



TeX 매크로

매크로 작성의 기초와 예제

남수진

2016년 11월 5일 토요일

2016 공주대학교 문서작성 워크숍 2016
공주대학교 인문사회과학관 1층 컴퓨터실 107호

A word cloud of TeX macro names on a dark background. The words are in various colors (green, yellow, orange, red) and orientations (horizontal, vertical, diagonal). The macros shown include: \number, \global, \if, \egroup, \def, \futurelet, \edef, \expandafter, \afterassignment, \let, \aftergroup, \relax, \long, \endcsname, \xdef, \the, \bgroup, \outer, \ifnum, \string, and \def.

매크로 정의

`\def<control sequence><parameter text>{<replacement text>}`

- 문서에서 여러 번 반복되어 사용되는 문구나 일련의 명령어의 나열을 하나의 명령어(control sequence)로 만든 것
- `\def`, `\gdef`, `\edef`, `\xdef`
- 라텍스의 `\newcommand`, `\newenvironment`
`\def\xvec{(x_1,\ldots,x_n)}`
`\def\row#1{(#1_1,\ldots,#1_n)}`
`\def\cs_#1.#2\par{...}`
`\cs_You_owe_$5.00.#Pay_it.\par`
- 텍의 처리 과정 중에서, **전개 과정**에서 매크로는 치환 텍스트로 전개된다.

- **입력(Input)** 파일을 줄(line) 단위로 읽어서 **토큰 리스트**를 만든다.
- **전개(Expansion)** 위의 토큰 리스트를 입력으로 받아서 전개할 수 있는 모든 토큰을 전개해서 더이상 전개 할 수 없는 토큰들로 구성된 새로운 **토큰 리스트**를 만든다.
- **실행(Execution)**
- **출력(Visual)**

입력 과정

- <return>은 공백 문자와 같다.
- 연속된 여러 개의 공백 문자는 하나의 공백 문자로 처리
- 빈 줄 다음에 \par 토큰 추가
- 명령어 다음의 공백은 무시된다.

```
{\hskip_36_ mm}
```

```
{hskip 312 612 _10 m11 m11 }2
```

토큰 리스트

```
\def\tokentwo{\iftrue_ this_ \else_ that_ \fi}  
...  
\tokentwo
```

입력 과정

tokentwo

전개 과정

`iftrue` `t`₁₂ `h`₁₂ `i`₁₂ `s`₁₂ `_`₁₀ `else` `t`₁₂ `h`₁₂ `a`₁₂ `t`₁₂
`_`₁₀ `fi`

`t`₁₂ `h`₁₂ `i`₁₂ `s`₁₂ `_`₁₀

매크로와 원시명령어는 생김새가 같이 구분하기 어렵다.

- 매크로는 **전개**의 대상
- 원시명령어는 **실행**의 대상
- 매크로와 원시명령어의 구분
 - 텍북의 찾아보기에 해당 항목에 '*'가 붙어있으면 원시명령어
 - `*\thickmuskip` (thick math space), 167–168, 274, 349, 446.
 - thin spaces, 161, 167–173, 305, 409.
 - `*\thinmuskip` (thin math space), 167–168, 274, 349, 446.
 - `\thinspace`, 5, 10, 305, 311, 352, 409.
 - `\show` 명령어로 확인

다음의 명령어들은 전개 과정에서 전개된다.

- 매크로
- 조건문 (`\if`, `\ifx`, `\ifnum`, `\ifcat`, ...)
- `\number`, `\romannumeral`
- `\string`, `\fontname`, `\jobname`, `\meaning`, `\the`
- `\csname ... \endcsname`
- `\expandafter`, `\noexpand`
- `\topmark`, `\botmark`, `\firstmark`,
`\splitfirstmark`, `\splitbotmark`
- `\input`, `\endinput`

매크로 전개

매크로가 전개되지 않는 경우

- 매크로 정의 시점, `\def`, `\gdef`, `\edef`, `\xdef`
`\def\sayhello{Hello, world}`
- let 할당문, `\let` 또는 `\futurelet`, `\let\ a=\ b`
- 파라미터 텍스트 또는 매크로 인자를 읽어 들일 때
`\def\foo#1\bar{...}`
`\foo abc \abc xyz \xyz \bar`
- `\def`, `\gdef`로 정의된 치환 텍스트를 읽어 들일 때
`\def\hello{Hello, \world}`
`\edef\hello{Hello, \world} % Error`
- `\uppercase`, `\lowercase` 를 읽어 들일 때
`\def\say{hello}`
`\uppercase{\say, abc} => hello, ABC`
`\uppercase\expandafter{\say, abc} => HELLO, ABC`

