Package **highlightlatex** manual

Vincent Kuhlmann
vincent.kuhlmann@hotmail.com

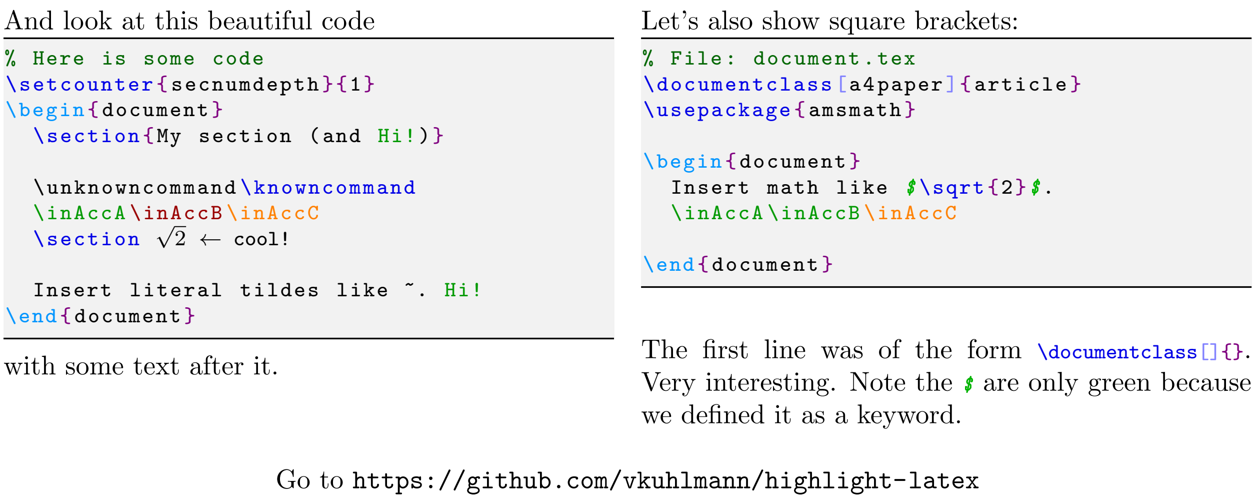
March 23, 2021

Abstract

This package provides colored syntax highlighting for LaTeX source code, without aid from outside LaTeX. This is in response to the general trend that people often fall back to verbatim for displaying code. This package aims to make good looking LaTeX source code feasible for all users. For this, it builds further on the generic 'listings' package. An example output is shown in Figure 1 .

Repository

https://github.com/vkuhlmann/highlight-latex



And look at this beautiful code

```
% Here is some code
\setcounter{secnumdepth}{1}
\begin{document}
  \section{My section (and Hi!)}

  \unknowncommand\knowncommand
  \inAccA\inAccB\inAccC
  \section √2 ← cool!

  Insert literal tildes like ~. Hi!
\end{document}
```

with some text after it.

Let's also show square brackets:

```
% File: document.tex
\documentclass[a4paper]{article}
\usepackage{amsmath}

\begin{document}
  Insert math like $\sqrt{2}$.
  \inAccA\inAccB\inAccC

\end{document}
```

The first line was of the form `\documentclass[]{}`. Very interesting. Note the $ are only green because we defined it as a keyword.

Go to https://github.com/vkuhlmann/highlight-latex

Figure 1: Output of 'demo.tex'.

Contents

1 Getting started

After having added the package, you can add LaTeX in two ways.

1.1 Inline style

EXAMPLE

```
Your file begins with a line of the form \hll|\documentclass[]{}|. The
square brackets ...
```

The first non-space character following `\hll` delimits the argument to this command.

1.2 Block style

EXAMPLE

```
Your basic document now looks like
\begin{highlightblock}[gobble=2]
    \documentclass[a4paper]{article}
    \begin{document}
    Hello world!
    \end{document}
\end{highlightblock}
```

To prevent indentation of our `highlightblock` (here one tab) to be shown as part of the code, the `gobble` parameter strips them off. Play around with it until everything looks right. I recommend to set this value globally using `\def\defaultgobble{2}`. You can still override it on a per-block basis, if necessary.

