

L^AT_EX3 教程四：常用库

项子越

ziyue.alan.xiang@gmail.com

<https://github.com/xziyue/latex3-chinese-video>

2021 年 12 月 17 日

凭据表 (token list)

- 用于存储文档，命令等内容
- 提供多种使用及修改的接口
- 可用于存储参数供其它函数调用
- 主要缺点：无法存储换行符

凭据表的基本操作

对凭据表变量赋值

```

1 \ExplSyntaxOn
2 \tl_gclear:N \g_tmpa_tl % 清除旧值
3 \tl_gset:Nn \g_tmpa_tl {新值}
4 \cs_meaning:N \g_tmpa_tl
5 \ExplSyntaxOff

```

macro:-> 新值

在左侧追加

```

1 \ExplSyntaxOn
2 \tl_gput_left:Nn \g_tmpa_tl {一个}
3 \cs_meaning:N \g_tmpa_tl
4 \ExplSyntaxOff

```

macro:-> 一个新值

在右侧追加

```

1 \ExplSyntaxOn
2 \tl_gput_right:Nn \g_tmpa_tl {被定义}
3 \cs_meaning:N \g_tmpa_tl
4 \ExplSyntaxOff

```

macro:-> 一个新值被定义

凭据表的基本操作

遍历

```

1 \ExplSyntaxOn
2 \tl_set:Nn \l_tmpa_tl {ab{cd}\P e}
3 \tl_map_inline:Nn \l_tmpa_tl {
4   \par 元素: #1
5 }
6 \ExplSyntaxOff

```

元素: a
 元素: b
 元素: cd
 元素: ¶
 元素: e

凭据表的基本操作

按元素序号访问

```

1 \ExplSyntaxOn
2 \tl_set:Nn \l_tmpa_tl {ab{cd}\P e}
3 \par\tl_count:N \l_tmpa_tl
4 \par\tl_item:Nn \l_tmpa_tl {1}
5 \par\tl_item:Nn \l_tmpa_tl {2}
6 \par\tl_item:Nn \l_tmpa_tl {3}
7 \par\tl_item:Nn \l_tmpa_tl {4}
8 \par\tl_item:Nn \l_tmpa_tl {5}
9 \ExplSyntaxOff

```

5
a
b
cd
P
e

凭据表的基本操作

取头/取尾

```

1 \ExplSyntaxOn
2 \tl_set:Nn \l_tmpa_tl {abcde}
3 \par\tl_head:N \l_tmpa_tl
4 \par\tl_tail:N \l_tmpa_tl
5 \ExplSyntaxOff

```

a
bcde

遍历

```

1 \ExplSyntaxOn
2 \tl_set:Nn \l_tmpa_tl {ab{cd}\P e}
3 \tl_map_inline:Nn \l_tmpa_tl {
4   \par 元素: #1
5 }
6 \ExplSyntaxOff

```

元素: a
元素: b
元素: cd
元素: ¶
元素: e

凭据表与字符串

字符串库 (string) 与凭据表十分相似。两者的区别主要在于字符串中所存储的所有内容都会以原样输出。

```

1 \ExplSyntaxOn
2 \tl_set:Nn \l_tmpa_tl
   \> {\$\frac{\alpha}{\beta}}$}
3 \tl_use:N \l_tmpa_tl
4 \ExplSyntaxOff
  
```

$$\frac{\alpha}{\beta}$$

```

1 \ExplSyntaxOn
2 \str_set:Nn \l_tmpa_str
   \> {\$\frac{\alpha}{\beta}}$}
3 \str_use:N \l_tmpa_str
4 \ExplSyntaxOff
  
