

kswdash — 물결표와 가운데점

Nova de Hi

2025/04/27 ver 0.8.1

차례

1	서론	2
2	사용법	2
3	물결표	3
3.1	기본 명령	3
3.2	물결표의 코드와 폰트	4
3.3	사이드 효과와 보충 명령	6
3.4	kopunct와 비교	7
3.5	[보론] 물결표는 언제 사용하는가?	8
4	가운데점과 줄임표	9
4.1	간편 입력 명령	9
4.2	폰트와 문자폭	11
4.3	보충 명령	13
5	긴 줄표	14
6	pdfTeX 지원	15
7	변경이력	15
	명령어 색인	16

1 서론

우리 어문의 문장부호에서 물결표(~)는 자주 사용되는 것임에도 불구하고 이것을 식자·조판하는 것이 아주 편하지 않았다. 이 패키지는 물결표와 가운데점, 줄임표 문장부호를 간편하게 입력하고 항상 일정한 폰트로 식자하게 함으로써 언제나 동일한 출력을 얻도록 하려는 데 목적이 있다.

이 작은 패키지를 작성하게 된 것은 progress 님의 문제 제기과 토론¹에서 비롯되었다. 그 이전에 필자는 물결표와 관련된 kskorsymb나 kstextks 같은 몇 가지 해결책을 제시해온 바 있으나 이들은 문장부호의 하나로 물결표 입·출력을 처리하는 것이었다. 이 패키지의 의도는 다른 복잡한 것 없이 오직 물결표(와 가운데점)만을 다루고자 하는 것이다.

물결표의 간편 입력은 강우현(애플민트초콜릿)의 kopunct 패키지가 구현한 아이디어에서 힌트를 얻었다.

2 사용법

```
\usepackage{kswdash}
```

ko_lTEX과 함께 쓸 것을 권장한다. 레거시 엔진은 지원하지 않는다.

패키지 옵션: 물결표 폰트 선택 패키지 옵션으로 물결표의 폰트를 미리 설정할 수 있다. [wdash=]로 다음 네 가지 선택사항 중 하나를 선택한다.

- [wdash=noto]
- [wdash=hcr]
- [wdash=baekmuk]
- [wdash=un]
- [wdash=currentfont]

각각 Noto Serif CJK KR, 함초롬 바탕 LVT, 백묵 바탕, 은바탕 폰트를 가리킨다. currentfont를 지시하면 특정 폰트가 아니라 현재 활성화된 폰트로 바로 찍게 한다. 옵션이 주어지지 않으면 은 바탕([wdash=un])을 디폴트로 한다. 이 옵션은 오직 물결표에만 적용된다. 3.2절에서 상세히 다룬다.

¹<https://nonametex.imweb.me/?bmode=view&idx=161109224>

패키지 옵션: 줄표와 물결표 다른 옵션으로 [ddashtowave]가 있다. 이것은 \-- 입력을 긴 줄표가 아니라 물결표로 식자하게 한다. 이 옵션이 주어지지 않으면 \--는 \textemdash와 같다.

요약 핵심은 물결표를 \~로, 가운데뎡점을 \., 줄임표를 \.. \...로 입력하는 것이다.²

```
10\~20, 4\..19, \...           10~20, 4·19, ……
```

3 물결표

3.1 기본 명령

\~

틸데 또는 물결표를 식자하는 간편 입력 명령이다. 만약 이 명령의 직후 문자가 한글 문자 또는 숫자이거나 빈 인자({})이면 물결표를 출력한다. 그 이외에는 L^AT_EX 틸데 명령과 동일하게 작동한다.

```
가\~마, abc\~{z}, 2000\~2025 \\  
\~abc, \~n, \~., \~{.}
```

```
가~마, abc~z, 2000~2025  
ãbc, ñ,~, ~.
```

물결표 간편 입력은 그 뒤에 오는 스페이스를 무시한다. 즉 \~_a는 \~a와 동일하다. 물결표를 단독으로 찍으려면 반드시 빈 인자를 주어야 한다.

