

Zpravodaj Československého sdružení uživatelů TeXu

Filip Blaho

Sadzbazdrojového kódu v LaTeXu pomocí balíka minted

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 35 (2025), No. 1-4, 44–48

Persistent URL: <http://dml.cz/dmlcz/153199>

Terms of use:

© Československé sdružení uživatelů TeXu, 2025

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:
The Czech Digital Mathematics Library <http://dml.cz>

Sadzba zdrojového kódu v L^AT_EXu pomocou balíka minted

FILIP BLAHO

Tento článok poskytuje základný prehľad L^AT_EXového balíka minted, ktorý slúži na zvýrazňovanie syntaxe zdrojového kódu v odborných dokumentoch. Balík využíva externú pythonovú knižnicu Pygments, čo umožňuje farebné formátovanie kódu v rôznych programovacích jazykoch s vysokou presnosťou a prispôbitelnosťou. Článok detailne popisuje základné použitie balíka, možnosti formátovania a metódy vkladania kódu. Záverečná časť analyzuje aktuálny vývoj balíka so zameraním na verziu 3 a jej bezpečnostné vylepšenia.

Kľúčové slová: L^AT_EX, minted, zvýrazňovanie syntaxe zdrojového kódu, Pygments

Minted je L^AT_EXový balík zameraný na zvýraznenie kódu programu, ktorý je vložený do dokumentu. Na zvýraznenie syntaxe používa Python, konkrétne knižnicu Pygments, ktorá vráti zvýraznený text [1, sekcia 1].

Vkladanie kódu

Najprv na začiatku dokumentu načítame balík príkazom `\usepackage{minted}`.

Na zvýraznenie syntaxe zdrojového kódu v dokumente sa používa prostredie minted. Základný tvar zápisu je nasledujúci:

```
\begin{minted}[\langle volby \rangle]{\langle jazyk \rangle}1
  \langle zdrojový kód \rangle
\end{minted}
```

Nasledujúci príklad zobrazuje zdrojový zápis a jeho zvýraznenú podobu prostredníctvom balíka minted:

Zdrojový zápis

```
\begin{minted}{python}
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
\end{minted}
```

Zvýraznený výstup

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

¹Pygments podporuje stovky programovacích a iných jazykov [1, sekcia 4.5]. (Pozn. red.)

Príkaz `\mint` slúži na zvýraznenie jedného riadka zdrojového kódu. Hoci sa zapisuje inline, vo výstupe preruší aktuálny odsek a zdrojový kód vysadí pod ním [1, sekcia 4.3]:

```
\mint{python}/import pandas as pd/
```

```
import pandas as pd
```

Príkaz `\mintinline` sa zapisuje rovnako ako príkaz `\mint`, ale na rozdiel od neho nepreruší aktuálny odsek a vloží zdrojový kód do výstupu ako súčasť odseku.

Príkaz `\inputminted` okrem *⟨volieb⟩* a *⟨jazyka⟩* berie aj *⟨názov súboru⟩*, ktorého syntax zobrazí a zvýrazní:

```
\inputminted{python}{cube_volume_and_content.py}
```

```
def cube_volume_and_content(a):  
    volume = a ** 3          # a * a * a  
    content = 6 * a ** 2     # 6 * a * a  
    return (volume, content)
```

Nastavenia

Okrem predvoleného vzhľadu umožňuje balík `minted` prispôbiť formátovanie zvýrazneného zdrojového kódu, čomu sa budeme venovať v tejto časti.

Font, veľkosť, farba, štýl písma, ...

Na podrobnejšie nastavenie formátu `minted` slúži atribút *⟨voľby⟩*, ktorý sa píše za príkaz, ale je dobrovoľný, a teda nie je ho tam potrebné písať, ak nám stačí základný formát, poprípade nami prednastavený formát. Balík `minted` obsahuje desiatky *⟨volieb⟩*, ktoré popisuje manuál [1, kapitola 7]. My sa budeme zaoberať základnými nastaveniami na zlepšenie estetiky kódu v dokumente.

