高亮 TeX 和 LaTeX3 代码——使用 texhigh 宏包

雾月*

2024 年 10 月 25 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 1.2M 左右（37000 余行）的 `expl3-code.tex` 只需不到 0.4s，处理速度是 minted 宏包使用的 pygmentize 的 6 倍左右。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 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、LaTeX3 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 1

```
\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]{
```

*longaster@163.com

[1]https://github.com/Sophanatprime/texhigh-rs

```
  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 2

```
\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 3

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

\sys_get_shell:nnNTF

底纹：

代码 4

```
\makeatletter
\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 原语带有下划线）：

代码 5

```
\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
```

```
%%% 以下输出本文源码 %%%
% !TEX program=xelatex
```

```
% !TEX options=-synctex=1 -shell-escape -interaction=nonstopmode ↩
↪  -file-line-error "%DOC%"
\documentclass[zihao=-4]{ctexart}
\usepackage[hmargin=2cm,vmargin=2.4cm]{geometry}
\usepackage[many]{tcolorbox}
\usepackage[color,tikz]{texhigh}
\SetKeys[texhigh]{
  font=\ttfamily\xeCJKsetup{CJKecglue={\hskip 0pt plus 0.08\baselineskip}}
}
\tcbset{listing engine=texhigh}
\newcounter{example}
\newtcblisting[use counter=example, number format=\arabic]
  {examcode}[2][]{listing and text,
  title=代码 \thetcbcounter, enhanced,
  comment={#2},
  sharp corners=downhill, arc=12pt, %skin=bicolor,
  fontupper=\linespread{1}\selectfont, left=6pt,
  colback=blue!1!white, colframe=blue!75!black,colbacklower=white,
  segmentation style={draw=blue,thick,solid},
  attach boxed title to top right={yshift=-\tcboxedtitleheight},
  boxed title style={
    colframe=blue!75!black,colback=blue!15!white,
    sharp corners=downhill,arc=12pt,
  },
  coltitle=blue!90!black, fonttitle=\bfseries,
  before skip balanced=2bp plus .5\baselineskip,
  after skip balanced=2bp plus .5\baselineskip,
  breakable,
  #1
}
\begin{document}

\title{高亮 \TeX 和 \LaTeX3 代码——使用\textsf{texhigh} 宏包}
\author{雾月\thanks{longaster@163.com}}
\maketitle

\textsf{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\footnote{https://github.com/Sophanatprime/texhigh-rs},
处理 1.2M 左右（37000 余行）的 \texttt{expl3-code.tex} 只需不到 0.4s，
处理速度是 \textsf{minted} 宏包使用的 pygmentize 的 6 倍左右。
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。

\textsf{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 \texttt{--shell-escape}。

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

\textsf{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，\textsf{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、\LaTeX3 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
\texhighverb|\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 \texttt{basic} 样式的源码。

\textsf{texhigh} 利用 \textsf{tikz} 实现了渐变和底纹效果，同时也可直接集成到
\textsf{tcolorbox} 宏包中。只需要在加载 \textsf{texhigh} 之前加载这几个宏包。
```
\begin{examcode}[listing only]{}
\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]{
  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。
\end{examcode}
```

识别行内数学公式：
```
\begin{examcode}{}
\texhighverb!公式 $ \int_a^b x^2 dx = \frac{1}{3} x^3 |_a^b $!。
\end{examcode}
```

渐变：
```
\begin{examcode}[texhigh options={use-ctab=latex3}]{}
\texhighverb[style=tikz.gradient, use-ctab=latex3, config-file=config.cfg]| ↩
↪ \sys_get_shell:nnNTF|
\end{examcode}
```