```

`\frac {\alpha }{\beta }$`

中英文数字转换

```

1 \ExplSyntaxOn
2 \tl_new:N \g_doc_chn_num_tl
3 \tl_gset:Nn \g_doc_chn_num_tl {一二三四五六七八
   ↪ 九}
4 \cs_gset:Npn \doc_arabic_to_chn_a:n #1 {
5   \tl_item:Nn \g_doc_chn_num_tl {#1}
6 }
7 \par\doc_arabic_to_chn_a:n {1}
8 \par\doc_arabic_to_chn_a:n {3}
9 \par\doc_arabic_to_chn_a:n {5}
10 \ExplSyntaxOff

```

一
三
五

中英文数字转换

```

1 \ExplSyntaxOn
2 \tl_new:N \g_doc_chn_dec_tl
3 \tl_gset:Nn \g_doc_chn_dec_tl {十百千万}
4 \cs_gset:Npn \doc_arabic_to_chn:n #1 {
5   \tl_set_eq:NN \l_tmpa_tl \g_doc_chn_dec_tl
6   \tl_clear:N \l_tmpb_tl
7   \doc_arabic_to_chn_b:nN {#1} \l_tmpa_tl \tl_use:N \l_tmpb_tl
8 }
9 \cs_gset:Npn \doc_arabic_to_chn_b:nN #1#2 {
10  \int_set:Nn \l_tmpa_int {\int_div_truncate:nn {#1}{10}}
11  \int_set:Nn \l_tmpb_int {\int_mod:nn {#1}{10}}
12  \int_compare:nNnT {\l_tmpb_int} > {0} {
13    \tl_put_left:Nx \l_tmpb_tl {\doc_arabic_to_chn_a:n {\l_tmpb_int}}
14  }
15  \int_compare:nNnT {\l_tmpa_int} > {0} {
16    \tl_put_left:Nx \l_tmpb_tl {\tl_head:N #2}
17    \tl_set:Nx #2 {\tl_tail:N #2}
18    \exp_args:Nx \doc_arabic_to_chn_b:nN {\int_use:N \l_tmpa_int} #2
19  }
20 }
21 \ExplSyntaxOff

```

中英文数字转换

```
1 \ExplSyntaxOn
2 \par\doc_arabic_to_chn:n {5}
3 \par\doc_arabic_to_chn:n {15}
4 \par\doc_arabic_to_chn:n {615}
5 \par\doc_arabic_to_chn:n {4615}
6 \par\doc_arabic_to_chn:n {84615}
7 \par\doc_arabic_to_chn:n {20}
8 \par\doc_arabic_to_chn:n {320}
9 \ExplSyntaxOff
```

五
一十五
六百一十五
四千六百一十五
八万四千六百一十五
二十
三百二十

接收文本参数的格式化

假设我们想设计一个命令，它能根据一个参数来选择如何格式化文字。比如 `\fmt{cu}{abc}` 会将 `abc` 加粗；`\fmt{xie}{abc}` 会使用斜体；`\fmt{xian}{abc}` 会使用下划线等。

```

1 \ExplSyntaxOn
2 \newcommand{\fmt}[2]{
3   \str_if_eq:nnT {#1} {cu} {\textbf{#2}}
4   \str_if_eq:nnT {#1} {xie} {\textit{#2}}
5   \str_if_eq:nnT {#1} {xian}
6     ↪ \underline{#2}
7 }
8 \par\fmt{cu}{abc}
9 \par\fmt{xie}{abc}
10 \par\fmt{xian}{abc}
11 \ExplSyntaxOff

```

abc
abc
abc

`\tl_if_eq:` 不仅比较字符，还会比较字符的类别码 (category code)。因此可能出现字符看上去相等但是程序判断为不等的情况。

接收文本参数的格式化

```

1 \ExplSyntaxOn
2 \newcommand{\fmt}[2]{
3   \str_case:nnTF {#1} {
4     {cu} {\textbf{#2}}
5     {xie} {\textit{#2}}
6     {xian} {\underline{#2}}
7   }{
8     \textcolor{green}{\ (good)}
9   }{
10    \textcolor{red}{#2\ (bad)}
11  }
12 }
13 \par\fmt{cu}{abc}
14 \par\fmt{xie}{abc}
15 \par\fmt{xian}{abc}
16 \par\fmt{123}{abc}
17 \ExplSyntaxOff