There are situations where width of the block could run out of the page. For example, when using beamer and storing a block as described in the section ‘Fragile breaking situations’, the normal full-width of a slide is assumed. If you use multiple columns, set the `linewidth` on the `highlightblock`. This can be a fraction of the total slide width available, `0.6\textwidth` is 60% of the width, or an absolute value, like `10em`, which seems to equal 20 characters.

There are more keys you can provide. Check the `listings` package documentation [1] for options available to the `lstlisting`-environment and `lstset` command.

2 Macro `\updatehighlight`

2.1 Adding a command to a highlighting rule

By default, only some LaTeX commands will be highlighted in blue. If there are others you need, like `\tableofcontents` and `\figref`, update the highlighting rules:

USE

```
\updatehighlight{
    name = default,
    add = {
        \tableofcontents, \figref
    }
}
```

The change will only affect code after it. I recommend issuing `updatehighlight` in your preamble (before the `\begin{document}`). In some situations you might want to change things mid-document. That’s possible too.

[1] https://www.ctan.org/pkg/listings

2.2 Custom highlighting rules

2.2.1 Example

As shown in `demo.tex`, you can put any command or keyword you want to highlight in a different color. You do this with

EXAMPLE

```
\updatehighlight{
    % The name allows you to modify the style later.
    name = spotlight,
    color = orange,
    add = {
        \tableofcontents
    }
}
```

You can use the `xcolor` syntax for describing colors as well. If you find the orange too bright, you can replace it with `orange!90!black`: 90% orange, remaining is black. For more information on color definitions and name, refer to LaTeX/Colors on Wikibooks [2] .

2.2.2 Specification

The argument to `\updatehighlight` is a key-value list. Keys are processed sequentially. For example, use `color` before `add` rather than after it, and a key can appear multiple times. Each one will be processed. You can merge any two `\updatehighlight` in one. No need to close and reopen `\updatehighlight` for each highlighting rule.

You might be tempted to add a blank line for clarity; that means a new paragraph to LaTeX, don't do it. Instead, just put a line with only a `\%` sign. Spacing within the argument is often irrelevant. If you need a comma in the value, surround your value with braces.

name

Creates or modifies a named rule. This key is optional.

The default keys are `default`, which includes a bunch of basic commands, and has by default a dark blue color, and `structure`, which consists of `\begin` and `\end` and prints them in light blue.

classoffset

Sets the `listings` classoffset manually. Try to avoid this. Use **name** to refer to existing rules instead.

add

Adds a command (`\mycommand`) or keyword (`Hi!`) to the current rule. The value can contain multiple values by opening braces, and comma separating values within them.

remove

Removes a commands or keywords from the current rule. The value can contain multiple values by opening braces, and comma separating values within them.

[2] `https://en.wikibooks.org/wiki/LaTeX/Colors`

clear

Removes all commands and keywords from the current rule. Use without value, for example

EXAMPLE

```
\updatehighlight{
    name = default,
    clear
}
```

color

Specifies a color for the rule. Equivalent to specifying `style` instead, with value `\color{value}` where `value` is the value for the **color** key. So `color=red` and `style=\color{red}` are equivalent.

style

Specifies a style for the rule. A rule can have only one style. If you specify a style after `add` or `remove`, this starts a new (unnamed) rule. In practice, the only style which will probably work for you is just a color. For that, using the 'color' key is just a bit easier and neater. But hey, you have the option to set whatever style you want. :)

3 Global settings

There are some global parameters involved in the appearance:

OVERVIEW

```
\colorlet{curlyBrackets}{red!50!blue}
\colorlet{squareBrackets}{blue!50!white}
\colorlet{codeBackground}{gray!10!white}
\colorlet{comment}{green!40!black}
\def\defaultgobble{0}
```

Each line can be set independent of each other, and each shows its default value.

There are package options you can use as well:

frame (default: `lines`)

Specifies the frame you want around code. My favorites are `lines` and `none`. Check the listings package documentation [3] for all possibilities.

noframe (use without value)

Equivalent to `frame=none`.

styleanywhere (use without value)

Overrides the default behavior that `style` starts a new style after commands like `add` and `remove`.

