

# Program na konvertování markdownu do html

---

Program, který pomocí regulárních výrazů konvertuje markdown elementy do html

Program potřebuje:

- python 3.x
- knihovny sys, re

Program spustíte pomocí následujícího příkazu na příkazové řádce: `python3 main.py (daný markdown soubor)` A vytvoří vám konverovaný soubor se stejným názvem a příponou ".html"

## Uživatelská dokumentace

Program podporuje základní markdown syntaxi, která se řídí podle [Markdown příručkové stránky](#)

### Nadpisy

Podobně jako v html, program podporuje 6 úrovní nadpisů. Pro vytvoření nadpisu, začněte řádek jedním až šesti `\#`. Celý řádek se počítá jako nadpis.

Markdown	Html
<code># Nadpis úrovně 1</code>	<code>&lt;h1&gt; Nadpis úrovně 1 &lt;/h1&gt;</code>
<code>## Nadpis úrovně 2</code>	<code>&lt;h2&gt; Nadpis úrovně 2 &lt;/h2&gt;</code>
<code>##### Nadpis úrovně 6</code>	<code>&lt;h6&gt; Nadpis úrovně 6 &lt;/h6&gt;</code>

### Paragrafy

Paragraf začíná a končí prázdným řádkem. Pokud chcete vytvořit paragraf, je potřeba prázdný řádek vytvořit jak před paragrafem, tak i za ním. Paragraf může mít více řádků.

Markdown	Html
jeden paragraf	<code>&lt;p&gt;jeden paragraf&lt;/p&gt;</code>
druhý paragraf	<code>&lt;p&gt;druhý paragraf&lt;/p&gt;</code>

### Únikové znaky

Pro vložení speciálních markdown znaků do textu, je nutné před nimi použít zpětné lomítko `\`. Jsou to znaky:

- `\*` - hvězda
- `\`` - zpětné uvozovky
- `\-` - pomlčka
- `\>` - větší než

### Blockquote

Pro citování sekce textu, použijte `>` na začátku řádku. Pokud chcete, aby citace měla více řádků, tak každý další řádek začněte dalším `>`, dokud má citace pokračovat. Blockquote také podporuje markdown syntaxi uvnitř daných bloků.

Markdown	Html
	<code>&lt;blockquote&gt;</code>
<code>&gt;citace</code>	<code>citace</code>
<code>&gt;přes několik řádků.</code>	<code>přes několik řádků.</code>
	<code>&lt;/blockquote&gt;</code>

## Seznamy

Program podporuje seznamy jak neočíslované pomocí pomlčky `-`, tak i očíslované. Očíslované seznamy mohou mít libovolné čísla, ovšem zobrazí se běžně od jedné. Seznamy do sebe mohou být vnořené.

Markdown	Html
	<code>&lt;ul&gt;</code>
<code>- neočíslovaný</code>	<code>&lt;li&gt;neočíslovaný&lt;/li&gt;</code>
<code>- seznam</code>	<code>&lt;li&gt;seznam&lt;/li&gt;</code>
	<code>&lt;ul&gt;</code>
	<code>&lt;ol&gt;</code>
<code>1. očíslovaný</code>	<code>&lt;li&gt;očíslovaný&lt;/li&gt;</code>
<code>2. seznam</code>	<code>&lt;li&gt;seznam&lt;/li&gt;</code>
	<code>&lt;ol&gt;</code>

## Kód

Pro označení části textu jako kódu, lze text ohraničit zpětnými uvozovkami ```. Všechny speciální markdown znaky jsou automaticky konvertovány na html kódové znaky a nemusí se používat zpětné lomítko.

Markdown	Html
<code>` zde je kód `</code>	<code>&lt;code&gt; zde je kód &lt;/code&gt;</code>

## Odkazy

Pro přidání odkazu do textu, napište název odkazu do hranatých úvozovek název odkazu `[Jméno]` a hned za to napište odkaz do kulatých úvozovek odkaz `(www.odkaz.cz)`.

Markdown	Html
<code>[Název](www.odkaz.cz)</code>	<code>&lt;a href="www.odkaz.cz"&gt;Název&lt;/a&gt;</code>

## Obrázky

Pro přidání obrázku do textu, napište vykřičník `!` následovaný hranatými úvozovkami s alt textem `[alt text]` a hned za ním kulaté závorky s cestou k obrázku a nepovinně i název obrázku v dvojitéch úvozovkách `(cesta`

"Název").

Markdown	Html
![alt text](obrázek.jpg "Text k obrázku")	

## Horizontální čára

Pro přidání horizontální čáry, napište do textu tři pomlčky nebo tři hvězdy za sebou: --- nebo \*\*\*

Markdown	Html
---	<hr>
***	<hr>

## Kurzíva

Pro označení části textu kurzívou, ohraničte daný text jednou hvězdou \*kurzivní text\*

Markdown	Html
*Kurzivní text*	<em>Kurzivní text</em>

## Tučné písmo

Pro označení části textu, ohraničte daný text dvěma hvězdami \*\*tučný text\*\*

Markdown	Html
**tučný text**	<b>tučný text</b>

# Programátorská dokumentace

Program používá regulární výrazy, aby našel v textu určité syntaktické znaky popisované výše, které postupně převede. Samotné znaky nahradí příslušnými html ekvivalenty a zbytek textu zachová.

Pro regulární výrazy je použita knihovna re, která je do pythonu přímo zabudovaná.

Program se dělí na třídy, která každá konvertuje jeden syntaktický element. Každá třída dědí z abstraktní třídy `Converter()`, která definuje metodu `convert()`, která jako argument bere daný text, který chceme konvertovat a pomocí funkce `re.sub()` konvertuje text.

Všechny konkrétní třídy musí nadefinovat proměnnou `regex`, do které se přiřadí funkce `re.compile()`, kde určíme vzorec, který bychom měli v textu najít.

Funkce `replace()` pak vrátí string, kterým chceme nahradit markdownový ekvivalent.

Třída, která vyžadovala složitější úpravy je `NestedListConverter()`, která měnila funkci `convert()`. Funkce projíždí text řádek po řádce a hledá regulární výraz seznamu. Funkce používá zásobník na uložení úrovní vnoření. Každá úroveň seznamu si uloží tag, zda-li budou její jednotlivé odrážky očíslované nebo neočíslované a pokud se úroveň sníží, tak ze zásobníku vezme tag a ukončí danou úroveň seznamu. Každý konvertovaný řádek se zapisuje do pythonového seznamu, který se pak sjednotí na jeden řetězec, který `convert()` vrátí a regexovou funkci `re.sub()` již nepoužívá.

Jednotlivé třídy, které chceme využít ke konvertování textu se zapisují na konec souboru do seznamu s názvem `converters`. Pomocí for loopu poté projde všechny třídy seznamu a jedna po druhé spustí funkci `convert()` na daný soubor. Je potřeba dávat pozor na pořadí tříd, aby některá syntaxe nebyla špatně překonvertována, pokud se některé části kryjí.