```

abc (good)
abc (good)
abc (good)
 abc (bad)

序列表 (sequence)

- 用于存储并管理一系列的元素
- 可以按序号访问，也可以只访问首元素或尾元素
- 可以把多个元素连接起来

序列表的基本操作

在右侧追加

```

1 \ExplSyntaxOn
2 \seq_gclear:N \g_tmpa_seq
3 \seq_gput_right:Nn \g_tmpa_seq {一}
4 \seq_gput_right:Nn \g_tmpa_seq {个}
5 \seq_gput_right:Nn \g_tmpa_seq {元素}
6 \par\seq_use:Nn \g_tmpa_seq {,~}
7 \ExplSyntaxOff

```

一, 个, 元素

在左侧追加

```

1 \ExplSyntaxOn
2 \seq_gput_left:Nn \g_tmpa_seq {不只}
3 \par\seq_use:Nn \g_tmpa_seq {,~}
4 \ExplSyntaxOff

```

不只, 一, 个, 元素

序列表的基本操作

获取最左侧元素

```
1 \ExplSyntaxOn
2 \seq_get_left:NN \g_tmpa_seq \l_tmpa_tl
3 \tl_use:N \l_tmpa_tl
4 \ExplSyntaxOff
```

不只

获取最右侧元素

```
1 \ExplSyntaxOn
2 \seq_get_right:NN \g_tmpa_seq \l_tmpa_tl
3 \tl_use:N \l_tmpa_tl
4 \ExplSyntaxOff
```

元素

`\seq_pop_left:` 以及 `\seq_pop_right:` 的功能类似, 但是会从序列中移除元素。

序列表的基本操作

遍历

```
1 \ExplSyntaxOn
2 \seq_map_inline:Nn \g_tmpa_seq {
3   \par 元素: #1
4 }
5 \ExplSyntaxOff
```

元素: 不只

元素: 一

元素: 个

元素: 元素

序列表的基本操作

按序号访问

```

1 \ExplSyntaxOn
2 \par\seq_item:Nn \g_tmpa_seq {1}
3 \par\seq_item:Nn \g_tmpa_seq {3}
4 \ExplSyntaxOff

```

不只
↑

序列表的本质是凭据表

```

1 \ExplSyntaxOn
2 \cs_meaning:N \g_tmpa_seq
3 \ExplSyntaxOff

```

macro:->\s__seq __seq_item:n {不
只}__seq_item:n {一}__seq_item:n
{个}__seq_item:n {元素}

序列表的基本操作

连接序列表中的元素

```

1 \ExplSyntaxOn
2 \par\seq_use:Nn \g_tmpa_seq {, }
3 \par\seq_use:Nn \g_tmpa_seq {—}
4 \ExplSyntaxOff

```

不只，一，个，元素

不只———一个——元素

制作正弦表

我们希望制作如下表格：

$\sin 0^\circ = 0$	$\sin 1^\circ = 0.01745240643728352$	$\sin 2^\circ = 0.03489949670250098$
$\sin 3^\circ = 0.05233595624294385$	$\sin 4^\circ = 0.06975647374412532$	$\sin 5^\circ = 0.08715574274765819$
$\sin 6^\circ = 0.1045284632676535$	$\sin 7^\circ = 0.1218693434051475$	$\sin 8^\circ = 0.1391731009600655$
$\sin 9^\circ = 0.1564344650402309$	$\sin 10^\circ = 0.1736481776669304$	$\sin 11^\circ = 0.1908089953765449$
$\sin 12^\circ = 0.2079116908177594$	$\sin 13^\circ = 0.2249510543438651$	$\sin 14^\circ = 0.2419218955996678$
$\sin 15^\circ = 0.2588190451025208$	$\sin 16^\circ = 0.2756373558169993$	$\sin 17^\circ = 0.2923717047227368$