\--

5절에서 설명할 \-- 명령은 긴 줄표로 온불임표(\textemdash)를 식자한다. 만약 패키지 옵션으로 [ddashtowave]가 주어진다면 \--는 긴 줄표가 아니라 물결표를 찍는다. 이 선택사항을 둔 이유는 예컨대 페이지 범위를 나타내는

²0.5 이전 버전에서 실험하던 *로 가운데뎡점을 입력하는 방식은 0.6 이후 폐기되었다.

입력 pp.~10--20에서 백슬래시 하나만을 추가하여 en-dash를 물결표로 바꾸거나, 이렇게 입력된 물결표의 백슬래시를 제거하여 붙임표로 바꾸는 것이 때로 유용할 수 있다고 판단해서이다.

```
1996\--2025, pp.~10\--20
```

1996~2025, pp. 10~20

3.2 물결표의 코드와 폰트

앞서 소개한 토론에 의하면 “물결표” 또는 그 비슷한 문자에 대하여 표 1을 제시하고 있다.

표 1: 여러 폰트의 물결표

Font	(223C)	(301C)	(FF5E)	(007E)
GNU Unifont	~	~	~	~
lmroman10-regular.otf				~
TeX Gyre Termes				~
Noto Serif	□	□	□	~
Libertinus Serif	~	⊠	⊠	~
Un Batang	~	□	~	~
Malgun Gothic	~	□	~	~
Baekmuk batang	~		~	~
HCR Batang LVT	~	~	~	~
Noto Serif CJK KR	⊠	~	~	~
KoPub Batang	~	□	~	~
Adobe Myungjo Std	⊠	⊠	~	~
Hancom Batang	~	~	~	~

[U+007E]는 아스키 틸데 문자, [U+223C]는 수학 기호 영역의 틸데 연산자(Tilde Operator), [U+301C]는 CJK 문장 부호 영역의 물결표(Wave Dash), [U+FF5E]는 KS 기호로서 전각 및 반각 형식 영역의 전각 틸데(Full-Width

Tilde)라는 이름을 가지고 있다. 현실적으로 [U+223C]를 우리 글 “물결표”로 대응해서 써야 했던 사정에 대해 앞서 소개한 토론에서 다루어졌다.

우리는 이 가운데 특정한 모양을 가진 물결표 하나를 일관되게 사용하려 한다. 그러기 위해서 물결표를 식자하는 폰트를 고정시켜서 항상 이 폰트의 특정 문자 코드로만 찍히도록 하는 것이다. 패키지 옵션으로 물결표 폰트를 {`noto`, `hcr`, `baekmuk`, `un`} 중에서 고를 수 있다. 기본값은 은 바탕의 [U+223C]이다.

`\KSWdashFont`

물결표를 식자하는 폰트를 재지정할 수 있다.

ko.TeX 없는 문서(babel이나 polyglossia)라면 `\fontspec` 명령을, 그렇지 않다면 [U+223C]와 [U+FF5E]에 대해서는 `\hangulfontspec`, [U+301C]에 대해서는 `\hanjafontspec`를 주어서 다음 예와 같이 `\KSWdashFont` 매크로를 재정의(`\renewcommand`)한다.

```
\renewcommand*\KSWdashFont{%  
  \hangulfontspec{Noto Serif CJK KR}  
  \~{}
```



위의 명령의 결과가 “tofu(두부)”로 나타나면 정상이다. Noto Serif CJK KR에는 [U+223C] 문자가 저런 모양으로 나타난다. (폰트 이름은 no tofu인데…….)

