

# Package `highlightlatex` manual

Vincent Kuhlmann  
`vincent.kuhlmann@hotmail.com`

March 23, 2021

## Abstract

This package provides colored syntax highlighting for  $\text{\LaTeX}$  source code, without aid from outside  $\text{\LaTeX}$ . This is in response to the general trend that people often fall back to verbatim for displaying code. This package aims to make typesetting good looking  $\text{\LaTeX}$  source code accessible. 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  $\sqrt{2}$   $\leftarrow$  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

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Inline style . . . . .	3
1.2	Block style . . . . .	3
<b>2</b>	<b>Macro <code>\updatehighlight</code></b>	<b>3</b>
2.1	Adding a command to a highlighting rule . . . . .	3
2.2	Custom highlighting rules . . . . .	4
2.2.1	Example . . . . .	4
2.2.2	Specification . . . . .	4
<b>3</b>	<b>Global settings</b>	<b>5</b>
<b>4</b>	<b>Fragile breaking situations (like beamer frames)</b>	<b>5</b>
<b>5</b>	<b>Adding extra space</b>	<b>6</b>
<b>6</b>	<b>License &amp; Credits</b>	<b>6</b>
<b>7</b>	<b>Implementation</b>	<b>6</b>
7.1	Package options . . . . .	7
7.2	Initialization . . . . .	8
7.3	Options . . . . .	9
7.4	Block and inline markup . . . . .	11
7.5	Block saving . . . . .	11
7.6	Macro <code>\updatehighlight</code> . . . . .	12
7.6.1	Key ‘name’ . . . . .	12
7.6.2	Key ‘classoffset’ . . . . .	13
7.6.3	Key ‘add’ . . . . .	13
7.6.4	Key ‘remove’ . . . . .	14
7.6.5	Key ‘style’ . . . . .	14
7.6.6	Key ‘color’ . . . . .	15
7.6.7	Key ‘clear’ . . . . .	15

# 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=4]
  \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 the block could run out of the page, for example you need to save the block for use in a beamer document (see Section 4). In that case, the normal full-width of a slide is assumed, but you might want to use it in a slide with 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<sup>1</sup> 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.

<sup>1</sup> <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](https://en.wikibooks.org/wiki/LaTeX/Colors)<sup>2</sup>.

### 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.

---

<sup>2</sup> <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 <sup>3</sup> 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 `\hll|\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 backslash before `\maketitle` in `\hll|\maketitle|` will be read as ‘just text’ (a *letter* technically).

---

<sup>3</sup> <https://www.ctan.org/pkg/listings>

When content gets interpreted early, 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 special characters with a backslash. For example, `\hl1|\documentclass[]{}|` becomes `\hl1|\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. 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}`, where the argument is the name used during saving. 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 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, `highlightlatex` 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

1 \NeedsTeXFormat{LaTeX2e}[1994/06/01]

```

2 \ProvidesPackage{highlightlatex}[2021/03/17 highlightlatex v1.0.0 + Dev]
3
4 %% Repository: https://github.com/vkuhlmann/highlight-latex
5 %% Version: v1.0.0 + Dev
6
7 %% Copyright (c) 2021 Vincent Kuhlmann
8 %%
9 %% Permission is hereby granted, free of charge, to any person obtaining a copy of
10 %% this software and associated documentation files (the "Software"), to deal in
11 %% the Software without restriction, including without limitation the rights to
12 %% use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
13 %% the Software, and to permit persons to whom the Software is furnished to do so,
14 %% subject to the following conditions:
15 %%
16 %% The above copyright notice and this permission notice shall be included in all
17 %% copies or substantial portions of the Software.
18 %%
19 %% THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
20 %% IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS
21 %% FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
22 %% COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
23 %% IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
24 %% CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
25
26 \RequirePackage{listings}
27 \RequirePackage{xcolor}
28
29 \RequirePackage{xkeyval}
30 \RequirePackage{etoolbox}
31
32 %% Snippet source:
33 %% https://tex.stackexchange.com/questions/406015/defining-macro-gsetlength-as-global-setlength-reliable
34 \gdef\gsetlength#1#2{%
35     \begingroup
36         \setlength\skip@{#2}%
37         \global#1=\skip@
38     \endgroup
39 }
40
41
42 \def\@highlight@frame{none}
43 \newlength{\hll@offset@fixskip}
44 \setlength{\hll@offset@fixskip}{0pt}
45
46 \newif\ifhllStyleAnywhere
47 \newcommand\defaultgobble{0}
48 \newif\ifhllDebugFrame
49 \newlength\hllBeforeSkip
50 \setlength\hllBeforeSkip{2pt}
51 \newlength\hllAfterSkip
52 \setlength\hllAfterSkip{0pt}
53