4 Fragile breaking situations (like beamer frames)

When passing command arguments around, or storing environment content, LaTeX interprets all characters. This includes seeing `\maketitle` in `\hl1|\maketitle|` as a real command. To prevent this behavior, everything from `\verb`, to the `verbatim`-environment, to the `listings` package (which this package depends on) temporarily changes the interpretation of characters that are still to be read. The blackslash before maketitle in `\hl1|\maketitle|` will be read as 'just text' (a *letter* technically).

[3] `https://www.ctan.org/pkg/listings`

When content has already been interpreted, like the `frame`-environment in `beamer` does, this trick can't be done anymore. Instead, you either need to *escape* code, or *pre-process* the code outside a fragile breaking situation.

Escaping is done by preceding the special character with a backslash. For example, `\hll|\documentclass[]{}|` becomes `\hll|\\documentclass[]\{\}|`.

For large code blocks, this is undesirable. Therefore, the package provides for a companion to the `highlightblock`-environment: surround it with a `saveblock`-environment which takes a single argument: a name to assign it. We use it to refer to it later. For example:

EXAMPLE

```
\begin{saveblock}{basicfigure}
    \begin{highlightblock}[linewidth=0.6\textwidth]
        \begin{figure}
            \includegraphics
            [width=0.9\linewidth]
            {myPlot.pdf}

            \caption{My plot}
            \label{fig:myplot}
        \end{figure}
    \end{highlightblock}
\end{saveblock}
```

Do this outside a fragile breaking situation. (For the `frame`-environment example, that means just before the `frame` for example.) Then, where you want to use it, use `\useblock{basicfigure}`. There is also a variant `\consumeblock{basicfigure}`. If you save many blocks, these will all remain loaded in memory till your PDF has fully generated. The `\consumeblock` works like `\useblock`, except the saved block is deleted from memory after its use. Note this can also result in unexpected behavior, for example animations in a beamer frame might need the code line to be executed multiple times. Use `\useblock` when you can't make the guarantee the last use of a block.

There is a separate demo for fragile breaking situations. You can find it at `deamerdemo/deamerdemo.tex`.

5 Adding extra space

By default, highlight-latex follows an approach where it minimizes spacing. This gives you full control over how tight or spacious your document looks. Just use commands like `\medskip` to add extra spacing. The package doesn't currently include an option to do it everywhere automatically.

6 License & Credits

The package is available under MIT License. See LICENSE.txt.

Thanks for minor fixes:

gemmaro

For any bug, feature request, unclarity, or whatever else related to this package, you're welcome to open an issue or pull-request. Issues can be opened on

URL

```
https://github.com/vkuhlmann/highlight-latex/issues
```

Thanks for contributing!

7 Implementation

```
\NeedsTeXFormat{LaTeX2e}[1994/06/01]
\ProvidesPackage{highlightlatex}[2021/03/17 highlightlatex v1.0.0 + Dev]

% Repository: https://github.com/vkuhlmann/highlight-latex
% Version: v1.0.0 + Dev

% Copyright (c) 2021 Vincent Kuhlmann
%
% Permission is hereby granted, free of charge, to any person
% obtaining a copy of this software and associated documentation
% files (the "Software"), to deal in the Software without
% restriction, including without limitation the rights to use,
% copy, modify, merge, publish, distribute, sublicense, and/or sell
% copies of the Software, and to permit persons to whom the
% Software is furnished to do so, subject to the following
% conditions:
%
% The above copyright notice and this permission notice shall be
% included in all copies or substantial portions of the Software.
%
% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
% EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
% OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
% NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
% HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
% WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
% FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
% OTHER DEALINGS IN THE SOFTWARE.

\RequirePackage{listings}
\RequirePackage{xcolor}

\RequirePackage{xkeyval}
\RequirePackage{etoolbox}

%% Snippet source:
%% https://tex.stackexchange.com/questions/406015/defining-macro-gsetlength-as-global-setlength-reliable
\gdef\gsetlength#1#2{%
    \begingroup
        \setlength\skip@{#2}%
        \global#1=\skip@
    \endgroup
}


\def\@highlight@frame{none}
\newlength{\hll@offset@fixskip}
\setlength{\hll@offset@fixskip}{0pt}

\newif\ifhllStyleAnywhere
\newcommand\defaultgobble{0}
\newif\ifhllDebugFrame
%\hllDebugFrametrue
\newlength\hllBeforeSkip
\setlength\hllBeforeSkip{2pt}
\newlength\hllAfterSkip
\setlength\hllAfterSkip{0pt}
```

