Dokumentace úlohy SYN: Zvýraznění syntaxe v Pythonu do IPP 2011/2012
Jméno a příjmení: Karel Hala
Login: xhalak00

Úvod
Za úkol jsem měl zvýraznit syntaxi dokumentu pomocí HTML tagů.

Práce s argumenty
Argumenty jsem zpracoval obdobně, jako v perlu. Postupně jsem prošel jednotlivé argumenty a ty jsem zkontroloval. Měl jsem na výběr get:opts a také argparse. Nicméně se mi nelíbil ani u jednoho možnost psaní neúplných argumentů. Tak jsem postupně prošel v cyklu argumenty, uložil co se dalo a poté zkontroloval počet. Pokud byli zadány argumenty, které by tam neměli co dělat končí script s chybou 1. Vzhledem k tomu že zadaný program nepracuje s příliš velkým počtem argumentů, jeden cyklus a par regulárních výrazů není na škodu. Navíc jsem tímto docílil 100% úspěšnosti a přesného chování, jak jsem chtěl.

Hlavní funkčnost
Hlavní funkčnost celého programu záleží na hlavním poli. Jsou v něm uloženy postupně celé regulární výrazy, značky ukládané, indexy, na které ukládat značky a také velikost jednotlivých značek. Struktura pole je následující:
[['regulární_výraz',
['řetězec_počátečních_značek','řetězec_koncových_značek'],'velikost_počátečních_značek','velikost_koncových_značek',['množina_indexů_pro_počáteční_značky'],['množina_indexů_pro_koncové_značky']].....]

Načítání dat
Nejdříve načtu formátovací soubor, poté načtu vstupní soubor. Formátovací soubor rozdělím před tabulátorem a po tabulátoru. Před prvním tabulátorem je regulární výraz a za druhým jsou formátovací značky. Regulární výraz zpracuji ve funkci my_regulars(string). Pošlu ji řetězec se zadaným regulárním výrazem. Nejdříve je trošku poupravím na výskyt !něco a poté postupně načítám jednotlivý znak za druhým. Pokud narazím na % zvýším čítač. Pokud na cokoliv jiného podívám se kolik jsem načetl %. Pokud jsem načetl lichý počet. Pravděpodobně se v regulárním výrazu nachází %speciální_znak. Zkontroluji to a pokud narazím na nějakou blbost. Script končí s chybou 4.
Pokud načtu sud počet % uložím polovinu % a zkontroluji, zda je regulární výraz stále validní. Pokud ne. Končím s chybou 4.
Značky postupně procházím podobným způsobem, ale s tím rozdílem, že využívám regulárního výrazu re.match, kterým kontroluji zároveň danou značku.
Tyhle data uložím do hlavního pole a pokračuji k hledání jednotlivých umístnění, kam budu ukládat jednotlivé značky.

Tisknutí dat
Po načtení jednotlivých regulárních výrazů a značek, které budu ukládat. Projdu cel soubor pomocí funkce re.finditer. Uložím k jednotlivým značkám na jaký index v textu jej budu ukládat.
Poté projdu celé pole. Vezmu postupně jeden index, kam ukládám danou značku, ať už počáteční, nebo koncovou a porovnám ji s dalšími značkami. Takže projdu několikrát dané pole a pokud narazím na shodu. Měl bych ukládat na stejné místo, jako již budu ukládat další značku, nebo nějaký index je dál v textu, musím tento index posunout o velikost značky. Je mi jasné že je tohle celkem neefektivní přístup a je celkem pomalý. Napadal mne přístup vkládání značek od konce, kde bych nemusel postupně posouvat jednotlivé značky, nicméně tohle již bylo celkem pozdě. Script je pomalejší, ale to snad nebude vadit. Alespoň jsem si procvičil práci s cykli a uvědomil si jak je důležité se nad danou prací rozmyslet a jak funguje vkládání textu do textu a problémy s tímto spojené.
Takže jak jsem již napsal, postupně procházím celé pole vybírám index dané značky a pokud se nachází nějaká značka, na stejné pozici, nebo za tímto indexem posunu ji. Projdu několikrát dané pole a poté ho vrátím k tisknutí.
Otevřu soubor pro čtení a vytisknu jej.

Problémy a ojedinělé případy
Při práci na tomto projektu jsem se setkal převážně s problémy zalamování řádků. Nevím z jakého důvodu, ale Python načítal o řádek navíc a proto nefungoval řádně diff. Nebo diff očekával konec řádku a nebyl tam. Nicméně převod regulárních výrazů a tudíž kontrolu formátovacího souboru bych měl mít perfektní, stejně tak jako vkládání jednotlivých značek na jejich pozice. Použil jsem sice pomalejší variantu, nicméně by měla být funkční. Chtěl bych vyzdvihnout kontrolu formátovacího souboru. A kontrolu jednotlivých regulárních výrazů. Původně jsem chtěl převádět regulární výrazy pouze přes re.sub, nicméně mne poté napadla možnost s mnohonásobným opakováním % a tuhle možnost jsem, dle mého úsudky vyřešil celkem spolehlivě.