

\LaTeX studio 在线教程—— \LaTeX 3 简介 (I)

历史回眸 & 背景知识

曾祥东

2018 年 4 月 24 日

复旦大学 物理系



1. \LaTeX 3 是啥?
2. \TeX 编程概要
 - 宏语言基础
 - 数字与编码
 - 案例赏析
3. 如何学习?

L^AT_EX3 是啥?



图 1 D. E. Knuth: $\text{T}_{\text{E}}\text{X}$

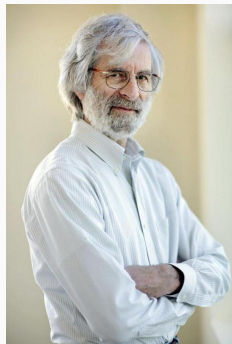


图 2 L. Lamport: \LaTeX



图 3 F. Mittelbach: $\text{\LaTeX 2}_{\epsilon}$

L^AT_EX 2_ε 的不足 (1)

1. 格式控制没有提供易用的接口:

```
% 脚注中编号不用上标
\makeatletter
\renewcommand\@makefnmark[1]{%
\hspace*{-2em}
\parindent 0em
\noindent
\hb@xt@ 1.8em{\hss
\@thefnmark. }%
#1}
\makeatother
```

2. 导言区成为宏包集锦:

```
% 导言区
\documentclass[UTF8]{ctexart}
\usepackage{amsmath,mathtools}
\usepackage{caption}
\usepackage{enumitem}
\usepackage[perpage]{footmisc}
\usepackage{geometry}
\usepackage{array,longtable}
\usepackage[group-digits]{siunitx}
\usepackage{physics}
```

解决方案: KOMA-Script、memoir 等文档类

L^AT_EX 2_ε 的不足 (2)

3. 缺乏良好的编程界面

```
% 来自 ctex-common.def (ctex 1.02d)
\def\CTEX@replacecommand#1#2#3{%
  \expandafter\expandafter\expandafter\let\expandafter
    \csname #1#3\expandafter\endcsname
    \csname #2#3\endcsname
  \expandafter\expandafter\expandafter\def\expandafter
    \csname #2#3\expandafter\endcsname
    {\csname #1#3\endcsname}}

```

4. 风骚的命名法

\@ne	\tw@	\thr@@	\sixt@@n	\@cclv	\m@ne	\z@	\p@
% 1	2	3	16	255	-1	0pt	1pt

5. 内容编写与格式控制的关系不够明晰

- Knuth → Lamport: 个人奋斗 vs 历史进程
- \LaTeX 的哲学:
 - `\documentclass{fduthesis}`, 而非几百行格式设置
 - `\emph{强调文字}`, 而非 `\textit{强调文字}`
 - `\title{标题}`, 而非 `~~~~~{\Large 标题}`
- **beamer** 的启发: “作者、设计师、程序员” 三个层次

\LaTeX 3

- 格式 (format)
- \LaTeX 2 ϵ : 不抛弃, 不放弃
- 遥不可及的未来

expl3

- 宏包 (“experimental \LaTeX 3”)
- 可以运行在 plain $\text{T}_{\text{E}}\text{X}$ 、 \LaTeX 2 ϵ 、Con $\text{T}_{\text{E}}\text{X}$ t 中
- 接口基本稳定:
 - $\text{C}_{\text{T}}\text{E}_{\text{X}}$, xeCJK, xypinyin
 - fontspec, unicode-math, siunitx 等
 - fduthesis



图 4 expl3 的标志——蜂鸟

TEX 编程概要

原语、宏与控制序列 (1)

原语 (primitive)：引擎所提供的命令

- 宏语言的基础
- 类比：C/C++ 关键字
- 例子：
 - 由 $\text{T}_{\text{E}}\text{X}$ 定义：`\def`、`\let`、`\if`、`\hbox` 等
 - 由 $\varepsilon\text{-T}_{\text{E}}\text{X}$ 定义：`\protected`、`\numexpr` 等
 - 由 $\text{LuaT}_{\text{E}}\text{X}$ 定义：`\directlua` 等
- 所有的原语？
 - $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ 中的 `l3names` 模块