底纹：

```
\begin{examcode}[texhigh options={use-ctab=latexcode}]{}
\makeatletter
\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|
}
\end{examcode}

中文命令识别 (\TeX 原语带有下划线):
\begin{examcode}[texhigh use ctab=cjk]{}
\begin{texhigh}[output=\jobname.texhigh, use-ctab=cjk]
  \def\好好好{中文 Good}
  \好好好\relax
\end{texhigh}
\end{examcode}

\bigskip
\noindent\texhighverb|%%% 以下输出本文源码 %%%|
\texhighfile[style=tikz.gradient, use-ctab=cjkl3]{\jobname.tex}
\noindent\texhighverb|%%% 以上是本文源码 %%%|

\vspace{1cm}
\noindent\texhighverb|%%%---- File: texhigh.sty ----%%%|
\texhighfile[use-ctab=latex3code, config-file=config.cfg]{texhigh.sty}

\vspace{1cm}
\noindent\texhighverb|%%%---- File: prelude.ths ----%%%|
\texhighfile[use-ctab=latexcode, config-file=config.cfg]{prelude.ths}

\end{document}
%%% 以上是本文源码 %%%


%%%---- File: texhigh.sty ----%%%
\ProvidesExplPackage{texhigh}{2024-10-20}{0.1.1}{highlight TeX string}
\def\texhigh@style@ext{ths}
\cs_new_protected:Npn \texhigh@inputstyle #1#2#3 % style, options, date
  { \@onefilewithoptions {#1}[{#2}][{#3}]\texhigh@style@ext }
\NewDocumentCommand \texhighloadstyle { O{} O{} m }
```

```
  { \clist_map_inline:nn {#3} { \texhigh@inputstyle {##1}{#1}{#2} } }

\cs_new_protected:Npn \THnl { \TH@nl } %: new line;
\cs_new_protected:Npn \THnlPlain { \TH@nl }
\cs_new_protected:Npn \THin #1 { \TH@in {#1} } %: indent;
\cs_new_protected:Npn \THinPlain #1 { \TH@in {#1} } %: indent;
\cs_new_protected:Npn \THbp #1 { \use:c { TH@bp@\texhigh@fallback{bp}{#1} } ↩
↪  } %: break point;
\cs_new_protected:Npn \THbpPlain #1 { \TH@bp@PLAIN } %: break point;
\cs_new_protected:Npn \THcs #1#2#3 %: control sequence;
  {
    \exp_last_unbraced:Ne \use:c
      { { TH@cs@\texhigh@fallback{cs}{#1} } { \texhigh@normalize@cs {#2} } { ↩
↪  \texhigh@normalize@cs {#3} } }
  }
\cs_new_protected:Npn \THcsPlain #1#2#3
  { \exp_args:Nee \TH@cs@PLAIN { \texhigh@normalize@cs {#2} } {  ↩
↪ \texhigh@normalize@cs {#3} } }
\cs_new_protected:Npn \THch #1#2 %: character;
  { \exp_args:Nee \use:c { TH@ch@\texhigh@fallback{ch}{#1} } {  ↩
↪ \texhigh@normalize@cs {#2} } }
\cs_new_protected:Npn \THchPlain #1#2
  { \exp_args:Ne \TH@ch@PLAIN { \texhigh@normalize@cs {#2} } } %: character;
\cs_new_protected:Npn \THrs #1 { \use:c { TH@rs@\texhigh@fallback{rs}{#1} } ↩
↪  } %: range start;
\cs_new_protected:Npn \THrsPlain #1 { \TH@rs@PLAIN }
\cs_new_protected:Npn \THre #1 { \use:c { TH@re@\texhigh@fallback{re}{#1} } ↩
↪  } %: range end;
\cs_new_protected:Npn \THrePlain #1 { \TH@re@PLAIN }
\cs_new_protected:Npn \THst #1#2 { \use:c { TH@st@\texhigh@fallback{st}{#1} ↩
↪  } {#2} } %: string (Letters and Others)
\cs_new_protected:Npn \THstPlain #1 { \TH@st@PLAIN }
\cs_new_protected:Npn \THes #1 { \use:c { TH@es@\texhigh@fallback{es}{#1} } ↩
↪  } %: escaped start;
\cs_new_protected:Npn \THesPlain #1 { \TH@es@PLAIN }
\cs_new_protected:Npn \THee #1 { \use:c { TH@ee@\texhigh@fallback{ee}{#1} } ↩
↪  } %: escaped end;
\cs_new_protected:Npn \THeePlain #1 { \TH@ee@PLAIN }
\cs_new_protected:Npn \THpn #1#2 { \use:c { TH@pn@\texhigh@fallback{pn}{#1} ↩
↪  } {#2} } %: punctuation;
\cs_new_protected:Npn \THpnPlain #1#2 { \TH@pn@PLAIN {#2} } %: punctuation;
\cs_new:Npn \texhigh@fallback #1#2