Začneme pridaním zdrojového kódu do rámcov, a to nasledovne:

```
\begin{minted}[ bgcolor=lightgray, frame=single,  
rulecolor=\color{black}, style=tango]{python}  
def hello():  
    print("Hello, world!")  
\end{minted}
```

```
def hello():  
    print("Hello, world!")
```

Dostali sme zdrojový kód v šedom rámci s čiernym rámkom. Farba `lightgray` nie je preddefinovaná balíkom `xcolor`, a teda potrebujeme najskôr definovať špeciálnu farbu:

```
\usepackage{xcolor}
\definecolor{lightgray}{rgb}{0.95,0.95,0.95}
```

Na zjednodušenie formátovania, aby sme stále nemuseli opisovať všetky doposiaľ vytvorené nastavenia balíka `minted`, slúži príkaz `\setminted`:

```
\setminted{
  bgcolor=lightgray,
  frame=single,
  rulecolor=\color{gray},
  style=tango}
```

Ďalšia možnosť formátovania je nastavenie štýlu a veľkosti písma:

```
\begin{minted}[ fontsize=\large, fontfamily=helvetica, style=tango,
bgcolor=lightgray, frame=single, rulecolor=\color{black}]{python}
```

```
def hello():
    print("Hello, world!")
```

```
\end{minted}
```

```
def hello():
    print("Hello, world!")
```

Nakoniec jedno z najdôležitejších formátovaní zdrojového kódu, číslovanie riadkov, sa vykoná pomocou atribútu `linenos` a odsadenie od kraja strany pomocou atribútu `xleftmargin=`*odsadenie* [1, sekcia 7.2]:

```
\begin{minted}[linenos, xleftmargin=50px]{python}
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
\end{minted}
```

```
1      class Node:
2          def __init__(self, value):
3              self.value = value
4              self.next = None
```

Štýly minted

Okrem samostatného nastavenia jednotlivých parametrov, ako sú farba, štýl a podobne, je možnosť použitia jedného z štýlov minted, ako sú napríklad `default`, `friendly`, `monokai`, `xcode`, `colorful`, `pastie`, `tango`, `vim` a veľa ďalších. Na použitie je ale potrebná definícia pred samotným použitím, a to nasledovne:

```
\usemintedstyle{<názov štýlu>}, alebo \begin{minted}[style=<názov štýlu>]
```

kde <názov štýlu> nájdeme v dokumentácii [1, sekcia 4.4].

