

텍에서 정수, 길이, 글루

김도현 동국대학교

2018년 2월 3일

한국텍학회 정기총회 및 학술대회
국가수리과학연구소 산업수학혁신센터

정수 integer

```
2018년 11시 20만원
```

- 여기서 숫자들은 **정수**가 아니라 **문자**로서 동작한다. 즉, 판면에 식자된다.

```
\count255=-24 \char44032 \penalty0
```

- 여기서 숫자들은 정수로 인식된다. 즉, 문맥상 정수를 받아들이는 곳에서는 숫자가 정수로 인식된다.
- `\penalty`<정수>는 줄나눔·쪽나눔 가능성을 지시한다. `-10000`은 줄나눔 강제, `10000`은 줄나눔 금지

정수 표현

```
\char44032
```

- 일반적으로 10진수를 쓴다.

```
\char'126000
```

- 8진수로 표현할 수 있다.

```
\char"AC00
```

- 16진수로 표현할 수 있다. A-F는 대문자만 쓴다.

```
\char`가 \char`\가 \char`%\`
```

- 문자코드로 표현할 수 있다. 특수문자는 문자명령을 이용한다.

정수표현의 끝은 어디인가

```
\count@=-24□
```

- 공백은 정수의 끝을 지시하고는 사라진다.

```
\count@=-24\relax
```

- `\relax`는 정수가 끝났음을 명시적으로 지시한다.

```
\count@='128000
```

- `\count@`에는 10이 저장된다.

```
\def\ten{10} \count@=\ten 20년을...
```

- `\count@`에는 1020이 저장된다. `\ten` 다음의 스페이스는 입력을 읽는 과정에서 이미 사라져버렸다.

내부정수internal integer –정수 레지스터

```
\count255=-1
```

```
\countdef\count@=255 \count@=-1
```

- 정수 레지스터 255번을 `\count@`으로 접근할 수 있다.
- 레거시 텍은 0...255까지, ϵ -TeX은 32767까지, 루아텍은 65535까지 레지스터를 제공한다.

```
\newcount\foo \foo=1
```

- 레지스터 빈 자리에 자동으로 `\foo`를 할당한다.
- 사용자들은 `\countdef` 대신 `\newcount`를 이용할 것.
- 레이텍의 `\newcounter`는 이렇게 정의돼 있다:

```
\expandafter\newcount\csname c@#1\endcsname
```

내부정수 — \chardef로 정의된 토큰

```
\chardef\foo=44032 \ifnum\foo=44032
```

- 그냥 \foo를 쓰면 44032번 문자, 즉 ‘가’를 식자한다. 하지만 정수를 받는 문맥에서는 정수로 인식한다.
- 숫자로 확장되지 않아서 다음 토큰과 합쳐지는 일도 없다. 레지스터를 낭비하지도 않는다.
- 레거시 텍은 0…255까지, 지텍이나 루아텍은 1,114,111까지 지정할 수 있다.
- 연산은 할 수 없다. 따라서 상수로 사용하기에 좋다.

```
\z@=0 \@ne \tw@ \thr@@ \@m=1000 \@M=10000
```

내부정수 —정수 파라미터

```
\clubpenalty=50 \widowpenalty=50
```

```
\year=2000 \month=12 \day=31
```

```
\hyphenchar\font=45
```

```
\catcode`@=11
```

 레이텍의 `\makeatletter` 명령

```
\spacefactor=1000
```

 레이텍의 `\@` 명령

```
\tracingmacros=1
```

- 일일이 다 열거하기 어렵다. *The T_EXbook*, 271쪽 이하 참조.

정수의 변환

`\number"AC00` ⇒ 44032

`\romannumeral1750` ⇒ mdcccl

`\uppercase\expandafter{\romannumeral1750}`

⇒ MDCCL

`\the\hyphenchar\font` ⇒ 45

`\the\z@` ⇒ 0.0pt

`\number\z@` ⇒ 0

- `\z@`은 정수가 아니라 길이로 정의돼 있으나 정수 맥락에서도 쓸 수 있다.

정수의 연산

```
\advance\count@ by -1
```

```
\multiply\count@ by 2
```

```
\divide\count@ by 3
```

- 연산 결과는 -2^{31} 부터 $2^{31} - 1$ 까지여야 한다.
- ϵ -TeX의 연산 표현:

```
\count@=\numexpr(\count@-1) * 2 / 3\relax
```

- `\divide` 연산은 소수점 이하를 버리는 반면, `\numexpr` 연산의 나눗셈은 반올림을 한다.

정수 조건문

```
\ifnum\count@ < 44032
```

- 관계부호는 =, >, < 중의 하나.

```
\ifodd\count@
```

- 홀수인지 짝수인지를 판단한다.

```
\ifcase\count@0\or1\or2... \or F\else\ERR\fi
```

- `\else` 부분은 기타 상황을 처리하기 위한 것으로 없어도 상관없다.