  {
```

```
    \cs_if_exist:cTF { TH@#1@#2 } {#2}
      {
        \seq_if_exist:cT { l__texhigh_#1/#2_seq }
          { \seq_map_tokens:cn { l__texhigh_#1/#2_seq } { \__texhigh_find:nn ↩
↪  {#1} } }
        \use:n { \__texhigh_dotted_fallback:nn {#1} {#2} }
      }
  }
\cs_new:Npn \__texhigh_find:nn #1#2
  { \cs_if_exist:cT { TH@#1@#2 } { \seq_map_break:n { \use_i:nnn {#2} } } }
\cs_new:Npn \__texhigh_dotted_fallback:nn #1#2 { ? }
\cs_new:Npx \texhigh@normalize@cs #1
  {
    \exp_not:N \__texhigh_normalize_cs:w #1 "
    \exp_not:N \q_recursion_tail
    \c_space_tl
    \exp_not:N \q_recursion_stop
  }
\cs_new:Npn \__texhigh_normalize_cs:w #1 "#2~%
  {
    #1
    \quark_if_recursion_tail_stop:n {#2}
    \char_generate:nn { "#2 } { 12 }
    \__texhigh_normalize_cs:w
  }

\cs_new_eq:NN \texhigh@replicate \prg_replicate:nn
\cs_new_protected:Npn \texhigh@pdfliteral { \__kernel_backend_literal_pdf:e }
\cs_new_eq:NN \TH@letcs \cs_set_eq:cc

\NewDocumentCommand \texhighsetclassfallback { s m m m } % type, class, fallback
  {
    \bool_if:nTF {#1}
      { \seq_set_from_clist:cn { l__texhigh_#2/#3_seq } {#4} }
      {
        \seq_if_exist:cF { l__texhigh_#2/#3_seq } { \seq_clear:c { ↩
↪  l__texhigh_#2/#3_seq } }
        \seq_set_from_clist:Nn \l__texhigh_tmp_seq {#4}
        \seq_concat:ccc { l__texhigh_#2/#3_seq } { l__texhigh_#2/#3_seq } { ↩
↪  l__texhigh_tmp_seq }
      }
  }
\protected\long\def\THSaveStyle #1 {
  \expandafter\relax\csname texhigh@savestyle\expanded{\endcsname{#1}}
}
```

```
\def\texhigh@savestyle#1#2 {
  \tl_set:cn {texhigh@style/#1}{
    \let\texhigh@saved@name\@currname \let\@currname\@empty #2\let\@currname
↪ \texhigh@saved@name
  }
}
\def\texhigh@curr@style{}
\protected\long\def\THUseSavedStyle #1
  {\expandafter\relax\csname texhigh@usesavedstyle\expanded{\endcsname{#1}}}
\long\def\texhigh@usesavedstyle#1{
  \@ifundefined{texhigh@style/#1}
    {\PackageWarning{texhigh}{Unknown texhigh style #1.}}
    {\def\texhigh@curr@style{#1}\@nameuse{texhigh@style/#1}}
}
\cs_new_protected:Npn \THSetPlainStyle #1
  {
    \str_if_eq:eeTF { #1 } { * }
      { \clist_map_inline:nn { bp,cs,ch,rs,re,st,es,ee,pn } }
      { \clist_map_inline:nn {#1} }
        { \cs_set_eq:cc { TH ##1 } { TH ##1 Plain } }
  }

\ExplSyntaxOff

\newcommand\THSetClassBP[3][\@currname]{\@namedef{TH@bp#1@#2}{#3}}
\newcommand\THLetClassBP[3][\@currname]{\TH@letcs{TH@bp#1@#2}{TH@bp#1@#3}}
\newcommand\THSetClassCS[3][\@currname]{\@namedef{TH@cs#1@#2}##1##2{#3}}
\newcommand\THLetClassCS[3][\@currname]{\TH@letcs{TH@cs#1@#2}{TH@cs#1@#3}}
\newcommand\THSetClassCH[3][\@currname]{\@namedef{TH@ch#1@#2}##1{#3}}
\newcommand\THLetClassCH[3][\@currname]{\TH@letcs{TH@ch#1@#2}{TH@ch#1@#3}}
\newcommand\THSetClassRS[3][\@currname]{\@namedef{TH@rs#1@#2}{#3}}
\newcommand\THLetClassRS[3][\@currname]{\TH@letcs{TH@rs#1@#2}{TH@rs#1@#3}}
\newcommand\THSetClassRE[3][\@currname]{\@namedef{TH@re#1@#2}{#3}}
\newcommand\THLetClassRE[3][\@currname]{\TH@letcs{TH@re#1@#2}{TH@re#1@#3}}
\newcommand\THSetClassST[3][\@currname]{\@namedef{TH@st#1@#2}{#3}}
\newcommand\THLetClassST[3][\@currname]{\TH@letcs{TH@st#1@#2}{TH@st#1@#3}}
\newcommand\THSetClassES[3][\@currname]{\@namedef{TH@es#1@#2}{#3}}