`\KSWdashCode`

[U+007E], [U+223C], [U+301C], [U+FF5E] 네 가지 틸데(물결표) 문자 코드 중 어떤 것을 선택할지 정의할 수 있다. 유니코드 코드 포인트에 해당하는 16진수 숫자를 적어준다.

```
\renewcommand*\KSWdashFont{%  
  \hangulfontspec{Noto Serif CJK KR}  
  \renewcommand*\KSWdashCode}{FF5E}  
  \~{  
  \renewcommand*\KSWdashCode}{007E}
```

```
\~{}
```

~ ~

\kswdashcurrentfont

이 패키지의 의도가 물결표를 특정 폰트로 식자하는 것이기는 하지만 때에 따라 “현재 폰트”로 찍어야 할 경우가 없다고 할 수 없다. 다음 예시는 “특정 폰트”로 인쇄하는 기능을 억제하는 것이다. 패키지 옵션 [wdash=currentfont]로 문서 전체에 대하여 이를 적용할 수 있다.

```
\hangulfontspec{Noto Sans CJK KR}
```

```
물결표 \~{}
```

```
\renewcommand*{\KSWdashFont}{\kswdashcurrentfont}
```

```
물결표 \~{}
```

물결표~ 물결표☒

3.3 사이드 효과와 보충 명령

이렇게 함으로써, \~{ }로써 ~를 적는 것이 조금 어려워지기는 한다. 이것을 “사이드 효과”라고 하자.

틸데 문자는 (1) 다음에 오는 알파벳 한 글자와 결합하거나 (2) url 주소의 일부로 쓰이는 것인데, 우리가 변경한 결과를 (1)의 목적에 쓰는 데는 아무 문제가 없다. (2)의 경우도 \url 명령이나 \path로 표현할 때는 문제가 되지 않는다. 다음 상황은 특별한 경우이다.

```
\href{https://any.com/~user}{\sffamily
```

```
→ https://any.com/\~{ }user}
```

https://any.com/~user

이럴 때는 부득이 \textasciitilde를 쓰는 것이 좋겠다.

```
\href{https://any.com/~user}{\sffamily  
→ https://any.com/\textasciitilde{user}}
```

https://any.com/~user

\tildeorgcmd

LaTeX의 원래 \sim 명령을 이 이름으로 보존해두었다. “사이드 효과”를 피하기 위해서 또는 다른 특별한 목적으로 변경된 \sim 를 쓰지 않으려 할 때 이 매크로를 활용할 수 있다. 한글 위에 틸데를 붙이는 일은 없을 것이고 숫자 위에 틸데를 굳이 붙여야 할 때는 (\sim_2 를 쓸 수 없으므로) 이 명령을 이용하면 된다.

```
\tildeorgcmd{  
→ \tildeorgcmd{0},  
→ \txfigures  
→ \tildeorgcmd{2}
```

$\sim, \tilde{O}, \tilde{z}$

3.4 kopunct와 비교

kopunct의 틸데 문자 관련 내용을 요약하면 다음과 같다.

- (1) 틸데 명령 \sim 는 그 뒤에 빈 인자({})가 오면 물결표로, 그 이외에는 tilde로 동작한다.
- (2) 틸데 문자를 단독으로 찍어야 할 때는 `\textasciitilde`를 쓴다.
- (3) 레거시 텍 엔진에서 물결표는 Computer Modern Roman의 틸데 문자를 1.5배 확대하고 높이를 조절하여 식자한다. 이 때 배율(scale)과 높이(height, raise)를 사용자화할 수 있다.
- (4) 유니코드 텍 엔진에서는 현재 폰트의 [U+007E]를 바로 찍는다. 옵션에 의하여 [U+301C] 또는 [U+FF5E]를 선택할 수 있다.³

유니코드 텍 엔진을 기준으로 말하자면 우리가 [U+223C]를 물결표로 쓰는 것이 가장 큰 차이이다. 그밖에 \sim 뒤에 숫자나 한글이 오면 물결표가 되도록

³이 패키지의 저자는 [U+223C]를 수학 기호로 보아 우리 글의 물결표로 인정하지 않고 있다.

