

České vysoké učení technické v Praze
Fakulta informačních technologií

tagsnip

Semestrální práce - Typografie a T_EX

Autor: Rostislav Brož
Rok: 2026

tagsnip

tagsnip je LaTeXový balíček pro vkládání označených úryvků kódu z lokálních souborů nebo vzdálených repozitářů dostupných přes webové odkazy. Extrahuje označené části kódu a sází je jednotným stylem, čímž podporuje reprodukovatelnou a udržitelnou dokumentaci.

Instalace

tagsnip pro parsování zdrojových souborů používá stejnojmenný Python balíček, který je publikovaný v softwarovém repozitáři PyPI [1]. Aby *tagsnip* fungoval, je potřeba backend nainstalovat následujícím příkazem.

```
1 pip install tagsnip
```

Použití

tagsnip definuje příkaz `\IncludeCode`, který se používá následovně:

```
1 \IncludeCode[options]{source}{tag}{language}{caption}
```

Nepovinný argument *options* umožňuje uživateli přepsat nastavení formátování kódu. V příkazu níže byly použity přepínače *firstnumber*, *fontsize* a *style*, které nastaví číslování řádků od čísla 1, velikost fontu na `\scriptsize` a styl zvýrazňování kódu na *monokai*. Přepínače musí existovat v balíčku Minted [2] a být oddělené čárkou. Je také důležité zmínit, že bez úpravy přepínače *firstnumber* *tagsnip* čísluje řádky tak, jak jsou číslovány ve zdrojovém souboru.

```
1 \IncludeCode[firstnumber=1, fontsize=\scriptsize, style=monokai]
  {source}{tag}{language}{caption}
```

Povinný argument *source* definuje zdroj, odkud se má úrvek kódu převzít. Argument může být ve formě cesty k lokálnímu souboru, nebo odkazu na webovou stránku. *tagsnip* dokáže tyto možnosti sám rozlišit. Povinný argument *tag* specifikuje unikátní klíčové slovo, které označuje jeden úrvek kódu. Tag musí být oddělený přesně jednou mezerou a následovat za frázemi *tagsnip-start* a *tagsnip-end*, které úrvek vymezují (viz Úryvek 1).

```
2 # tagsnip-start tag1
3 def main():
4     x = 1
5     y = 2
6     print(x + y)
7
8     return 0
9 # tagsnip-end tag1
```

Úryvek 1: Ukázka používání tagů.

Povinným argumentem *language* uživatel zvolí programovací jazyk, v jehož stylu má být kód v úryvku zvýrazněn. Posledním argumentem je popis, který může být prázdný. Nicméně složené závorky je potřeba psát i pro prázdný popis.

Lokální úrvek

Pro ukázkou toho, jak příkaz `\IncludeCode` funguje pro lokální soubory využijeme soubor *example.py* s Úryvkem 1. V něm je funkce *main()* obalená tagem *tag1*. Příkaz pro zobrazení kódu funkce v dokumentu by tedy vypadal takto:

```
1 \IncludeCode{example.py}{tag1}{Python}{Úryvek 2: ...}
```

Tento příkaz se přeloží na Úryvek 2:

```
3 def main():
4     x = 1
5     y = 2
6     print(x + y)
7
8     return 0
```

Úryvek 2: Úrvek vymezený tagem *tag1* z Úryvku 1.

Úryvek z webové stránky

Součástí domovského repozitáře *tagsnipu* na GitHubu je i soubor *example2.py*, ve kterém se nachází funkce pro extrakci úryvku. Část této funkce je ohraničená tagem *tag2*, tu zobrazíme následujícím příkazem:

```
1 \IncludeCode{raw.githubusercontent.com/brozrost
  /tagsnip/main/docs/example2.py}{tag2}{Python}{Úryvek 3: ...}
```

Výsledkem tohoto příkazu je Úryvek 3:

```
42     for i, line in enumerate(lines):
43         if marker_matches(line, start_marker):
44             if start_index is not None:
45                 raise TagsnipError(f"Multiple start tags found for
46                     '{tag}'")
47
48                 start_index = i
49
50     if start_index is None:
51         raise TagsnipError(f"Start tag not found for '{tag}'")
```

Úryvek 3: Úryvek kódu z webové stránky.

Architektura

tagsnip se skládá ze dvou hlavních částí, LaTeXového frontend balíčku a externí backend utility napsané v Pythonu. Frontend definuje příkaz `\IncludeCode` a zajišťuje sazbu výsledných úryvků pomocí balíčku *Minted*. Backend je zodpovědný za přístup ke zdrojovým souborům, vyhledávání označených úseků a jejich extrakci do dočasných souborů. Při překladu dokumentu frontend pomocí shell escape spustí backend utilitu, předá jí cestu ke zdrojovému souboru a název tagu, a následně vloží vygenerovaný dočasný soubor do dokumentu.

Backend

Python balíček *tagsnip* je jednoduché command line rozhraní, které přijímá zdrojový soubor, název tagu a cestu k výstupnímu souboru. Rozhraní načte zdrojový soubor, vyhledá odpovídající dvojici značek *tagsnip-start* a *tagsnip-end* a následně extrahuje text mezi nimi.

Balíček také provádí kontrolu chybových stavů. Chybovým stavem je myšlen například neexistující zdrojový soubor, chybějící tag nebo vícenásobný výskyt stejného tagu ve zdrojovém souboru. V případě detekce chyby backend vyvolá výjimku, která přeruší překlad dokumentu a vypíše odpovídající chybovou zprávu. Pro práci s lokálními soubory balíček používá modul *pathlib*, pro vyhledávání tagů modul *re* a pro načítání vzdálených souborů přes HTTP knihovnu *requests*.

Omezení

- *tagsnip* vyžaduje překladač LuaLaTeX.
- Při kompilaci dokumentu musí být povolen shell escape pomocí přepínače `-shell-escape`.
- Python backend musí být nainstalovaný a dostupný v systémové proměnné PATH pod příkazem *tagsnip*.
- Balíček závisí na balíčku *Minted*.
- Vzdálené zdrojové soubory musí být dostupné přes HTTP(S) a ve formátu čitelného textového souboru.

Reference

- [1] Rostislav Brož. Domovská stránka balíčku *tagsnip*. <https://pypi.org/project/tagsnip/>, 2026.
- [2] Geoffrey M. Poore. The minted package: Highlighted source code in LaTeX. <https://tug.ctan.org/macros/latex/contrib/minted/minted.pdf>, 2026.