Dokumentace úlohy SYN: Zvýraznění syntaxe v Python 3 do IPP 2016/2017
Jméno a příjmení: Maroš Cocuľa
Login: xcocul00

Úvod

Tato dokumentace přibližuje řešení druhého projektu do předmětu IPP. Jedná se o jednoduchý skript v jazyce Python ve verzi 3.6. Hlavním účelem tohoto projektu je automatické zvýrazňování syntaxe části textu určeného regulérním výrazem na základě formátovací tabulky.

Dokumentace popisuje způsob řešení. Zpracování jednotlivých parametrů či formátovací tabulky. Způsob zpracování regulérních výrazů a aplikaci zadaného formátování na text.

Zpracování parametrů

K zpracování parametrů jsem využil postupu se dvěma částmi. V té první se parametry kontrolují pomoci regulérních výrazů. Pro každý správně zadaný parametr existuje jeden regulérní výraz. Kontroluje se shoda pomocí funkce `re.match()`. Tato část kontroluje špatně zadaný parametr, či duplicitu parametrů. Taktéž se zde vyhodnotí neznámý parametr. Následně se zkontroluje správnost parametru `--help`. Tento parametr se nemůže kombinovat s jinýma. V druhé části kontroly se používá funkce `parse_args()` z modulu `argparse`. Výstupem kontroly je struktura s jednotlivými parametry.

Načtení vstupního souboru a formátovací tabulky

Vstupní data se nacházejí v souboru zadaným v parametru `--input`. Data potřebné k formátování se nacházejí v parametru `--format`. Jestli není zadán parametr `--input`, očekává se vložení vstupních dát ze standardního vstupu `stdin`. Při neexistenci formátovacího souboru se výstup rovná vstupním datem.

Zpracování regulérních výrazů

Po načítání potřebných formátovacích dat se tyto data rozdělí podle řádků. Každý řádek obsahuje regulární výraz a značky, které je potřeba doplnit do textu na místa, které jsou ve shodě s daným regulérním výrazem. Výrazy jsou od značek oddělené libovolným počtem tabulátorů. Podle toho se tak části rozdělí do dvou polí.

První pole obsahuje regulérní výrazy. Každá položka z pole se zkontroluje, jestli je správně syntaktický zadána ve funkci `check_regex()`. Kontrola se provádí po jednotlivých znacích. Zde se regulární výrazy upraví do tvaru, který se dá zpracovat pomocí funkce `re ()`. Každý znak, který nesouhlasí, se změní na sekvenci znaků nebo jiný znak. Zde je potřebné pamatovat si případnou negaci výrazu.

Druhé pole obsahuje značky, které se přidají do textu. Data se opět rozdělí podle tabulátoru. Na rozdíl od prvního se však vezme část za tabulátory.

Aplikování formátování

K aplikování značek se používá seznam velikosti vstupního textu. Na začátku jsou všechny položky prázdné. Díky stejné velikosti se pozice textu rovná pozici v seznamu. Tím pádem je práce s textem jednoduchá a stačí jenom přidat tagovací značky.

To se děje nasledovně. Nejprve se vytvoří řetězce s tagovacími značkami. Následně se prochází iterativně text, který se porovnává s regulérním výrazem. Při shodě se na začátek shody vloží začáteční tag. Na pozici konce shody se přidá tag koncový.

Výpis výstupních dat

Nakonec se vyhodnotí parametr `--br`. Data se upraví podle potřeby tohoto parametru. Následně se výstupní data zapíšou do souboru z parametru `--output`. Když tento parametr není použit, výstup se vypíše na standardní výstup `stdout`.

Závěr

Při pracování na projektu jsem využíval školní server `merlin`. Tímto jsem si ulehčil práci a nemusel jsem hledat způsob, jak nainstalovat potřebné moduly jazyka Python na můj počítač. K práci jsem využíval i dokumentaci k Python jazyku dostupnou z internetových stránek.

Pro testování projektu jsem využil ukázkové testy ze stránek předmětu. Testy jsem si upravil podle mého gusta a doplnil o další testy. Dále jsem použil testy zveřejněné kolegy, které jsem trochu poměnil.