原语、宏与控制序列 (2)

宏 (macro)：通过 `\def`、`\edef` 及其兄弟姐妹声明的东西

- 原语的封装，会被进一步展开
- 以 `\` 开头的命令（控制序列）+ 活动字符
- 例子：
 - \LaTeX 中的 `\newcommand`、`\item`、`\emph` 等
 - 活动字符：~
- 环境？
 - `\myenv + \endmyenv`

控制序列 (control sequence, 简称 cs): 所有以 `\` 开头的东西

- 原语
- 以 `\` 开头的宏 (不包括活动字符)
- 由 `\countdef`、`\dimendef` 等声明的寄存器变量
 - e.g. `\m@ne`、`\baselineskip`
- 由 `\chardef` 或 `\mathchardef` 定义的字符常量
 - e.g. `\$`、`\alpha`、`\sum`

\def 的兄弟姐妹

- 语法：

```
\def<命令模板>{<定义>}
```

- 参数用 #1、#2 等来表示
- 最多有 9 个参数
- 花括号不表示分组
- 重复定义不会报错

- 允许参数分段：\long\def
- 全局定义：\global\def = \gdef
- 展开控制：\edef

\edef —— 展开控制（邪恶之源）

- 原因分析：

```
% \def\double#1{#1,#1}  
% \def\double#1{\double{#1};\double{#1}}  
\double{A}  
→ \double{A}; \double{A}  
→ \double{A}; \double{A}; \double{A}; \double{A}  
→ ...
```

子又生孫，孫又生子；子又有子，子又有孫；子子孫孫無窮匱也。

——北山愚公

- 使用 \edef 之后：

```
\double{#1} → #1,#1;#1,#1
```

\let —— 命令的拷贝

- 语法：

```
\let<命令>=<标识符> % “=” 可以忽略
```

- \let 会将原有命令保存起来
- \def 只是记号的替换

常见进制

- 十进制 (DEC)
逢十进一
- 二进制 (BIN)
逢二进一，只有 0 和 1 —— 机器语言
- 八进制 (OCT)
逢八进一，只有 0 ~ 7
- 十六进制 (HEX)
逢十六进一，0 ~ 9 外加 A ~ F