```

## 7.1 Package options

```

54 \DeclareOptionX{frame}{
55     \setframe{#1}
56 }
57

```

```

58 \DeclareOptionX{noframe}{
59     \setframe{none}
60 }
61
62 \DeclareOptionX{debugframe}{
63     \hllDebugFrame true
64 }
65
66 \def\setframe#1{%
67     \def\frameval{#1}%
68     \ifx\relax#1\relax
69         \def\frameval{lines}%
70     \fi
71     \def\linesvalue{lines}%
72     %
73     \xdef\@highlight@frame{\frameval}%
74     %
75     \ifx\frameval\linesvalue
76         \gsetlength{\hll@offset@fixskip}{6pt}%
77     \else
78         \gsetlength{\hll@offset@fixskip}{0pt}%
79     \fi
80 }
81
82 \DeclareOptionX{styleanywhere}{
83     \hllStyleAnywhere true
84 }
85
86 \newcounter{hlltabsize}
87 \setcounter{hlltabsize}{4}
88
89 \DeclareOptionX{tabsize}{
90     \setcounter{hlltabsize}{#1}
91 }
92
93 \DeclareOptionX{gobble}{
94     \def\defaultgobble{#1}
95 }
96
97 \DeclareOptionX*{\PackageWarning{highlightlatex}{‘\CurrentOption’ ignored}}
98 \ExecuteOptionsX{frame=lines}
99
100 \ProcessOptionsX\relax
101

```

## 7.2 Initialization

```

102 \colorlet{curlyBrackets}{red!50!blue}
103 \colorlet{squareBrackets}{blue!50!white}
104 \colorlet{codeBackground}{gray!6!white}
105 \colorlet{comment}{green!40!black}
106
107 \colorlet{accDefault}{blue!90!black}
108 \definecolor{accStructure}{RGB}{0,149,255}
109
110 \colorlet{accentB}{green!60!black}
111
112 \catcode‘\{=11
113 \let\myopenbrace{

```



```

114 \catcode'\{=1
115
116 \catcode'\}=11
117 \let\myclosebrace=}
118 \catcode'\}=2
119
120 \lstdefinelanguage{ColoredLaTeX}
121 [LaTeX]{TeX}{
122     backgroundcolor=\color{codeBackground},
123     keywordstyle=\bfseries\color{accDefault},
124     basicstyle=\footnotesize\ttfamily,
125     commentstyle=\color{comment},
126     numbers=none,
127     tabsize=\thehl1tabsize,
128     mathescape=false,
129     alsoletter={$_\!$!|?$_$},
130 %
131     classoffset=0,
132     keywordstyle=\color{accDefault},
133     texcsstyle=*\color{accDefault},
134     moretexcs={subsection,thesubsection,thesection,theoremstyle},
135     deletetexcs={begin,end},
136 %
137     classoffset=1,
138     keywordstyle=\color{accStructure},
139     texcsstyle=*\color{accStructure},
140     moretexcs={begin,end},
141 %
142     classoffset=2,
143     escapeinside=~~,
144 %
145     classoffset=0,
146     %literate={textA}{textB}{3}%
147     %https://tex.stackexchange.com/questions/172945/coloring-in-listing
148     literate={\{\}\{\textcolor{curlyBrackets}{\myopenbrace}\}\{1}
149             {\}\{\textcolor{curlyBrackets}{\myclosebrace}\}\{1}
150             {\[\]\{\textcolor{squareBrackets}{[\]}\}\{1}
151             {\]\]\{\textcolor{squareBrackets}{[\]}\}\{1}%
152     %      {}{\textcolor{delimiterColor}{(}}\{1}
153     %      {}{\textcolor{delimiterColor}{)}}\{1}%
154     %texcsstyle=\color{blue}
155 }
156 \lstset{language=ColoredLaTeX}
157
158 \def\hll@assignlabel#1#2{%
159     \expandafter\gdef\csname hll@labeltoclassoffset@#1\endcsname{#2}%
160 }
161
162 \hll@assignlabel{default}{0}
163 \hll@assignlabel{structure}{1}
164