토큰 리스트

```
\def\tokentwo{\iftrue_ this_ \else_ that_ \fi}  
\def\tokenone#1{...}  
...  
\expandafter\tokenone\tokentwo
```

입력 과정

expandafter tokenone tokentwo

전개 과정

tokenone t₁₂ h₁₂ i₁₂ s₁₂ \lrcorner 10 (X)

tokenone iftrue t₁₂ h₁₂ i₁₂ s₁₂ \lrcorner 10 else t₁₂
h₁₂ a₁₂ t₁₂ \lrcorner 10 fi

그룹

- {, }
- `\bgroup`, `\egroup` `\let\bgroup={ \let\egroup=}`
- `\begingroup`, `\endgroup` (원시명령어)

```
{\bf Hello}
```

```
\bgroup \bf Hello}
```

```
\bf Hello \egroup
```

```
\begingroup \bf Hello} (X)
```

```
\def\hello{Hello}
```

```
{ \baselineskip=14pt \def\hello{Hola} \hello }
```

```
\hello
```

```
Hola Hello
```

예제: `\bold` 매크로

```
{\bf Hello world}
```

```
\bold{Hello world}
```

Programmer

```
\def\bold#1{{\bf #1}}
```

\bold 매크로

```
\bold{  
  Hello  
  
  world  
}
```

Runaway argument?

```
{ Hello  
! Paragraph ended before \bold was complete.
```

Programmer first class

```
\long\def\bold#1{{\bf #1}}
```

Hacker

```
\def\beginbold{\bgroup\bf}
```

```
\def\endbold{\egroup}
```

```
\beginbold
```

```
Hello
```

```
world
```

```
\endbold
```

Wizard

```
\def\bold{\bgroup\bf\let\next=}
```

```
\bold{text}
```

```
\bgroup\bf\let\next={text}
```

```
bgroup bf t11 e11 x11 t11 }2
```

```
{1 bf t11 e11 x11 t11 }2
```


예제 : 스크립트 매크로

```
\beginscript
```

Now, at last, you can easily typeset conversations you eavesdrop on in restaurants and on planes.

Really? That's just what I've been waiting for! How do I do it?

Exactly the way this script was done.

Is it easy?

Extremely.

```
\endscript
```

Now, at last, you can easily typeset conversations you eavesdrop on in restaurants and on planes.

Really? That's just what I've been waiting for! How do I do it?

Exactly the way this script was done.

Is it easy?

Extremely.

```
\let\endgraf=\par
\newcount\spk
\def\beginscript{\bgroup \parindent=0pt \color{red}
  \spk=1 \rightskip.4in
  \def\par{\ifnum\spk=1 \endgraf \color{blue} \spk=2
    \leftskip.4in \rightskip0in
    \else \endgraf \color{red} \spk=1
    \leftskip0in \rightskip.4in \fi}}
\def\endscript{\egroup}
```

예제 : FIFO 매크로

- `\if <token1> <token2>`
 - 두 토큰의 문자 코드(character code)가 같은지 검사한다.
 - 검사 전에, `\if` 뒤에 나오는 토큰을 두 개의 토큰이 나올 때까지 전개한다.

```
\def\foo{abc} \def\bar{abc}  
\if\foo\bar true \else false \fi % => false
```

- `\ifx <token1> <token2>`
 - 두 개의 토큰이 서로 일치하는지 검사한다.
 - `\if`와 달리 `\ifx` 뒤에 나오는 토큰을 전개하지 않는다.

```
\def\foo{abc} \def\bar{abc}  
\ifx\foo\bar true \else false \fi % => true
```

```
\def\fifo#1{\ifx\ofif#1\ofif\fi  
  \process#1\fifo}  
\def\ofif#1\fifo{\fi}
```

```
\fifo abc\ofif  
=> \process a \process b \process c
```

```
\def\fifo#1{\ifx\ofif#1\ofif\fi  
  \process#1\fifo}  
\def\ofif#1\fifo{\fi}
```

```
\def\getlength#1{  
  \count255=0  
  \def\process##1{\advance\count255 by 1 }  
  \fifo #1\ofif \number\count255}  
\getlength{argument} % => 8
```

```
\zipcode#1{%  
  \def\process##1{\fbox{##1}\,}  
  \fifo #1\ofif}
```

```
\zipcode{17077} % => 

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 7 | 0 | 7 | 7 |
|---|---|---|---|---|


```


- The T_EXbook
- T_EX for the Impatient
- T_EX By Topic
- Notes On Programming in TeX
- Macros for Jill
- FIFO and LIFO sing the BLUes
- The TeX Hierarchy

¿Tienes alguna pregunta?