- 例子：

1614	% DEC
= 110 0100 1110	% BIN
= 3116	% OCT
= 64E	% HEX

3.14	% DEC
= 11.0010 0011 1101 ...	% BIN
= 3.1075 ...	% OCT
= 3.23D7 ...	% HEX

数字的表示

C 语言

- 八进制以 `0` 开头
- 十六进制以 `0x` 开头

T_EX

- 八进制用 `'⟨数字⟩` 表示
- 十六进制用 `"⟨数字⟩` 表示（需要大写）

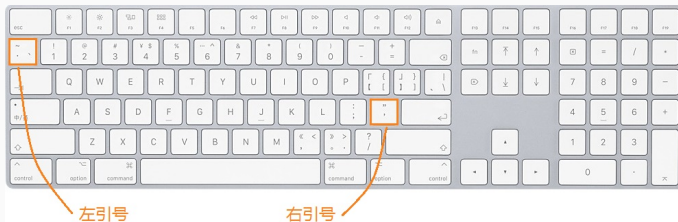


图 5 苹果 Magic Keyboard

ASCII 码表

控制字符				符号、数字				大写字母、符号				小写字母、符号			
00	NUL	10	DLE	20	SP	30	0	40	@	50	P	60	`	70	p
01	SOH	11	DC1	21	!	31	1	41	A	51	Q	61	a	71	q
02	STX	12	DC2	22	"	32	2	42	B	52	R	62	b	72	r
03	ETX	13	DC3	23	#	33	3	43	C	53	S	63	c	73	s
04	EOT	14	DC4	24	\$	34	4	44	D	54	T	64	d	74	t
05	ENQ	15	NAK	25	%	35	5	45	E	55	U	65	e	75	u
06	ACK	16	SYN	26	&	36	6	46	F	56	V	66	f	76	v
07	BEL	17	ETB	27	'	37	7	47	G	57	W	67	g	77	w
08	BS	18	CAN	28	(38	8	48	H	58	X	68	h	78	x
09	HT	19	EM	29)	39	9	49	I	59	Y	69	i	79	y
0A	LF	1A	SUB	2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
0B	VT	1B	ESC	2B	+	3B	;	4B	K	5B	[6B	k	7B	{
0C	FF	1C	FS	2C	,	3C	<	4C	L	5C	\	6C	l	7C	
0D	CR	1D	GS	2D	-	3D	=	4D	M	5D]	6D	m	7D	}
0E	SO	1E	RS	2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
0F	SI	1F	US	2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

- 连续两个上标符号 ^^ 后跟一个字符 c ，可以表示另一个字符 c' :
 - $64 \leq \text{ASCII}_c \leq 127 \Rightarrow \text{ASCII}_{c'} = \text{ASCII}_c - 64$
 - $0 \leq \text{ASCII}_c \leq 63 \Rightarrow \text{ASCII}_{c'} = \text{ASCII}_c + 64$
- ^^ 后跟两个小写十六进制数字，可以表示任意 ASCII 符号

- 字符 → 编码：`⟨字符⟩`
- 编码 → 字符：`\\char{⟨编码⟩}`

- 特殊字符

% 以下字符无法用键盘直接输入，
% 必须用 `\\` 转义：

~ # \$ % ^ & { } _ \\

% 以下字符可以用键盘直接输入：

, . ; : ! ? ` ' - / *
() [] @ | < > + =

类别码

类别码	含义	符号	ASCII 码 (HEX)
0	转义符	\	5C
1	组开始	{	7B
2	组结束	}	7D
3	切换数学环境	\$	24
4	表格对齐	&	26
5	回车	CR	0D
6	参数	#	23
7	上标	^	5E
8	下标	_	5F
9	可忽略字符	NUL	00
10	空格	SP	20
11	字母	A ~ Z, a ~ z	-
12	其他字符	本表未列出的其他符号	-
13	活动字符	~	7E
14	注释	%	25
15	无效符	DEL	7F

特殊的类别码

- 9（可忽略字符）
T_EX 遇到它后将直接跳过，就仿佛它不在那里一样
- 13（活动字符）
本身就是一个宏。例：~ = `\nobreakspace`
- 15（无效符）
T_EX 遇到它也会跳过，但会打印如下错误信息：

```
! Text line contains an invalid character.
```

换行 & 回车

- 换行 (Line feed, LF)
 - C 语言: `\n`
 - T_EX: `^^J`
- 回车 (Carriage return, CR)
 - C 语言: `\r`
 - T_EX: `^^M`
- 各系统下的“换行”符:
 - Windows: CR + LF
 - Linux: LF
 - Mac (old): CR
- T_EX 读入文件之后, 每行均以 `^^M` 结尾



图 6 HERMES 3000 打字机

\catcode —— 类别码的切换

- 语法:

```
\catcode⟨字符编码⟩=⟨类别码⟩ % “=” 可以忽略
```

- 例子:

```
\catcode`\\"=\active
%          0=|          7=#          3=}          2={
\catcode`00 \catcode`76 \catcode`32 \catcode`21
% \def"#1"{\textbf{#1}}
\def"71"20\textbf27133
This is an "important" word.
```

⇒ This is an **important** word.