하는 정도의 수정을 가했다. 문자의 폭, 배율, 높이를 조절하는 기능은 따로 마련하지 않았는데 우리가 출력하는 글자는 폰트에 따라 달라질 것이므로 그런 것이 필요하다면 `\raisebox` 등을 이용하는 편이 낫겠다고 판단하였기 때문이다.

kopunct 설명서의 <예시>를 같은 방식으로 입력하여 보았다.

```

제1\~{ }2장 $\rightarrow$ 제1\~2장 \\
Espa\~no1 \\
틸데 악센트(\textasciitilde) $\rightarrow$
→ \tildeorgcmd{ }
-----
제1~2장 → 제1~2장
Español
틸데 악센트(~) → ~

```

3.5 [보론] 물결표는 언제 사용하는가?

유니코드 텍 엔진이 등장하기 전 HLaTeX을 쓰던 때 물결표를 `$$\sim$$`으로 식자하는 관행이 있었다. 이 부호는 닳음을 나타내는 수학 기호이지만 생긴 모양이 물결표와 비슷하였던 것이다. 그러나 ~은 물결표라기에는 모양이 적절치 않고 다른 문제도 있어서 지금은 권하지 않게 되었다.

우리 글에서 물결표는 “기간, 거리, 범위”를 나타내는 데 쓰는 부호로서 서양 글의 en-dash(-)에 상당한다. 2014년 개정 전에는 “~사회, 국제~”와 같이 “어떤 말의 앞이나 뒤에 들어갈 말 대신” 쓴다는 규정이 있었으나 삭제되었다. (그런 식의 용법을 금지하는 것은 아니다.)

서양 글에서 기간, 범위를 나타내기 위해 물결표를 쓰는 것은 허용되지 않지만 우리 글에서 물결표를 쓸 곳에 en-dash를 쓰는 것은 허용된다. 어문 규정에는 “붙임표’를 쓸 수 있다”고 하고 있는데 여기서 말하는 ‘붙임표’는 틀림없이 en-dash일 것이다.

9월 15일~9월 25일
 김정희(1786~1856)

서울~천안 정도는 출퇴근이 가능하다.

이번 시험의 범위는 3~78쪽입니다.

『한글 맞춤법』 「부록: 문장 부호」 17의 “붙임”에 의하여 다음도 가능하다.

9월 15일~9월 25일

김정희(1786~1856)

서울~천안 정도는 출퇴근이 가능하다.

이번 시험의 범위는 3~78쪽입니다.

4 가운데점과 줄임표

가운데점은 [U+00B7], 줄임표는 [U+2026] 문자를 인쇄한다.

이 패키지는 물결표 외에 가운데점(·)과 줄임표(…, …….)의 간편 입력을 제공한다.

입력뿐 아니라 이 매크로를 통하여 고정된 폭의 가운데점과 일정한 모양의 줄임표를 얻는 것이 목표인데, 가운데점은 폰트에 따라 문자폭이 달라서⁴ 이를 보정할 필요가 있었고 줄임표는 (특히 Noto Serif CJK KR 폰트에서) 위치가 부적절하였기에 적절한 폰트로 고정시켜 찍기 위한 것이다.

패키지 개발 초기에는 마침표가 아니라 * 별표를 이용하여 가운데점과 줄임표를 입력하도록 하였으나 0.6버전부터 마침표를 이용하는 명령\..만을 정의하였다. 별표 사용은 그 나름대로 장점이 있었지만 아무래도 키보드 상의 위치가 불편하다는 것이 단점처럼 느껴졌던 것이다. 간편 입력 매크로로 \..를 선택한 이유는 흔히 글을 쓸 때 줄임표를 여러 개의 마침표로 표현하곤 하는 습관도 영향을 미쳤다.

4.1 간편 입력 명령

```
\..  
\...
```

줄임표를 다음과 같이 입력한다.

⁴할판 인쇄에서 가운데점은 “전각” 문자로 취급되었다. 오늘날 폰트 설계자들은 한글 폰트에서도 [U+00B7]을 전각(EM)으로 디자인하는 경우가 거의 없다. 컴퓨터 조판 시대에 이 문자의 “전각”을 고집할 이유는 없으나 너무 폭이 좁으면 곤란하다고 생각한다.