\newcommand\THLetClassES[3][\@currname]{\TH@letcs{TH@es#1@#2}{TH@es#1@#3}}
\newcommand\THSetClassEE[3][\@currname]{\@namedef{TH@ee#1@#2}{#3}}
\newcommand\THLetClassEE[3][\@currname]{\TH@letcs{TH@ee#1@#2}{TH@ee#1@#3}}
\newcommand\THSetClassPN[3][\@currname]{\@namedef{TH@pn#1@#2}##1{#3}}
\newcommand\THLetClassPN[3][\@currname]{\TH@letcs{TH@pn#1@#2}{TH@pn#1@#3}}
```

```
\long\def\TH@nl{\leavevmode\par}
\long\@namedef{TH@in}#1{\texhigh@replicate{#1}{ }}
\long\@namedef{TH@bp@?}{\hskip\z@skip}
\long\def\TH@bp@PLAIN{\hskip\z@skip}
\long\@namedef{TH@cs@?}#1#2{\mbox{#1#2}}
\long\def\TH@cs@PLAIN#1#2{\mbox{#1#2}}
\long\@namedef{TH@ch@?}#1{\mbox{#1}}
\long\def\TH@ch@PLAIN#1{\mbox{#1}}
\long\@namedef{TH@rs@?}{\begingroup}
\long\def\TH@rs@PLAIN{\begingroup}
\long\@namedef{TH@re@?}{\endgroup}
\long\def\TH@re@PLAIN{\endgroup}
\long\@namedef{TH@st@?}#1{{#1}}
\long\def\TH@st@PLAIN#1{{#1}}
\long\@namedef{TH@es@?}{\begingroup}
\long\def\TH@es@PLAIN{\begingroup}
\long\@namedef{TH@ee@?}{\endgroup}
\long\def\TH@ee@PLAIN{\endgroup}
\long\@namedef{TH@pn@?}#1{\expandafter\@gobble\string#1}
\long\def\TH@pn@PLAIN#1{\expandafter\@gobble\string#1}

\RequirePackage{verbatim}
\ExplSyntaxOn

\iow_new:N \g__texhigh_verb_iow
\cs_new_protected:Npn \__texhigh_verb_text:nn #1#2
  {
    \iow_open:N \g__texhigh_verb_iow {#1}
    \iow_now:Nn \g__texhigh_verb_iow {#2}
    \iow_close:N \g__texhigh_verb_iow
  }
\cs_new_protected:Npn \__texhigh_verb_start:n #1
  {
    \@bsphack
    \iow_open:Nn \g__texhigh_verb_iow {#1}
    \let\do\@makeother\dospecials
    \catcode`\^^M\active
    \def\verbatim@processline
      { \iow_now:Ne \g__texhigh_verb_iow {\the\verbatim@line} }
    \verbatim@start
  }
\cs_new_protected:Npn \__texhigh_verb_end:
  {
    \iow_close:N \g__texhigh_verb_iow
    \@esphack
```

```
  }

\cs_new_protected:Npn \texhighdefstyle #1#2
  { \keys_define:nn { texhigh } { #1 .meta:nn = { texhigh } {#2} } }
\keys_define:nn { texhigh }
  {
    filename .tl_set:N = \l__texhigh_fn_tl ,
    filename .initial:n = \jobname.texhigh.verb ,
    filename .groups:n = command ,
    banner .bool_set:N = \l__texhigh_banner_bool ,
    banner .groups:n = command ,
    use-ctab .clist_set:N = \l__texhigh_current_ctab_clist ,
    use-ctab .groups:n = command ,
    ctab-file .clist_set:N = \l__texhigh_ctab_file_clist ,
    ctab-file .groups:n = command ,
    __config .clist_set:N = \l__texhigh_config_clist ,
    config-file .clist_set:N = \l__texhigh_config_file_clist ,
    config-file .groups:n = command ,
    output .tl_set:N = \l__texhigh_output_tl ,
    output .groups:n = command ,

    font .tl_set:N = \l__texhigh_font_tl ,
    font .initial:n = \ttfamily ,
    style .code:n = \clist_map_inline:nn {#1} { \THUseSavedStyle {##1} } ,
    style .groups:n = format ,
    this-cs .code:n = {
        \clist_put_right:Nn \l__texhigh_config_clist { { cs_categories.this ~ '.+' } }
        \THSetClassCS[]{this}{#1{{##1}{##2}}} % #1=tokens, ##1=escape char, ##2=cs name
    },
    this-cs .groups:n = format ,
  }
\cs_new_protected:Npn \__texhigh_options:
  {
    \l__texhigh_font_tl
  }
\NewDocumentCommand\texhighverb{ s +O{} +v }
  {
    \group_begin:
    \keys_set:nn { texhigh } {#2}