Štýl xcode

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

Štýl tango

```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

Aktuálny vývoj

Tretia verzia balíka prináša významné zmeny v oblasti bezpečnosti, inštalácie a rozšíriteľnosti. Hlavnou novinkou je zavedenie samostatného pythonového spustiteľného súboru `latexminted`, ktorý nahrádza pôvodný nástroj `pygmentize` a odstraňuje predchádzajúce zraniteľnosti spojené s mechanizmom shell escape [2], zjednodušuje používanie a umožňuje rozširovať funkcionality pomocou Pythonu [3].

V predchádzajúcich verziách bolo nutné využívať neobmedzený shell escape, ktorý umožňoval spúšťať ľubovoľné príkazy počas kompilácie, čo predstavovalo bezpečnostné riziko. Nový nástroj `latexminted` je plne kompatibilný s obmedzeným shell escape, teda povoľuje len dôveryhodné externé programy (napr. `bibtex`, `kpsewhich` atď.) [3, sekcia 2]. Spustiteľný súbor `latexminted` pristupuje k súborovému systému prostredníctvom novej Python knižnice `latexrestricted`. Tá používa nastavenia konfigurácie `LATEX`u na určenie, ktoré súbory môže čítať a zapisovať.

Bezpečné odovzdávanie dát medzi `LATEX`om a Pythonom využíva štruktúrované súbory (Python literal format), spracovávané pomocou balíka `latex2pydata`. Tento formát umožňuje efektívnu komunikáciu a znižuje potrebu práce s dočasnými súborami [3, sekcia 2.2].

Inštalácia tretej verzie balíka sa výrazne zjednodušila. Balík obsahuje `LATEX`ové makrá, samotný spustiteľný súbor `latexminted` a všetky potrebné pythonové knižnice (vrátane `Pygments`). Používateľ tak nemusí inštalovať Python ani `Pygments` samostatne ani upravovať systémovú premennú `PATH`. Inštalácia je teda rovnaká ako pri bežných `LATEX`ových balíkoch [3, sekcia 3].

Pred treťou verziou balíka, ak sa počas spúšťania `pygmentize` vyskytla chyba, zobrazili sa v `LATEX`ovom logu surové Python chyby. V novej verzii sa chybové hlásenia konvertujú z Pythonu do formátu zrozumiteľného pre `LATEX`, doplnia sa o číslo riadku a vypíšu sa ako štandardná `LATEX`ová chyba. Nový parameter

debug umožňuje uchovávať dočasné súbory a zapisovať dodatočné informácie do logu na podrobnejšie ladenie [3, sekcia 4].

Balík tiež umožňuje rozšíriteľnosť pomocou Pythonu. Nové možnosti ako `rangestartstring` a `rangestopstring` umožňujú vybrať úryvok kódu podľa delimitačných reťazcov textu, zatiaľčo `rangeregex` umožňuje vybrať úryvok pomocou regulárneho výrazu Pythonu. Možnosť `extrakeywords` a jej varianty (`extrakeywordstype`, `extrakeywordsconstant` atď.) umožňujú pridávať vlastné kľúčové slová bez nutnosti programovať vlastný lexer. Pre pokročilých používateľov je možné načítať aj vlastný pythonový lexer zo súboru `.py`, pričom jeho povolenie je riadené konfiguráciou `.latexminted_config` a overuje sa cez hash súboru na zachovanie bezpečnosti [3, sekcia 5].

Príkladom novej funkcie je voľba `literatecomment`, ktorá odstraňuje určitý znak (napr. %) z každého riadku zdrojového kódu, ak sa nachádza na začiatku všetkých riadkov. Táto voľba je užitočná pri práci s literárnym programovaním, ako je formát `.dtx` používaný pri písaní L^AT_EXových balíkov [3, sekcia 6].

Celkovo tretia verzia balíka `minted` predstavuje modernejší a bezpečnejší spôsob, ako prepájať L^AT_EX s Pythonom prostredníctvom štandardizovaného dátového formátu, pričom zachováva kompatibilitu so systémom obmedzeného shell escape. Nová architektúra otvára cestu pre budúci vývoj ďalších Pythonom rozšíriteľných funkcií v L^AT_EXových balíkoch [3, sekcia 7].

Odkazy

1. POORE, Geoffrey M. *The L^AT_EX Package minted: Highlighted source code in L^AT_EX* [online]. CTAN, 2025-05-14. Ver. 3.7.0 [cit. 2025-11-13]. Dostupné z: <https://tug.ctan.org/macros/latex/contrib/minted/minted.pdf>.
2. *The shellesc Package* [online]. 2023-07-08. [cit. 2025-11-20]. Dostupné z: <http://mirrors.ctan.org/macros/latex/required/tools/shellesc.pdf>.
3. POORE, Geoffrey M. Typesetting code with syntax highlighting: Introducing minted version 3. *TUGboat*. 2025, **46**(1), 54–59.

Summary: Typesetting Source Code in L^AT_EX Using the Minted Package

This article provides a comprehensive overview of the L^AT_EX package `minted`, which is used for syntax highlighting of the source code in scholarly documents. The package uses the Python library `Pygments`, enabling color formatting of code in various programming languages with high accuracy and customization options. The article details the basic usage of the package, formatting options, and methods for inserting code. The final section analyzes the current development of the package, focusing on version 3 and its security enhancements.

Keywords: L^AT_EX, `minted`, syntax highlighting, `Pygments`

Filip Blaho, 550233@mail.muni.cz