세 점 줄임표 `\dots` `\ll`

여섯 점 줄임표 `\lll`

세 점 줄임표 ...

여섯 점 줄임표

`\.` `\cdotbydotorgcmd`

`\.`로 가운데점을 식자할 수 있다. `kskorsymb` 패키지의 가운데점이 `"`인 것과 비교해볼 수 있겠다. 표준 `LATEX`에서 이 명령이 문자 액센트 매크로로 쓰이기 때문에 원래의 기능을 최대한 유지하는 데 주의를 기울였다. 즉, `\.` 다음에 오는 문자가 한글이나 숫자여야 가운데점이 된다. 그 이외의 경우에는 `LATEX` 기본 명령대로 알파벳 위에 점을 찍는 명령이다. 따라서 가운데점을 단독으로 표현하려면 그 뒤에 빈 인자를 주는 것이 좋다.

시장에서 사과`\.`복숭아, 고추`\.`마늘`\.`파, 조기`\.`고등어를 샀다.`\ll`
`\.{} 3\.`1절, `\.{}a{} \.{}x{} \.{}n`
`\.5 \txfigures \cdotbydotorgcmd{5}`

시장에서 사과·복숭아, 고추·마늘·파, 조기·고등어를 샀다.
· 3·1절, à x ñ · 5 §

만약—그런 일을 상상하기 어렵지만—숫자 위에 점을 찍어야 하는 희한한 경우라면(수학식 내에서 `5($\dot{5}$)`와 같이 하는 것을 가리키는 것이 아니고 텍스트 상태의 상황. 원래 `\.`는 숫자를 인자로 취하도록 만들어진 명령이 아닌데 굳이 이렇게 하려 하는 경우를 예로 든 것이다.), 위 예시의 마지막 보기와 같이 `\cdotbydotorgcmd` 매크로를 활용하여야 한다.

스페이스 가운데점의 간편 입력 매크로는 그 뒤에 오는 스페이스를 무시한다. 즉 `\.a`와 `\.a`는 동일한 명령이다. 그러므로 가운데점 뒤에 아무런 문자도 오지 않고 입력이 종료되는 경우(환경이 끝나는 경우 등)에는 오류가 발생할 수 있다. 가운데점을 단독으로 찍으려면 빈 인자를 주어야 한다.

반면 줄임표 매크로는 그 뒤에 오는 스페이스가 유지된다. 즉 `\..`와 `\...`는 서로 다른 명령이다. 만약 세 점 줄임표 뒤에 마침표를 찍으려면 `\..{}`와 같이 해야 할 것이다. 여섯 점 줄임표 뒤의 마침표는 `\....`도 괜찮다.

가운뎃점 매크로와 줄임표 매크로가 이어지는 토큰에 대하여 다르게 반응하는 이유는 `\.`가 원래 다른 목적의 매크로였고 그 기능을 상황에 따라 유지하려 하였기 때문이다.

```

\..{ } \. a \.a \\
\.. . \..{ }. \.. a \\
\.... \....{ }.
-----
· á á
···. ···. ··· a
·····. ·····.

```

4.2 폰트와 문자폭

`\KSWdashCdotFont`
`\KSWdashEllipsisFont`

`\KSWdashCdotFont`는 가운뎃점을 식자하는 폰트이다.

`\KSWdashEllipsisFont`는 세 점 및 여섯 점 줄임표를 식자하는 데 사용하는 폰트이다.

기본값은 둘 모두 함초롬 바탕 LVT이다. `\fontspec`류 명령을 이용하여 `\renewcommand`할 수 있다.

```

\..{ } \.. \... (변경전)
\renewcommand*{\KSWdashCdotFont}%
  {\hangulfontspec{NanumMyeongjo}}
\renewcommand*{\KSWdashEllipsisFont}%
  {\hangulfontspec{UnDotum}}
\..{ } \.. \... (변경후)

```

· · · · · (변경전) · · · · · (변경후)

