}
\ExplSyntaxOff
\test{가나다라마바사아{자차}카타파하}

```

하	파	타	카	자차	아	사	바	마	라	다	나	가
---	---	---	---	----	---	---	---	---	---	---	---	---

\ks\_clist\_set\_from\_tl:NN은 이름 그대로 token list의 각 item을 comma separated list로 만든다. \ks\_seq\_set\_from\_tl:NN도 마찬가지로 tl의 item을 sequence로 만든다. <clistvar>와 <seqvar>는 전역적(global)으로 정의된다.

```

\ExplSyntaxOn
\NewDocumentCommand \testa { m }
{
    \tl_set:Nn \l_tst { #1 }
    \ks_clist_set_from_tl:NN \g_tst_cl \l_tst
    \clist_use:Nnnn \g_tst_cl {\과~} {\과~} {~,~그리고~}
}
\ExplSyntaxOff
\testa{123456789{10}}

```

1과 2와 3과 4와 5와 6과 7과 8과 9, 그리고 10