길이 dimen

길이 표현

```
4pt 1.5ex .5\hspace 5\count@
```

- sp, pt, bp, cm, mm, in, pc, cc, dd 등 물리적 단위 외에, em, ex, 길이 레지스터 등 내부길이, 내부글루, 심지어 **내부정수**—암묵적 단위는 sp—도 단위로 쓸 수 있다.
- 단위 앞에는 정수, 소수점 표기, 내부정수가 온다.
- `\kern<dimen>`은 `<dimen>` 만큼의 공백을 삽입한다.
- 내부적으로 길이는 sp 단위의 정수와 마찬가지로:

```
\dimen0=1pt \number\dimen0 sp ⇒ 65536sp
```

내부길이 internal dimen — 길이 레지스터, 파라미터

```
\dimen0=10pt
```

```
\dimendef\dimen@=0 \dimen@=10pt
```

```
\newdimen\foo \foo=1em
```

```
\the\fontdimen2\font
```

 ⇒ 3.348pt. 현재 폰트의 어간

```
\setbox0=\hbox{가}\the\wd0
```

 ⇒ 10.287pt. 박스너비

```
\the\hsize
```

 ⇒ 307.28987pt. 판면너비

```
\the\parindent
```

 ⇒ 0.0pt. 들여쓰기

```
\the\maxdimen
```

 ⇒ 16383.99998pt = $2^{30} - 1$ sp

길이 연산 및 조건문

```
\advance\dimen@ by -10pt
```

```
\multiply\dimen@ by 3 \dimen@=3\dimen@
```

```
\divide\dimen@ by 2 \dimen@=.5\dimen@
```

- 나눗셈, 곱셈의 by 뒤에는 정수만 온다.

```
\dimen@=\dimexpr(\dimen@-10pt)*3/2\relax
```

```
\dimen@=1.5\dimexpr\dimen@-10pt\relax
```

```
\ifdim\dimen@ < 1em
```

글루 glue

글루 표현

```
0pt plus .4pt minus .2pt
```

- 늘어나거나 줄어들 수 있는 가변 공백
- 예: 줄간격, 문단간격, 단어간격, ko.T_EX에서 자간
- `\hskip<glue>`, `\vskip<glue>`는 각각 수평모드와 수직모드에서 가변 공백을 삽입한다.

```
\hbox to 10cm{가\hskip0pt plus.1pt 나}
```

- 축소는 지시한 값을 한도로 하지만, 확대는 지시한 값 이상으로 늘어날 수 있다—Underfull 발생.

```
\hskip0pt plus 1fil 가\hskip0pt plus 2fil
```

- `fil`, `fill`, `filll` 단위는 가능한 최대치의 확대·축소를 뜻한다. ‘l’이 많을수록 무한히 강력하다.

내부글루internal glue —글루 레지스터, 파라미터

```
\skip0=10pt plus3pt\relax
```

```
\skipdef\skip@=0 \skip@=10pt plus3pt\relax
```

```
\newskip\foo \foo=1em plus 1ex\relax
```

- 레이텍의 `\newlength` 정의: `\newskip#1`

```
\the\baselineskip ⇒ 14.88007pt. 줄간격(행송)
```

```
\the\parskip ⇒ 6.0pt. 문단간격
```

```
\the\leftskip ⇒ 0.0pt. 문단 왼쪽 여백
```

```
\the\rightskip ⇒ 0.0pt plus 1.0fil. 문단 오른쪽 여백
```

글루 연산 및 조건문

```
\advance\skip@ by 10pt plus4pt\relax
```

```
\multiply\skip@ by 3
```

```
\divide\skip@ by 2
```

```
\skip@=2.5\skip2
```

- <승수><단위>는 길이 표현. 따라서 `\skip@`은 plus 부분과 minus 부분이 사라진 값이 된다.

```
\glueexpr(\skip@+10pt plus4pt)*3/2\relax
```

```
\ifdim\skip@ < \skip2
```

- plus 부분과 minus 부분 제외한 원래값만 비교

글루와 줄나눔·쪽나눔

- **버려질 수 없는 아이템**—glue, kern, penalty 외의 것— 다음에 오는 글루에서 줄나눔이 일어날 수 있다.
- 글루 앞에 오는 커닝kern에서도 줄나눔이 일어날 수 있다.
- 줄나눔이 일어나면 이후의 **버려질 수 있는 아이템**은 모두 사라진다.

레이텍의 `\hspace*` 명령의 정의:

```
\vrule width\z@\nobreak\hskip#1\hskip\z@skip
```

- 명령 앞에 혹시 글루가 오더라도 `\vrule`이 장벽이 된다.
- `\nobreak`는 `\penalty10000`. 줄나눔 금지
- `\hskip#1`은 페널티 다음의 글루여서 줄나눔이 안 된다.
- 명령 뒤에 혹시 `\unskip`이 오더라도 마지막 `\hskip`만 사라진다.

직전 글루·커닝·페널티의 측정 및 제거

`A\the\lastskip` ⇒ A 3.348pt
plus 1.67232pt minus 1.11711ptA0.0pt

- 직전 아이템이 글루가 아니면 0pt를 반환한다.

`가\unskip` ⇒ 가나

`\vskip40pt \par \vskip-\lastskip`

- 외부수직모드에서는 `\unskip`을 쓸 수 없다. 대신 `\vskip-\lastskip`을 쓴다.
- `\lastkern`으로 직전 커닝의 길이를 측정한다. `\unkern`, `\kern-\lastkern`은 이것을 제거한다.
- `\lastpenalty`. `\unpenalty`.

- Donald Knuth. *The T_EXbook*. Addison-Wesley. 1986.
- Victor Eijkhout. *T_EX by Topic*. <http://eijkhout.net/texbytopic/texbytopic.html>. 2013.