앞서 소개한 `\kswdashcurrentfont`로 현재 식자 중인 폰트를 지정할 수 있다.

```

\.{ } \.. \... (변경전)
\renewcommand*{\KSwdashCdotFont}{\kswdashcurrentfont}
\renewcommand*{\KSwdashEllipsisFont}{\kswdashcurrentfont}
\.{ } \.. \... (변경후)

```

· · · · · (변경전) · · · · · (변경후)

`\setcdotboxwidth`

가운데점을 찍는 박스의 폭을 설정해준다. 인자는 길이값이 온다. 디폴트는 0.67em.

```

가\.나;
\setcdotboxwidth{.5em}
다\.라;
\setcdotboxwidth{1em}
마\.바

```

가·나; 다·라; 마·바

만약 이 명령의 인자로 `natural`을 선언하면 문자 박스의 폭을 강제하는 기능을 억제한다. 즉 폰트 자체의 `width`가 적용된다. 이 선언은 전역적이므로 이전 값으로 되돌리려면 `\setcdotboxwidth`를 다시 실행해야 한다.

```

가\.나;
\setcdotboxwidth{natural}
다\.라;

```

가·나; 다·라;

줄임표는 박스 폭을 수정하지 않는다.

4.3 보충 명령

```
\kswdash_cdot  
\kswdash_cdots  
\kswdash_Cdots
```

내부적으로 `\.` 등의 간편 입력 명령은 위의 세 명령의 이명(alias)이다. 가운데 점을 지원하는 문장부호 패키지 `kopunct`에는 `\cdot`, `\cdots`, `\Cdots`가 정의되어 있고 (math mode 아닌) text 맥락에서 가운데점과 줄임표로 동작한다. 또 `oblivoir`는 `\cntrdot`, `\cntrdots`, `\obellipsis` 명령이 제공된다. 이들 각각은 문자 폭이나 폰트 등에 아무런 변경을 가하지 않는 점이 우리가 정의하는 것과 다르다.

이 패키지는 `\cdot`, `\cdots`, `\Cdots`, `\cntrdot` 등 이미 정의된 명령을 수정하지 않는다. 다음 예시에서 `\cntrdot`과 `\obellipsis`가 현재 기본 한글 폰트(Noto Serif CJK KR)로 식자되는 것을 볼 수 있다.

```
가\cntrdot 나, 마\obellipsis.\  
가\.나, 마\....
```

가·나, 마.....
가·나, 마.....

그런데 굳이 이 패키지의 가운데점이나 줄임표 정의가 필요하고(박스 폭 등의 이유로) 간편 입력이 아닌 매크로(`\cdot`등)로 입력되어 있다면 다음 예는 이 명령을 복사하여 활용하는 예이다.

`\kswdash_cdot`과 `\kswdash_cdots`는 수학 모드에서 `\cdot`, `\cdots`로 동작하게 정의되어 있다.

이때 `\ExplSyntaxOn`, `\ExplSyntaxOff`에 주의하라.

수학식에서는 $a \cdot b$ 와 같이, 문장에서는 가·나와 같이
↪ 나타난다. \\

```
\ExplSyntaxOn
```

```
\let\cdot\kswdash_cdod
```

```
\ExplSyntaxOff
```

수학식에서는 $a \cdot b$ 와 같이, 문장에서는 가·나와 같이
↪ 나타난다.

수학식에서는 $a \cdot b$ 와 같이, 문장에서는 가·나와 같이 나타난다.

수학식에서는 $a \cdot b$ 와 같이, 문장에서는 가·나와 같이 나타난다.

5 긴 줄표

```
\--
```

```
\---
```

대개 `\textendash(-)`는 `--`로, `\textemdash(—)`는 `---`로 입력한다. 클래스 `oblivoir`는 `\expldash`라는 명령을 정의하고 있고 `em-dash`보다 더 긴 explanatory long dash를 식자해준다—. 이것을 간편하게 입력하는 방법을 제공한다. `\--` — `\---` —. `\--`는 `\textemdash`(즉 `---`)와 같고 `\---`는 `\expldash`와 같다.

