

高亮 T_EX 和 L^AT_EX3 代码——使用 texhigh 宏包

霽月*

2025 年 2 月 21 日

`texhigh` 宏包是专用来高亮 \TeX 文件的宏包。基于由 Rust 编写的命令行工具 `texhigh`¹，处理 1.24MB 左右（37,700 余行）的 `expl3-code.tex` 只需 0.16s 左右（操作系统为 Windows，CPU 为 i7-12700H），处理速度约为 `minted` 宏包使用的 `pygmentize` 的 17 倍，`texhigh` 的增强模式也比它快 10 到 15 倍。对于普通大小的 \TeX 代码，处理它们所需的时间相比于 \TeX 文件本身编译所需的时间，已经可以忽略不记。

`texhigh` 主要是在 `LATEX` 中为 `texhigh` 命令行工具提供交互接口。这要求在编译 `TEX` 文件时启用 `--shell-escape`。

texhigh 可以输出颜文字: $\varepsilon(\pi\sim\pi)3$, 只需使用 `\kaomoji{颜文字}`。默认使用系统字体, 也可自行设置 `*(\geq\nabla\leq*)o`。

ရွှေ(၂*၂)၂* ၁၅.၁၀ ပြ(၈.၂၀)၆၅ ၇ ၆၆.၁.၁၅၂၅ ၇ ၆၆.၁.၁၅၂၅ (၂°၁၀') ၂၇ ၁၁

使用颜文字时可能会遇到字体问题，这时在字符间插入零宽词连接符 U+2060 或可解决。

使用 `\kaomoji*` 还支持把单行文字输出为图片:

% 这里的 fontsize 影响图片的大小, 从而影响清晰度

`\kaomoji*[fontSize*=\Huge]{}` \longleftrightarrow

\hookrightarrow `\includegraphics[height=12bp]{`

```
\kaomoji*[fontsize=50bp]{\Uchar"1F43C }{\includegraphics[height=25bp]}
```



也可以自己封装一下这个命令：

*longaster@163.com

¹<https://github.com/Sophanatprime/texhigh-rs>

代码 2

```

\makeatletter
\NewDocumentCommand\inmoji{ D<>{\f@size\p@} ={\fontsize+} 0{} m }
  {\kaomoji*[fontsize={(\#1)*3},\#2]{\#3}
    {\includegraphics[height=\dimeval{\#1}]{}}}
\makeatother
\inmoji{🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒} \inmoji{^o^y}

```

🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒🕒 ^o^y

texhigh 提供 `\texhighverb`、`\texhighfile`、`\texhighinput` 这几个命令以及一个 texhigh 环境用于高亮 TeX 代码。

texhigh 还有很强的可配置性。

为了实现处理 TeX 源码与输出结果的分离，texhigh 使用“类型”和“类别”来区分不同的记号。字符和控制序列是不同的“类型”，控制序列之间可以有不同的“类别”，例如是原语、 \LaTeX 3 函数等。类型不可改变，而“类别”可以自由修改。

每个类型都有一些命令用于更改它们的“类别”的显示效果，如，对于一个控制序列，可以使用 `\THSetClassCS` 改变显示效果。可以为它们设置前景色、背景色，甚至渐变色和底纹等等。实际上普通文字可以显示成什么效果，它们就可以做到同样的效果。具体修改方式可以参考文末 basic 样式的源码。

texhigh 利用 tikz 实现了渐变和底纹效果，同时也可直接集成到 tcolorbox 宏包中。只需要在加载 texhigh 之前加载这几个宏包。

代码 3

```

\usepackage{tikz}
\usepackage{tcolorbox}
\usepackage{texhigh}
\tcbset{listing engine=texhigh} % 使用这个即可切换至 texhigh
% 若使用 xeCJK，即在 XeLaTeX 中使用 ctex，最好设置
\SetKeys[texhigh/high]{
  font=\ttfamily\xeCJKsetup{CJKecglue={\hskip 0pt plus 0.08\baselineskip}}
}
% 这样可避免在显示代码时中英文之间出现不必要的空格。