7.1 Package options

```
\DeclareOptionX{frame}{
    \setframe{#1}
}

\DeclareOptionX{noframe}{
    \setframe{none}
}

\def\setframe#1{%
    \def\frameval{#1}%
    \ifx\relax#1\relax
        \def\frameval{lines}%
    \fi
    \def\linesvalue{lines}%
    %
    \xdef\@highlight@frame{\frameval}%
    %
    \ifx\frameval\linesvalue
        \gsetlength{\hll@offset@fixskip}{6pt}%
    \else
        \gsetlength{\hll@offset@fixskip}{0pt}%
    \fi
}

\DeclareOptionX{styleanywhere}{
    \hllStyleAnywheretrue
}

\DeclareOptionX*{\PackageWarning{highlightlatex}{‘\CurrentOption’ ignored}}
\ExecuteOptionsX{frame=lines}

\ProcessOptionsX\relax
```

7.2 Initialization

```
\colorlet{curlyBrackets}{red!50!blue}
\colorlet{squareBrackets}{blue!50!white}
\colorlet{codeBackground}{gray!10!white}
\colorlet{comment}{green!40!black}

\colorlet{accDefault}{blue!90!black}
\definecolor{accStructure}{RGB}{0,149,255}

\colorlet{accentB}{green!60!black}

\catcode‘\{=11
\let\myopenbrace={
\catcode‘\{=1

\catcode‘\}=11
\let\myclosebrace=}
\catcode‘\}=2

\lstdefinelanguage{ColoredLaTeX}
[LaTeX]{TeX}{
    backgroundcolor=\color{codeBackground},
    keywordstyle=\bfseries\color{accDefault},
    basicstyle=\footnotesize\ttfamily,
    commentstyle=\color{comment},
    numbers=none,
    tabsize=4,
    mathescape=false,
    alsoletter={$@_!|?$},
%
    classoffset=0,
    keywordstyle=\color{accDefault},
    texcsstyle=*\color{accDefault},
    moretexcs={subsection,thesubsection,thesection,theoremstyle},
    deletetexcs={begin,end},
%
    classoffset=1,
    keywordstyle=\color{accStructure},
    texcsstyle=*\color{accStructure},
    moretexcs={begin,end},
%
    classoffset=2,
    escapeinside=~~,
%
    classoffset=0,
    %literate={textA}{textB}{3}%
    %https://tex.stackexchange.com/questions/172945/coloring-in-listing
    literate={\{}{{\textcolor{curlyBrackets}{\myopenbrace}}}{1}
        {\}}{{\textcolor{curlyBrackets}{\myclosebrace}}}{1}
        {[}{{\textcolor{squareBrackets}{[}}}{1}
        {]}{{\textcolor{squareBrackets}{]}}}{1}%
    %        {(}{{\textcolor{delimiterColor}{(}}}{1}
    %        {)}{{\textcolor{delimiterColor}{)}}}{1}%
    %texcsstyle=\color{blue}
}
\lstset{language=ColoredLaTeX}

\def\hll@assignlabel#1#2{%
    \expandafter\gdef\csname hll@labeltoclassoffset@#1\endcsname{#2}%
}

\hll@assignlabel{default}{0}
\hll@assignlabel{structure}{1}

```

7.3 Options

```
\newcommand{\hllDelimDollar}[1][\color{green!40!black}]{%
    \lstset{
        classoffset=2,
        moredelim=**[s][#1]{$}{$},
        classoffset=0
    }%
}

\newcommand{\hllDelimParen}[1][\color{green!40!black}]{%
    \lstset{
        classoffset=2,
        moredelim=**[s][#1]{\(}{\)},
        classoffset=0
    }%
}