制作正弦表

```

1 \ExplSyntaxOn
2 \seq_gclear:N \g_tmpa_seq
3 \int_step_inline:nnn {0} {18} {
4   \seq_put_right:Nn \l_tmpa_seq {
5     $\sin \#1^\circ = \fp_eval:n {\sin(\#1 * \c_one_degree_fp)}$
6   }
7   \int_compare:nNnT {\#1} > {0} {
8     \int_compare:nNnT {\int_mod:nn {\#1 + 1} {3}} = {0} {
9       \seq_gput_right:Nx \g_tmpa_seq {
10         \seq_use:Nn \l_tmpa_seq {\&}
11       }
12       \seq_clear:N \l_tmpa_seq
13     }
14   }
15 }
16 \ExplSyntaxOff

```

制作正弦表

```

1 \ExplSyntaxOn
2 \tiny
3 \centering
4 \begin{tabular}{lll}
5 \toprule
6 \seq_use:Nn \g_tmpa_seq {\}
7 \\\bottomrule
8 \end{tabular}
9 \par
10 \ExplSyntaxOff

```

$\sin 0^\circ = 0$	$\sin 1^\circ = 0.01745240643728352$	$\sin 2^\circ = 0.03489949670250098$
$\sin 3^\circ = 0.05233595624294385$	$\sin 4^\circ = 0.06975647374412532$	$\sin 5^\circ = 0.08715574274765819$
$\sin 6^\circ = 0.1045284632676535$	$\sin 7^\circ = 0.1218693434051475$	$\sin 8^\circ = 0.1391731009600655$
$\sin 9^\circ = 0.1564344650402309$	$\sin 10^\circ = 0.1736481776669304$	$\sin 11^\circ = 0.1908089953765449$
$\sin 12^\circ = 0.2079116908177594$	$\sin 13^\circ = 0.2249510543438651$	$\sin 14^\circ = 0.2419218955996678$
$\sin 15^\circ = 0.2588190451025208$	$\sin 16^\circ = 0.2756373558169993$	$\sin 17^\circ = 0.2923717047227368$

随机列表

设计两个命令：

- **\AddToList**：把元素加入列表
- **\ShowList**：以随机顺序输出列表中的元素

```
1 \ExplSyntaxOn
2 \seq_gclear:N \g_tmpa_seq
3 \cs_gset:Npn \AddToList #1 {
4   \seq_gput_right:Nn \g_tmpa_seq {#1}
5 }
6 \ExplSyntaxOff
```

随机列表

```
1 \ExplSyntaxOn
2 \cs_gset:Npn \ShowList {
3   \seq_shuffle:N \g_tmpa_seq
4   \begin{itemize}
5     \seq_map_inline:Nn \g_tmpa_seq {
6       \item ##1
7     }
8   \end{itemize}
9 }
10 \ExplSyntaxOff
```

随机列表

```
1 \AddToList{两}  
2 \AddToList{只}  
3 \AddToList{老虎}  
4 \AddToList{跑}  
5 \AddToList{得}  
6 \AddToList{快}  
7 \ShowList
```

- 快
- 跑
- 只
- 两
- 老虎
- 得

逗号分隔表 (comma-separated list)

- 接口与序列基本相同
- 区别：可以用逗号分隔的内容来赋值

```

1 \ExplSyntaxOn
2 \clist_set:Nn \l_tmpa_clist {a,b,cd,\P,e}
3 \clist_map_inline:Nn \l_tmpa_clist {
4   \par 元素: #1
5 }
6 \ExplSyntaxOff

```

元素: a
 元素: b
 元素: cd
 元素: ¶
 元素: e

逗号分隔表 (comma-separated list)