```

识别行内数学公式：

代码 4

```
\texhighverb!公式 $ \int_a^b x^2 dx = \frac{1}{3} x^3 |_a^b $!。
```

```
公式 $ \int_a^b x^2 dx = \frac{1}{3} x^3 |_a^b $。
```

渐变：

代码 5

```
\texhighverb[style=tikz.gradient, use-ctab=latex3, ↵
↵ config-file=config.cfg]|\sys_get_shell:nnNTF|
```

```
\sys_get_shell:nnNTF
```

底纹:

代码 6

```
\makeatletter
% #1: tikz options, #2: text
\def\myshadetext#1#2{\texhigh@shadetext{#1}{\bfseries #2}}
\makeatother
{\LARGE
% 在加载 texhigh 之前加载 tikz 宏包!
% 使用 grass.png 作为文字底纹, 依赖 tikz 的 fill.image 库, 会自动加载这个库。
\texhighverb[use-ctab=latex3, this-cs=\myshadetext{fill stretch ↵
↵ image=grass.png}]
|\sys_get_shell:nnNTF|
}
```

```
\sys_get_shell:nnNTF
```

中文命令识别 (TeX 原语带有下列线):

代码 7

```
\begin{texhigh}[output=\jobname.texhigh, use-ctab=cjk]
\def\好好好{中文 Good}
\好好好\relax
\end{texhigh}
```

```
\def\好好好{中文 Good}
\好好好\relax
```

类别码混合使用:

代码 8

```
% 自动检测 \makeatletter 和 \ExplSyntaxOn 块,
% 但 \makeatletter 这几个命令必需在行首, 前面可以有空格
\begin{texhigh}[lexer-catcode={atletter, explon}]
\def\foo@#1{[#1]} \foo@{F00} \@kernel
\makeatletter
\def\foo@:#1{[#1]} \foo@:#1{F00} \@kernel \scan_stop:
\ExplSyntaxOn
\cs_set:Npn \foo@: #1 { [#1] }
\foo@: {F00} \@kernel \scan_stop:
\ExplSyntaxOff
\@kernel \scan_stop:
```

```
\makeatother
\@kernel \scan_stop:
\end{texhigh}
```

```
\def\foo@#1{[#1]} \foo@{F00} \@kernel
\makeatletter
\def\foo@:#1{[#1]} \foo@:#1{F00} \@kernel \scan_stop:
\ExplSyntaxOn
\cs_set:Npn \foo@: #1 { [#1] }
\foo@: {F00} \@kernel \scan_stop:
\ExplSyntaxOff
\@kernel \scan_stop:
\makeatother
\@kernel \scan_stop:
```

也可手动调整类别码：

代码 9

```
\begin{texhigh}[
  lexer-catcode*={3}{9}{@=11, ?=11}, % [3, 9) 行, @ 和 ? 的类别码为 11
  % lexer-catcode*={3}{9}{ @?=11 }, % <- 可以合并
  lexer-catcode*={5}{7}{explon},      % [5, 7) 行, 启用 expl3 的类别码
]
\def\foo@#1{[#1]} \foo@{F00} \@kernel
\makeatletter
\def\foo@:#1{[#1]} \foo@:#1{F00} \@kerne? \scan_stop:
\ExplSyntaxOn
\cs_set:Npn \foo@: #1 { [#1] }
\foo@: {F00} \@kernel \scan_stop:
\ExplSyntaxOff
\@kernel \scan_stop:
\makeatother
\@kernel \scan_stop:
\end{texhigh}
```

```
\def\foo@#1{[#1]} \foo@{F00} \@kernel
\makeatletter
\def\foo@:#1{[#1]} \foo@:#1{F00} \@kerne? \scan_stop:
\ExplSyntaxOn
\cs_set:Npn \foo@: #1 { [#1] }
\foo@: {F00} \@kernel \scan_stop:
\ExplSyntaxOff
\@kernel \scan_stop:
\makeatother
\@kernel \scan_stop:
```