```

7.4 Block and inline markup

```
\newcommand{\hllPrepareBlock}{%
    \lstset{framesep=0.1cm,framerule=0.6pt}%
    \def\setHighlightFrame##1{%
        \lstset{frame=##1}%
    }%
    \expandafter\setHighlightFrame\expandafter{\@highlight@frame}%
    \lstset{belowskip=0pt,aboveskip=0pt,belowcaptionskip=0pt,gobble=\defaultgobble}%
    \setlength\parskip{0pt}%
    \setlength\parindent{0pt}%
}

\lstnewenvironment{highlightblock}[1][]
{%
    \vspace{\hllBeforeSkip}%
    \begingroup
    \hllPrepareBlock\lstset{#1}%
}{%
    \nointerlineskip
    \endgroup
    \vspace{\hllAfterSkip}%
}

\lstnewenvironment{hllblock}[1][]
{%
    \vspace{\hllBeforeSkip}%
    \begingroup
    \hllPrepareBlock\lstset{#1}%
}{%
    \endgroup
    \vspace{\hllAfterSkip}%
}

\def\hll{\lstinline}

```

7.5 Block saving

```
\newenvironment{saveblock}[1]{%
    \expandafter\ifx\csname @codebox@#1\endcsname\relax
        \expandafter\newbox\csname @codebox@#1\endcsname\relax
    \fi
    %
    %% The trick from latex/base/latex.ltx
    \edef\@bracehack{%
        \endgroup
        \expandafter\setbox\csname @codebox@#1\endcsname\vbox{%
            \begingroup\aftergroup}%
        \def\noexpand\@currenvir{\@currenvir}%
        \def\noexpand\@currenvline{\on@line}}%
    \@bracehack
    \begingroup\vspace{\hll@offset@fixskip}
    \ignorespaces%
}{%
    \endgroup
}

\newcommand\useblock[1]{%
    \ifhllDebugFrame
        \par
        {%
            \setlength{\fboxsep}{0pt}%
            \fcolorbox{orange}{red}{\expandafter\usebox\csname @codebox@#1\endcsname}%
        }%
        \par
    \else
        \par\expandafter\usebox\csname @codebox@#1\endcsname\par
    \fi
}

\newcommand\consumeblock[1]{%
    \ifhllDebugFrame
        \par
        {%
            \setlength{\fboxsep}{0pt}%
            \setlength{\fboxrule}{1pt}%
            \fcolorbox{orange}{red}{\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax}%
        }
        \par
        \expandafter\let\csname @codebox@#1\endcsname\relax
    \else
        \par\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax\par
        \expandafter\let\csname @codebox@#1\endcsname\relax
    \fi
}

```

7.6 Macro \updatehighlight

```
\def\hll@update@label{}

\newcounter{hll@update@state}

\newcounter{hll@update@nextclassoffset}
\setcounter{hll@update@nextclassoffset}{5}

\newcounter{hll@update@classoffset}

\def\hll@update@fixclassoffset{%
    \ifnum\value{hll@update@state}<1\relax
    \setcounter{hll@update@classoffset}{\thehll@update@nextclassoffset}%
    \stepcounter{hll@update@nextclassoffset}%
    \setcounter{hll@update@state}{1}%
    \fi
}

\newcommand{\updatehighlight}[1]{%
    \setcounter{hll@update@state}{-1}%
    \gdef\hll@update@label{}%
    \setkeys{updatehighlight}{#1}%
    %
    \lstset{classoffset=0}%
}

```

7.6.1 Key ‘name’

```
\define@key{updatehighlight}{name}{%
    \hll@update@setlabel{#1}%
}

\def\hll@update@setlabel#1{
    \gdef\hll@update@label{#1}%
    \expandafter\def\expandafter\@capturedlabel\expandafter{\csname hll@labeltoclassoffset@#1\endcsname}%
    \expandafter\unless\expandafter\ifx\@capturedlabel\relax\relax
        % Label exists
        \edef\@val{\@capturedlabel}%
        \setcounter{hll@update@classoffset}{\@val}%
        \setcounter{hll@update@state}{1}%
    \else
        % Allocate new label
        \edef\@val{\thehll@update@nextclassoffset}%
        \setcounter{hll@update@classoffset}{\@val}%
        \stepcounter{hll@update@nextclassoffset}%
        \setcounter{hll@update@state}{1}%
        %
        \expandafter\xdef\@capturedlabel{\@val}%
    \fi
}