단, 앞서 설명한 대로 패키지 옵션으로 `[ddashtowave]`가 선언되어 있으면 `\--`는 `\textemdash` 대신 물결표(~)로 동작한다.

`\-` 하이픈 명령은 그대로 유지된다.

```
\expldash
```

만약 `\expldash`가 정의되어 있지 않은 경우에는 `\textemdash`를 이용한다. 그러므로 이 패키지를 로드하면 굳이 간편 입력이 아니라도 `\expldash`라는 명령을 쓸 수 있지만 출력되는 모양은 `oblivoir`와 달리 `\textemdash`인데 앞뒤로 약간의 간격을 준 모양 — 이다.

설명줄이란 \---{} 국어 문장 부호 규정에 있는 \---{} '줄표'를 설명적 긴 줄표로 쓰는 것을 말한다.

설명줄이란 — 국어 문장 부호 규정에 있는 — '줄표'를 설명적 긴 줄표로 쓰는 것을 말한다.

설명줄은 관용적인 것이다. 「문장 부호」 어문 규정에는 줄표를 제목 아래의 부제를 적는 데 사용하는 것만 정하고 있다.

6 pdfTeX 지원

제한적이지만 pdfTeX에서 kotex-utf를 지원한다. 컴파일 엔진이 pdfTeX이면 다음 명령들은 의미가 없다.

```
\KSWdashFont, \KSWdashCdotFont, \KSWdashEllipsisFont,  
\kswdashcurrentfont, \KSWdashCode, \setcdotboxwidth
```

패키지 옵션은 아무 것도 줄 수 없다. 그러나 \., \.., \..., \--, \---, \~ 들은 이 패키지의 의도대로 동작한다. 폰트는 항상 현재 폰트이며 물결표는 [U+223C]이다.

7 변경이력

2025/04/11. ver 0.2. 설명문서 작성.

2025/04/11. ver 0.3. '긴 줄표' 추가, \.. 추가.

2025/04/11. ver 0.4. \kswdashcurrentfont, [cdot=].

2025/04/12. ver 0.6. * 입력 방법 폐기.

2025/04/12. ver 0.6.3. 기본값을 은 바탕으로 변경.

2025/04/14. ver 0.7 wdash=, ddashtowave 옵션.

2025/04/26. ver 0.8. pdftex 지원.

명령어 색인

`*`, 9
`\-`, 14
`\--`, 3, 14, 15
`\---`, 14, 15
`\.`, 3, 9–11, 13, 15
`\..`, 3, 9, 15
`\...`, 3, 9, 15
`\~`, 3, 7, 15

`\cdot`, 13
`\cdotbydotorgcmd`, 10
`\Cdots`, 13
`\cdots`, 13
`\cntrdot`, 13
`\cntrdots`, 13

`\expldash`, 14
`\ExplSyntaxOff`, 13
`\ExplSyntaxOn`, 13

`\fontspec`, 5, 11
`\hangulfontspec`, 5
`\hanjafontspec`, 5

`\kswdash_cdots`, 13
`\kswdash_Cdots`, 13
`\kswdash_cdots`, 13
`\KSWdashCdotFont`, 11
`\KSWdashCode`, 5
`\kswdashcurrentfont`, 6, 12
`\KSWdashEllipsisFont`, 11

`\KSWdashFont`, 5

`\obellipsis`, 13

Options
 `[ddashtowave]`, 3, 14
 `[wdash=]`, 2
 `[wdash=currentfont]`, 6

Packages
 babel, 5
 HLaTeX, 8
 ko.TeX, 2, 5
 kopunct, 2, 7, 8, 13
 kotex-utf, 15
 kskorsymb, 2, 10
 kstextks, 2
 oblivoir, 13, 14
 polyglossia, 5

`\path`, 6

`\raisebox`, 8
`\renewcommand`, 5, 11
`\setcdotboxwidth`, 12
`\textasciitilde`, 6, 7
`\textemdash`, 3, 14
`\textendash`, 14
`\tildeorgcmd`, 7

`\url`, 6