除了用 `use-ctab` 设置主类别码表以外，还可以添加额外的类别码表，优先选择靠后的类别码表：

```
\texhighverb[extra-catcode*=cjk, extra-catcode*=explon]
  {\cs_set:Npn \a_好的: #1 { aaa } \a_好的:}

% extra-catcode* 实际相当于设置 lexer-catcode* 的前两个值为 {0}{ }
% 即，从第 0 行开始，永不结束
\texhighverb[lexer-catcode*={0}{ }{cjk}]{\def\a和的#1{aaa} \a好的}

\cs_set:Npn \a_好的: #1 { aaa } \a_好的:
\def\a和的#1{aaa} \a好的
```

代码 10

还能进一步细化到列数：

```
% 1 相当于 [1, 0],
% [1, 9] 表示第 1 行第 9 个字符
% 这里就是从第一行的开始，直到第一行第 9 个字符（不含），空格的类别码为 0
% 设置类别码时，特殊字符必须转义，其它字符可转义也可不转义
\texhighverb[lexer-catcode*={1}{[1,9]}{\ =0}]{abd def #1{\space }}
%
%      ^--转义      ^--空格的类别码不再是 0

\texhighverb{abd def #1"\space"}

abd_def_#1{\space }
abd def #1"\space"
```

代码 11

`lexer-catcode*` 的前两个参数还支持更加复杂的模式，如字符串正则表达式和 $\text{T}_{\text{E}}\text{X}$ 正则表达式：

代码 12

```
% \I 放在开头表示这是字符串正则表达式，匹配源码
\begin{texhigh}[gobble=2, % 每行删除前两个字符
  lexer-catcode*=
    {\I\catcode`\!=11[\s\]}
    {\I\endgroup\s}
    {\!=11},
]
\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode`\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup
\!mark
\end{texhigh}
```

```
\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode`\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup
\!mark
```

代码 13

```
% \T 放在开头表示这是 TeX 正则表达式，匹配 token
% \T 大多数时候可以省略，但当模式以数字或 [ 开头
% 时，\T 不可省略，否则被当作行号
\begin{texhigh}[gobble=auto, % 检测空格并删除
  lexer-catcode*=
    {\T\c{catcode}\c{!}\!=11} % 注意字符转义
    {\c{endgroup}}
    {\!=11},
]
\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode`\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup
\!mark
\end{texhigh}
```

```
\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode`\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup
\!mark
```

代码 14

```

\begin{texhigh}[gobble=auto, % 检测空格并删除
lines={2,8}, % 只保留 [2, 8) 行
lexer-catcode*=
  {\T\c{catcode}\!\c{!}\=11} % 注意字符转义
  {\c{endgroup}}
  {\!=11},
]
  \kill this line
\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup
\!mark
\end{texhigh}

```

```

\def\!mark#1{MARK1} \!mark
\begingroup
  \catcode\!=11
  \def\!mark#1{MARK2} \!mark
  \endgroup \!mark
\endgroup

```

代码 15

```

% \THSetCharReplacement{\ }{\textvisiblespace} % texhigh 已设置
\THSetCharReplacement{\$}{\S} % 把 $ 替换为 \S
\texhighverb[char-replacements={\ \$}, % 设置哪些字符要替换
char-category*={symbol}{[\ \$]}{\mbox{\color{red}#1}}, % 修改颜色
] {\def\!a #1{\$#1\$} \!a {\ b}}

```

```

\texhighverb[char-replacements={a=A, b=B, {=}{,}}]{abcd=$}
\texhighverb[enhanced=false, char-replacements={a=A, b=B, {=}{,}}]{abcd=$}

```

```

\def\!a_#1{\$#1\$}_\!a_{\!b}

```

```

ABcd,$ abcd=$

```