```

### 7.3 Options

```

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

```

```

170     }%
171 }
172
173 \newcommand{\hllDelimParen}[1][\color{green!40!black}]{%
174     \lstset{
175         classoffset=2,
176         moredelim=**[s][#1]{\{ }\{\}},
177         classoffset=0
178     }%
179 }
180
181 \define@key{hllglobal}{tabsize}{%
182
183 }
184
185 \define@key{hllglobal}{frame}{%
186
187 }
188
189 \define@key{hllglobal}{noframe}{%
190
191 }
192
193 \define@key{hllglobal}{mathdollarstyle}{%
194
195 }
196
197 \define@key{hllglobal}{mathdollarcolor}{%
198
199 }
200
201 \define@key{hllglobal}{mathparenstyle}{%
202
203 }
204 \define@key{hllglobal}{mathparencolor}{%
205
206 }
207
208 \define@key{hllglobal}{gobble}{%
209
210 }
211
212 \define@key{hllglobal}{alsoletter}{%
213
214 }
215
216 \define@key{hllglobal}{backgroundcolor}{%
217
218 }
219 \define@key{hllglobal}{bracketcolor}{%
220
221 }
222 \define@key{hllglobal}{bracecolor}{%
223
224 }
225
226
227
228

```

## 7.4 Block and inline markup

```
229 \newcommand{\hllPrepareBlock}{%
230   \lstset{framesep=0.1cm,framerule=0.6pt}%
231   \def\setHighlightFrame##1{%
232     \lstset{frame=##1}%
233   }%
234   \expandafter\setHighlightFrame\expandafter{\@highlight@frame}%
235   \lstset{belowskip=0pt,aboveskip=0pt,belowcaptionskip=0pt,gobble=\defaultgobble}%
236   \setlength\parskip{0pt}%
237   \setlength\parindent{0pt}%
238 }
239
240 \lstnewenvironment{highlightblock}[1] []
241 {%
242   \vspace{\hllBeforeSkip}%
243   \begingroup
244   \hllPrepareBlock\lstset{#1}%
245 }{%
246   \nointerlineskip
247   \endgroup
248   \vspace{\hllAfterSkip}%
249 }
250
251 \lstnewenvironment{hllblock}[1] []
252 {%
253   \vspace{\hllBeforeSkip}%
254   \begingroup
255   \hllPrepareBlock\lstset{#1}%
256 }{%
257   \endgroup
258   \vspace{\hllAfterSkip}%
259 }
260
261 \def\hll{\lstinline}
262
```

