

Dokumentace SYN

Zadání:

Zadáním bylo vytvořit skript, který vloží na určitá místa HTML tagy, čímž se pro jazyk HTML zvýrazní syntaxe. Vkládání typu tagů a jejich umístění je dáno formátovacím souborem. Nebylo zadáno žádné omezení.

Postup:

Celý skript je rozdělen do několika logických částí. Nejprve je zapotřebí zjistit, s jakými parametry uživatel skript spouští. Dále je nezbytné zpracovat soubory, které má skript používat. Poté se prochází vstupní soubor, přidávají se HTML tagy a dohromady utváří výstupní soubor. Nakonec je zapotřebí, popřípadě, přidat na konec řádků znak „`
`“ a výsledek zapsat.

Zpracování parametrů:

Pro zpracování parametrů jsem vytvořil třídu *args*, obsahující metody *nactiarg* a *help*. Metoda *nactiarg* prochází všechny argumenty zadané skriptu, kontroluje jejich správnost, počet a nastavuje proměnné objektu. V případě chyby ukončí skript s daným návratovým kódem. Metoda *help* vypíše nápovědu.

Zpracování souborů:

Pro všechny soubory, se kterými skript pracuje, je definována třída *soubor*. Třída obsahuje mnoho metod pro otevření, uzavření, načtení znaků a jejich zapsání. Pokud je soubor určený pro čtení, třída jej otevře a načte všechny znaky do řetězce, jinak jej pouze otevře.

Zpracování formátovacího řetězce:

Dalším problémem bylo ověřit správnost řetězce uloženého ve formátovacím souboru a uložit zadaná pravidla pro vkládání tagů. K tomuto jsem využil funkci *partition* nacházející se v knihovně *re*, která rozdělí řetězec na dvě poloviny oddělené tabulátorem. Seznam formátovacích příkazů je rozdělen do pole funkcí *split* ze stejné knihovny.

Regulární výrazy:

Zadání umožňuje zadat formátovacímu souboru regulární výrazy, které jsou ovšem různé od regulárních výrazů známých programovacímu jazyku Python. Proto je nutné zkontrolovat každý regulární výraz a popřípadě jej převést na reg. výraz, se kterým Python umí pracovat.

Hledání regulárních výrazů ve vstupu:

Dále je potřeba najít pozice, na kterých budou umístěny HTML tagy. Nejvhodnější funkce mi přišla *finditer*, opět z knihovny *re*. Funkce vyhledá indexy začátku a konce výrazu v procházeném řetězci, které se uloží do pole.

Generování výstupu:

Prochází se vstupní řetězec a pokud se index znaku rovná indexu uloženému v poli se začátky a konci výrazu, pak se přidá určitý tag do výstupního řetězce.

Rozšíření:

Rozhodl jsem se implementovat NQS rozšíření z důvodů jeho jednoduchosti. Ve skriptu bez rozšíření by byla tato funkčnost brána za chybu, jejíž výskyt jsem již měl při implementaci rozšíření ošetřen. Proto mi stačilo pro správnost pozměnit pouze pár řádků kódu.

Závěr:

Jelikož jsem v minulosti již s jazykem Python3 pracoval, tak bych zařadil tento projekt mezi jednodušší. Ale i přesto mě tento projekt mnohé naučil.