按序号访问

<pre> 1 \ExplSyntaxOn 2 \clist_set:Nn \l_tmpa_clist {a,b,cd,\P,e} 3 \par\clist_item:Nn \l_tmpa_clist {1} 4 \par\clist_item:Nn \l_tmpa_clist {2} 5 \par\clist_item:Nn \l_tmpa_clist {3} 6 \par\clist_item:Nn \l_tmpa_clist {4} 7 \par\clist_item:Nn \l_tmpa_clist {5} 8 \ExplSyntaxOff </pre>	<pre> a b cd ¶ e </pre>
--	-------------------------

接收文本参数的格式化

```

1 \ExplSyntaxOn
2 \newcommand{\fmt}[2]{
3   \group_begin:
4   \clist_set:Nn \l_tmpa_clist {#1}
5   \clist_map_inline:Nn \l_tmpa_clist {
6     \str_case:nn {##1} {
7       {cu}  {\bfseries}
8       {xie} {\itshape}
9       {hong} {\color{red}}
10    }
11  } #2
12  \group_end:
13 }
14 \ExplSyntaxOff
15 \par\fmt{cu}{abc}
16 \par\fmt{xie}{abc}
17 \par\fmt{hong}{abc}
18 \par\fmt{cu,hong}{abc}
19 \par\fmt{xie,hong}{abc}
20 \par\fmt{cu,xie,hong}{abc}

```

abc*abc*

abc

abc*abc***abc**

属性表 (property list)

- 与 Python 中的字典 (dict) 类似，提供键—值访问

属性表的基本操作

设置值

```

1 \ExplSyntaxOn
2 \prop_gput:Nnn \g_tmpa_prop {key1} {val1}
3 \prop_gput:Nnn \g_tmpa_prop {key2} {val2}
4 \ExplSyntaxOff

```

取回值

<pre> 1 \ExplSyntaxOn 2 \par\prop_item:Nn \g_tmpa_prop {key1} 3 \par\prop_item:Nn \g_tmpa_prop {key2} 4 \ExplSyntaxOff </pre>	<div style="border-left: 1px dashed black; padding-left: 10px;"> val1 val2 </div>
---	---

取回值的函数还有 `\prop_get:` 及 `\prop_pop` 等。

属性表的基本操作

遍历

```
1 \ExplSyntaxOn
2 \prop_map_inline:Nn \g_tmpa_prop {
3   \par 键: #1 \quad 值: #2
4 }
5 \ExplSyntaxOff
```

键: key1 值: val1
键: key2 值: val2

遍历输出的顺序是未定义的。

属性表的基本操作

属性表也可通过类似逗号分隔表的方式初始化。

```

1 \ExplSyntaxOn
2 \prop_set_from_keyval:Nn \l_tmpa_prop {
3   1= 一,
4   2= 二,
5   3= 三
6 }
7
8 \prop_map_inline:Nn \l_tmpa_prop {
9   \par 键: #1 \quad 值: #2
10 }
11 \ExplSyntaxOff

```

键: 1	值: 一
键: 2	值: 二
键: 3	值: 三

缩写管理

设计两个命令：

- `\AddAcronym`：用于添加缩写到系统中
- `\Acro`：用于取回缩写的全称

```

1 \ExplSyntaxOn
2 \prop_new:N \g_doc_acro_prop
3 \newcommand{\AddAcronym}[2]{
4   \prop_gput:Nnn \g_doc_acro_prop {#1} {#2}
5 }
6 \newcommand{\Acro}[1]{
7   \textbf{\prop_item:Nn \g_doc_acro_prop
8     ↪ {#1}}
9 }
9 \ExplSyntaxOff
10
11 \AddAcronym{gmm}{高斯混合模型}
12 \AddAcronym{em}{期望最大化算法}
13
14 使用\Acro{em}来优化\Acro{gmm}

```

使用期望最大化算法来优化高斯混合模型