## 7.5 Block saving

```
263 \newenvironment{saveblock}[1]{%
264   \expandafter\ifx\csname @codebox@#1\endcsname\relax
265     \expandafter\newbox\csname @codebox@#1\endcsname\relax
266   \fi
267   %
268   %% The trick from latex/base/latex.ltx
269   \edef\@bracehack{%
270     \endgroup
271     \expandafter\setbox\csname @codebox@#1\endcsname\vbox{%
272       \begingroup\aftergroup}%
273     \def\noexpand\@currentvir{\@currentvir}%
274     \def\noexpand\@currentvline{\on@line}}%
275   \@bracehack
276   \begingroup\vspace{\hll@offset@fixskip}
277   \ignorespaces%
278 }{%
279   \endgroup
280 }
281
282 \newcommand\useblock[1]{%
283   \ifhllDebugFrame
```

```

284     \par
285     {%
286         \setlength{\fboxsep}{0pt}%
287         \fcolorbox{orange}{red}{\expandafter\usebox\csname @codebox@#1\endcsname}%
288     }%
289     \par
290 \else
291     \par\expandafter\usebox\csname @codebox@#1\endcsname\par
292 \fi
293 }
294
295 \newcommand\consumeblock[1]{%
296     \ifhllDebugFrame
297         \par
298         {%
299             \setlength{\fboxsep}{0pt}%
300             \setlength{\fboxrule}{1pt}%
301             \fcolorbox{orange}{red}{\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax}%
302         }
303         \par
304         \expandafter\let\csname @codebox@#1\endcsname\relax
305     \else
306         \par\leavevmode\expandafter\box\csname @codebox@#1\endcsname\relax\par
307         \expandafter\let\csname @codebox@#1\endcsname\relax
308     \fi
309 }
310

```

## 7.6 Macro \updatehighlight

```

311 \def\hll@update@label{}
312
313 \newcounter{hll@update@state}
314
315 \newcounter{hll@update@nextclassoffset}
316 \setcounter{hll@update@nextclassoffset}{5}
317
318 \newcounter{hll@update@classoffset}
319
320 \def\hll@update@fixclassoffset{%
321     \ifnum\value{hll@update@state}<1\relax
322         \setcounter{hll@update@classoffset}{\thehll@update@nextclassoffset}%
323         \stepcounter{hll@update@nextclassoffset}%
324         \setcounter{hll@update@state}{1}%
325     \fi
326 }
327
328 \newcommand{\updatehighlight}[1]{%
329     \setcounter{hll@update@state}{-1}%
330     \gdef\hll@update@label{}%
331     \setkeys{updatehighlight}{#1}%
332     %
333     \lstset{classoffset=0}%
334 }
335

```

### 7.6.1 Key ‘name’

```

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

```

```

338 }
339
340 \def\hll@update@setlabel#1{
341   \gdef\hll@update@label{#1}%
342   \expandafter\def\expandafter\@capturedlabel\expandafter{\csname hll@labeltoclassoffset@#1\endcsname}%
343   \expandafter\unless\expandafter\ifx\@capturedlabel\relax\relax
344     % Label exists
345     \edef\@val{\@capturedlabel}%
346     \setcounter{hll@update@classoffset}{\@val}%
347     \setcounter{hll@update@state}{1}%
348   \else
349     % Allocate new label
350     \edef\@val{\the\hll@update@nextclassoffset}%
351     \setcounter{hll@update@classoffset}{\@val}%
352     \stepcounter{hll@update@nextclassoffset}%
353     \setcounter{hll@update@state}{1}%
354     %
355     \expandafter\xdef\@capturedlabel{\@val}%
356   \fi
357 }
358

```

## 7.6.2 Key ‘classoffset’

```

359 \define@key{updatehighlight}{classoffset}{%
360   \setcounter{hll@update@classoffset}{#1}%
361   \setcounter{hll@update@state}{1}%
362 }

```

## 7.6.3 Key ‘add’

```

363 \define@key{updatehighlight}{add}{%
364   \hll@update@add{#1}%
365 }
366
367 \def\hll@update@add#1{%
368   \hll@update@fixclassoffset
369   \def\@inv##1{\forcsvlist{\hll@update@additem{##1}}{#1}}%
370   \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
371   %
372   \setcounter{hll@update@state}{2}%
373 }
374
375 \def\hll@update@additem#1#2{%
376   \if\noexpand#2\relax
377     \edef\keywordpure{\expandafter\@gobble\string#2}%
378     \def\callwithpure##1{\hll@update@addmacro{#1}{##1}}%
379     \expandafter\callwithpure\expandafter{\keywordpure}%
380   \else
381     \hll@update@addkeyword{#1}{#2}%
382   \fi
383 }
384
385 \def\hll@update@addmacro#1#2{%
386   \lstset{
387     classoffset=0,deletetexcs={#2},
388     classoffset=1,deletetexcs={#2},
389     classoffset=#1,moretexcs={#2},
390     classoffset=0
391   }%