    \__texhigh_options:
    \__texhigh_get_text:n {#3}
    \group_end: \ignorespaces
  }
```

```
\NewDocumentCommand\texhighfile{ +O{} m }
  {
    \par
    \group_begin:
    \__texhigh_env_init:
    \tl_clear:N \l__texhigh_output_tl
    \keys_set:nn { texhigh } {#1}
    \__texhigh_options:
    \tl_set:Nn \l__texhigh_fn_tl {#2}
    \__texhigh_get_last:
    \par
    \group_end:
  }
\NewDocumentCommand\texhighinput{ +O{} m }
  {
    \par
    \group_begin:
    \__texhigh_env_init:
    \keys_set:nn { texhigh } {#1}
    \__texhigh_options:
    \file_input:n {#2}
    \par
    \group_end:
  }
\NewDocumentEnvironment{texhigh}{+O{}}
  {
    \par \__texhigh_env_init:
    \keys_set:nn { texhigh } {#1}
    \group_begin: \__texhigh_verb_start:n \l__texhigh_fn_tl
  }
  {
    \__texhigh_verb_end: \group_end:
    \__texhigh_options: \__texhigh_get_last: \par
  }

\cs_new_protected:Npn \__texhigh_env_init:
  {
    \sloppy \hbadness\@M
    \dim_set:Nn \parindent { 0pt }
    \linespread{1} \selectfont
  }
\cs_new_protected:Npn \__texhigh_get_text:n #1
  {
    \str_if_in:nnTF {#1} { " }
      {
```

```
        \__texhigh_verb_text:nn \l__texhigh_fn_tl {#1}
        \tex_input:D | " \__texhigh_args_with_file:n \l__texhigh_fn_tl "
      }
      { \tex_input:D | " \__texhigh_args_with_text:n {#1} " }
  }
\cs_new_protected:Npn \__texhigh_get_last:
  {
    \tl_if_empty:NTF \l__texhigh_output_tl
      {
        \tex_input:D | " \__texhigh_args_with_file:n \l__texhigh_fn_tl "
      }
      {
        \sys_shell_now:e { \__texhigh_args_with_file:n \l__texhigh_fn_tl }
        \file_input:V \l__texhigh_output_tl
      }
  }
\cs_new:Npn \__texhigh_args_with_text:n #1
  {
    texhigh \c_space_tl
    \__texhigh_args:
    --text \c_space_tl '#1'
  }
\cs_new:Npn \__texhigh_args_with_file:n #1
  {
    texhigh \c_space_tl
    \__texhigh_args:
    --file \c_space_tl #1
  }
\cs_new:Npn \__texhigh_args:
  {
    \bool_if:NF \l__texhigh_banner_bool { --no-banner \c_space_tl }
    \clist_if_empty:NF \l__texhigh_current_ctab_clist
      {
        --current-ctab \c_space_tl
        \clist_use:Nn \l__texhigh_current_ctab_clist { --current-ctab ~ }
        \c_space_tl
      }
    \clist_if_empty:NF \l__texhigh_ctab_file_clist
      {
        --ctab-file \c_space_tl
        \clist_use::Nn \l__texhigh_ctab_file_clist { --ctab-file ~ }
        \c_space_tl
      }
    \clist_if_empty:NF \l__texhigh_config_file_clist
      {
```

```
        --config-file \c_space_tl
        \clist_use:Nn \l__texhigh_config_file_clist { --config-file ~ }
        \c_space_tl
      }
    \clist_if_empty:NF \l__texhigh_config_clist
      {
        --config \c_space_tl
        \clist_use:Nn \l__texhigh_config_clist { --config ~ }
        \c_space_tl
      }
    \tl_if_empty:NF \l__texhigh_output_tl
      { --output \c_space_tl \l__texhigh_output_tl \c_space_tl }
  }

\DeclareKeys[texhigh/options/prelude]{
  color .if = @texhighload@color ,
  tikz .if = @texhighload@tikz ,
}

\keys_define:nn { texhigh/options }
  {
    color .meta:n = { prelude/color } ,
    tikz .meta:n = { prelude/tikz } ,
  }

\ProcessKeyOptions[texhigh/options]

\texhighloadstyle{texhigh.prelude}


%%%---- File: prelude.ths ----%%%
```