

Koncept skriptu

Skript se skládá ze vstupního .py souboru(soubor syn.py) a zbylé části aplikace, která je objektová.

Ve vstupním souboru je pouze vytvoření základních instancí aplikace(`Config` a `App`) a zachycování výjimek, které aplikace při své činnosti vyvolává. Nikde v aplikaci tedy není použito `ukončen`, ale vždy se vyvolá výjimka, která je zpracována ve vstupním souboru a až zde je skript ukončen s příslušnou návratovou hodnotou a chybovou zprávou.

Aplikace je rozčleněna do 5 balíků:

- `arguments` - konfigurace a zpracování argumentů
- `format` - třídy reprezentující HTML tagy
- `formatting` - samotné formátování vstupního textu
- `io` - vstup a výstup
- `regex` - parsování vstupního souboru s formátováním, převod regulárního výrazu do formátu pro python

Zpracování argumentů

Zpracování argumentů se odehrává ve třídě `Config`. Nebyla využita žádná knihovna na zpracování argumentů, ale vlastní parser využívající regulární výrazy na parsování vstupu. Každý argument má i zkrácenou variantu, např. `--input a -i`.

Veškerá konfigurace aplikace je uložena v instanci třídy `Config`.

Formátovací soubor

Každý řádek formátovacího souboru je složen z regulárního výrazu, odděleného jedním nebo více tabulátory od formátovacích příkazů. Formátovací příkazy(např. `bold`, `size:5`) jsou od sebe oddělené čárkou a libovolným počtem mezer nebo tabulátorů.

Zpracování souboru probíhá po řádcích s použitím regulárních výrazů pro parsování. Každý řádek je rozdělen na 2 části podle první skupiny tabulátorů. První částí je regulární výraz, který je převeden na regulární výraz kompatibilní s regulárními výrazy pythonu. Druhá část představuje seznam formátovacích příkazů, které jsou následně rozděleny podle znaku čárka a převedeny na instance tříd odvozených od třídy `HtmlTag`. Tento převod zajišťuje třída `TagFactory`, která podle formátovacího řetězce vytvoří a vrátí instanci příslušné třídy. Každý objekt třídy `HtmlTag` uchovává řetězec, reprezentující uzavírající a otevírající tag.

Načítání vstupu a vypisování výstupu

Způsob načítání vstupu ovlivňuje argument `--input`, který určuje cestu k souboru, ze kterého bude načten vstupní soubor, který má být naformátován. Pokud tento argument chybí, je vstupní soubor načítán z `stdin`.

Výpis na výstup je řízen argumentem `--output(-o)` a v případě jeho absence je výpis směřován na `stdout`.

Pokud je aktivována možnost `--br`, je před výpisem na každý řádek přidána značka `
`.

Formátování

Formátování má na starost třída `Formatter`. Pro formátování se vstupní řetězec prochází znak po znaku a tyto znaky se s polu s formátovacími značkami kopírují na výstup. Pro každý znak se kontroluje, zda-li na jeho pozici není nutné nějaký tag uzavřít. Tyto tagy se uzavírají v opačném pořadí, než se otevíraly. Poté se hledají regulární výrazy, které odpovídají části vstupu od aktuálního znaku do konce. Pro každý takový regulární výraz jsou do výstupu vloženy otevírací formátovací značky a zároveň je uložen index, na kterém mají být tyto značky uzavřeny.