```

7.6.2 Key ‘classoffset’

```
\define@key{updatehighlight}{classoffset}{%
    \setcounter{hll@update@classoffset}{#1}%
    \setcounter{hll@update@state}{1}%
}
```

7.6.3 Key ‘add’

```
\define@key{updatehighlight}{add}{%
    \hll@update@add{#1}%
}

\def\hll@update@add#1{%
    \hll@update@fixclassoffset
    \def\@inv##1{\forcsvlist{\hll@update@additem{##1}}{#1}}%
    \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
    %
    \setcounter{hll@update@state}{2}%
}

\def\hll@update@additem#1#2{%
    \if\noexpand#2\relax
        \edef\keywordpure{\expandafter\@gobble\string#2}%
        \def\callwithpure##1{\hll@update@addmacro{#1}{##1}}%
        \expandafter\callwithpure\expandafter{\keywordpure}%
    \else
        \hll@update@addkeyword{#1}{#2}%
    \fi
}

\def\hll@update@addmacro#1#2{%
    \lstset{
        classoffset=0,deletetexcs={#2},
        classoffset=1,deletetexcs={#2},
        classoffset=#1,moretexcs={#2},
        classoffset=0
    }%
}

\def\hll@update@addkeyword#1#2{%
    \lstset{
        classoffset=0,deletekeywords={#2},
        classoffset=1,deletekeywords={#2},
        classoffset={#1},otherkeywords={#2},morekeywords={#2},
        classoffset=0
    }%
}

```

7.6.4 Key 'remove'

```
\define@key{updatehighlight}{remove}{%
    \hll@update@remove{#1}%
}

\def\hll@update@remove#1{%
    \hll@update@fixclassoffset
    \def\@inv##1{\forcsvlist{\hll@update@removeitem{##1}}{#1}}%
    \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
    %
    \setcounter{hll@update@state}{2}%
}

\def\hll@update@removeitem#1#2{%
    \if\noexpand#2\relax
        \edef\keywordpure{\expandafter\@gobble\string#2}%
        \def\callwithpure##1{\hll@update@removemacro{#1}{##1}}%
        \expandafter\callwithpure\expandafter{\keywordpure}%
    \else
        \hll@update@removekeyword{#1}{#2}%
    \fi
}

\def\hll@update@removemacro#1#2{%
    \lstset{
        classoffset={#1},
        deletetexcs={#2},
        classoffset=0
    }%
}

\def\hll@update@removekeyword#1#2{%
    \lstset{
        classoffset={#1},
        deletekeywords={#2},
        classoffset=0
    }%
}

```

7.6.5 Key 'style'

```
\define@key{updatehighlight}{style}{%
```

```
    \hll@update@fixclassoffset
    \unless\ifhllStyleAnywhere
        \ifnum\value{hll@update@state}>1\relax
            \setcounter{hll@update@state}{0}%
            \hll@update@fixclassoffset
        \fi
    \fi
    %
    \expandafter\hll@update@setstyle\expandafter{\the\c@hll@update@classoffset}{#1}%
}

\def\hll@update@setstyle#1#2{%
    \lstset{%
        classoffset={#1},
        keywordstyle={#2},
        texcsstyle={*#2},
    }%
}

```

7.6.6 Key ‘color’

```
\define@key{updatehighlight}{color}{%
    \hll@update@fixclassoffset
    \unless\ifhllStyleAnywhere
        \ifnum\value{hll@update@state}>1\relax
            \setcounter{hll@update@state}{0}%
            \hll@update@fixclassoffset
        \fi
    \fi
    %
    \expandafter\hll@update@setcolor\expandafter{\the\c@hll@update@classoffset}{#1}%
}

\def\hll@update@setcolor#1#2{%
    \lstset{%
        classoffset={#1},
        keywordstyle={\color{#2}},
        texcsstyle={*\color{#2}},
    }%
}

```

7.6.7 Key ‘clear’

```
\define@key{updatehighlight}{clear}[]{%
    \hll@update@fixclassoffset
    \expandafter\hll@update@clear\expandafter{\the\c@hll@update@classoffset}%
}

\def\hll@update@clear#1{%
    \lstset{
        classoffset={#1},
        texcs={},
        keywords={},
        classoffset=0
    }%
}
```