```

```

392 }
393
394 \def\hll@update@addkeyword#1#2{%
395     \lstset{
396         classoffset=0,deletekeywords={#2},
397         classoffset=1,deletekeywords={#2},
398         classoffset={#1},otherkeywords={#2},morekeywords={#2},
399         classoffset=0
400     }%
401 }
402

```

#### 7.6.4 Key ‘remove’

```

403 \define@key{updatehighlight}{remove}{%
404     \hll@update@remove{#1}%
405 }
406
407 \def\hll@update@remove#1{%
408     \hll@update@fixclassoffset
409     \def\@inv##1{\forcsvlist{\hll@update@removeitem{##1}}{#1}}%
410     \expandafter\@inv\expandafter{\the\c@hll@update@classoffset}
411     %
412     \setcounter{hll@update@state}{2}%
413 }
414
415 \def\hll@update@removeitem#1#2{%
416     \if\noexpand#2\relax
417         \edef\keywordpure{\expandafter\@gobble\string#2}%
418         \def\callwithpure##1{\hll@update@removemacro{#1}{##1}}%
419         \expandafter\callwithpure\expandafter{\keywordpure}%
420     \else
421         \hll@update@removekeyword{#1}{#2}%
422     \fi
423 }
424
425 \def\hll@update@removemacro#1#2{%
426     \lstset{
427         classoffset={#1},
428         deletetexcs={#2},
429         classoffset=0
430     }%
431 }
432
433 \def\hll@update@removekeyword#1#2{%
434     \lstset{
435         classoffset={#1},
436         deletekeywords={#2},
437         classoffset=0
438     }%
439 }
440

```

#### 7.6.5 Key ‘style’

```

441 \define@key{updatehighlight}{style}{%
442     \hll@update@fixclassoffset
443     \unless\ifhllStyleAnywhere
444         \ifnum\value{hll@update@state}>1\relax
445             \setcounter{hll@update@state}{0}%

```

```

446         \hll@update@fixclassoffset
447     \fi
448 \fi
449 %
450 \expandafter\hll@update@setstyle\expandafter{\the\c@hll@update@classoffset}{#1}%
451 }
452
453 \def\hll@update@setstyle#1#2{%
454     \lstset{%
455         classoffset={#1},
456         keywordstyle={#2},
457         texcsstyle={*#2},
458     }%
459 }
460

```

### 7.6.6 Key ‘color’

```

461 \define@key{updatehighlight}{color}{%
462     \hll@update@fixclassoffset
463     \unless\ifhllStyleAnywhere
464         \ifnum\value{hll@update@state}>1\relax
465             \setcounter{hll@update@state}{0}%
466             \hll@update@fixclassoffset
467         \fi
468     \fi
469     %
470     \expandafter\hll@update@setcolor\expandafter{\the\c@hll@update@classoffset}{#1}%
471 }
472
473 \def\hll@update@setcolor#1#2{%
474     \lstset{%
475         classoffset={#1},
476         keywordstyle={\color{#2}},
477         texcsstyle={*\color{#2}},
478     }%
479 }
480

```

### 7.6.7 Key ‘clear’

```

481 \define@key{updatehighlight}{clear}[]{%
482     \hll@update@fixclassoffset
483     \expandafter\hll@update@clear\expandafter{\the\c@hll@update@classoffset}%
484 }
485
486 \def\hll@update@clear#1{%
487     \lstset{
488         classoffset={#1},
489         texcs={},
490         keywords={},
491         classoffset=0
492     }%
493 }

```