Show time!

```
\let~\catcode~`76~`A13~`F1~`j00~`P2jdefA71F~`7113jdefPALLF
PA''FwPA;;FPAZZFLaLPA//71F71iPAHHFLPAzzFenPASSFthP;A$$FevP
A@@FfPARR717273F737271P;ADDFRgniPAWW71FPATTfVePA**FstRsamP
AGGFRruoPAqq71.72.F717271PAYY7172F727171PA??Fi*LmPA&&71jfi
Fjfi71PAVVFjbigskipRPWGAAU71727374 75,76Fjpar71727375Djifx
:76jelset&U76jfiPLAKK7172F71l7271PAXX71FVLn0SeL71SLRyadR@oL
RrhC?yLRurtKFeLPFovPgaTLtReRomL;PABB71 72,73:Fjif.73.jelset
B73:jfiXF71PU71 72,73:PWs;AMM71F71diPAJJFRdriPAQQFRsreLPAI
I71Fo71dPA!!FRgiePBt'el@ lTLqdrYmu.Q.,Ke;vz vzLqipip.Q.,tz;
;Lql.IrsZ.eap,qn.i. i.eLlMaesLdRcna,;;!;h htLqm.MRasZ.il,%
s$;z zLqs'.ansZ.Ymi,/sx ;LYegseZRyal,@i;@ TLRlogdLrDsw,@;G
LcYlaDLbJsW,SWXJW ree @rzchLhzsW;;WERcesInW qt.'oL.Rtrul;e
doTsW,Wk;Rri@stW aHAHHFndZPpqar.tridgeLinZpe.LtYer.W,:jbye
```

—*xii.tex* by David Carlisle

你尽管看



看得懂算我输

案例：verbatim 环境

- verbatim 环境用来表示抄录代码片段，使得 \LaTeX 中的特殊符号不发挥作用：

```
\begin{verbatim}
#!/usr/bin/env perl
$name = "world";
print "Hello, $name!\n";
\end{verbatim}
```

⇒

```
#!/usr/bin/env perl
$name = "world";
print "Hello, $name!\n";
```

- 另有 verbatim* 环境，可将空格显示输出为 $_$ ：

```
\begin{verbatim*}
#!/usr/bin/env perl
$name = "world";
print "Hello, $name!\n";
\end{verbatim*}
```

⇒

```
#!/usr/bin/env\_perl
$name\_=\_ "world";
print\_ "Hello,\_ $name!\n";
```

verbatim 的实现

- 开始位置的核心代码：

```
1 {\catcode`\_=\active%
2 \gdef\@vobeyspaces{\catcode`\_ \active\let\_ \@xobeysp}}
3 \def\@verbatim{
4   % 此处略去三百字……
5   \let\do\@makeother \dospecials
6   \obeylines \verbatim@font \@noligs
7   \everypar \expandafter{\the\everypar \unpenalty}}
```

- 结束位置的核心代码：

```
8 \begingroup
9   \catcode`=0 \catcode`[=1
10  \catcode`] =2 \catcode`\{=12 \catcode`\}=12 \catcode`\\=12
11  \gdef\@xverbatim#1\end{verbatim}[#1\end[verbatim]]
12 \endgroup
```

如何学习？

- 复习 \LaTeX
 - 刘海洋. 《 \LaTeX 入门》
 - 包太雷 (黄新刚). 《 \LaTeX Notes —— 雷太赫排版系统简介 (第二版)》
 - 胡伟. 《 $\text{\LaTeX} 2_{\epsilon}$ 完全学习手册 (第二版)》
 - 胡伟. 《 $\text{\LaTeX} 2_{\epsilon}$ 文类和宏包学习手册》
- 深入 \TeX
 - D E Knuth. *The \TeX book*
 - P W Abrahams. *\TeX for the Impatient*
 - 英文版: `texdoc impatient`
 - 中文版: `texdoc impatient-cn`

- 手册、代码

- T_EX 源代码: `tex.pdf`
- L^AT_EX 2_ε 源代码: `source2e.pdf`
- L^AT_EX3 介绍: `expl3.pdf`
- L^AT_EX3 接口: `interface3.pdf`
- L^AT_EX3 源代码: `source3.pdf`
- C_T_EX 宏集手册: `ctex.pdf`
- x_eC_JK 宏包文档: `xeCJK.pdf`

获取方法: `texdoc` `<文件名>`

- 网站

- GitHub
 - latex3/latex3
 - C_T_EX-org/ctex-kit
- TeX - LaTeX StackExchange
 - macros 板块
 - expl3、latex3 板块
- Google