

## Struktura skriptu

Skript se skládá jen z jednoho hlavního souboru `csv.py`, který již neimportuje žádné další lokální soubory, ale jen vestavěné moduly samotného interpretu. Mezi importované a nejdůležitější moduly patří modul `argparse` pro parsování vstupních argumentů, třída `ElementTree` z balíku `xml.etree` pro tvorbu xml objektu, modul `csv` pro parsování csv souboru, třída `minidom` z balíku `xml.dom` a třída `HTMLParser` pro převedení zakódovaných entit zpět do znaků. Skript je napsán čistě procedurálním stylem s minimem pomocných funkcí, nebyly vytvářeny ani upravovány jakékoliv třídy ani rozhraní. Program samotný spíše využívá funkcí ostatních modulů, které spojuje, místo aby vytvářel jedinečný postup k řešení zadaného úkolu. Žádné z nabízených rozšíření nebyla implementována.

## Zpracování argumentů

Ke zpracování vstupních argumentů využívá skript modul `argparse`, který je standardně zabudován v interpretu pythonu od verze 3.2. Jednotlivé argumenty jsou postupně přidávány pomocí funkce `add_argument()`, která vyhledává zadaný řetězec v poli vstupních znaků `sys.argv`. Argumentů které se mohou předat skriptu je celkem 14 a každý je něčím specifický oproti ostatním. Všechny kontroly argumentů však `argparse` samotný nezvládá, a proto se některé kontroly prováděly po zparsování funkcí `parse_args()`. Mezi tyto kontroly patří kontrola, že volba `--help`, byla zadaná jako jediná bez dalších voleb nebo argumentů, kontrola že zadaný vstupní soubor existuje a kontrola, že u skupin parametrů byl zadán i povinný argument. Všechny argumenty a jejich hodnoty se pak ukládají do pole `args`, jehož hodnoty se využívají při procházení zparsovaným csv souborem a při zápisu xml souboru.

## Parsování CSV souboru

Parsování CSV souboru probíhá pomocí modulu `csv`, který je standardně zabudován v interpretu pythonu od verze 2.3. Pro začátek parsování musíme otevřít vstupní soubor pomocí funkce `open` a předat tento ukazatel na soubor funkci `csv.reader`, která sama soubor zparsuje a vrátí dvourozměrné pole řádků a buněk s hodnotami z csv souboru. Funkce sama bere jako pomocný parametr znak, který má použít jako oddělovač jednotlivých sloupců v csv souboru. Tato hodnota je ve výchozím stavu nastavena na znak čárky (`,`). Díky modulu `csv` probíhá parsování csv souboru na jediném řádku programu. Pokud bychom modul neměli museli bychom problém řešit složitěji pomocí lexikálního a syntaktického automatu.

## Průchod dvourozměrným polem

Pole získané z parsování csv souboru poté procházíme ve dvou zanořených cyklech a vytváříme xml objekt pomocí modulu `xml.dom.minidom`. První vnější cyklus prochází řádky csv souboru a druhý vnitřní cyklus prochází sloupce daného řádku. Ve vnějším cyklu zvlášť procházíme hlavičku (první řádek csv souboru) a zvlášť ostatní řádky, pokud byl zadán parametr `-h`, abychom mohli specifikovat názvy jednotlivých sloupcových element xml souboru. Během procházení dvourozměrného pole si při budování xml objektu vystačíme s funkcemi `ElementTree.SubElement()`, která vytváří podobjekty hlavního kořenového element, který byl vytvořen ještě před cyklem funkcí `ElementTree.Element()` a `Element.set()`, která nastavuje atribut řádkových elementů, pokud je zadán argument `-i`. Při průchodu se ve vnějším cyklu kontroluje zda je počet sloupců správný vzhledem k prvnímu řádku csv souboru. Pokud není a volba `-e` nebyla zadána, skript se ukončí s návratovou chybou 32. Pokud argument `-e` je zadán sloupce se buďto doplní nebo odstraní aby vyhovovaly počtu sloupců na prvním řádku. K odstranění sloupců dochází jen tehdy pokud není zadána volba `--all-columns`. Hodnoty sloupců se vyplňují hodnotou argumentu `--missing-field`, který je ve výchozím stavu nastaven na prázdný znak. Při tvorbě jednotlivých element xml souboru se jejich názvy odvozují od argumentů `-h`, `-c`, `-l` a hodnot v hlavičce pokud byl zadán argument `-h`. Tyto názvy se před každým použitím zkontrolují tak aby vyhovovaly standardům xml. To jest, že xml element může začínat jen písmenem nebo podtržítkem a dále může obsahovat jen písmena, číslice, podtržítka, tečku a pomlčku. Název také nesmí začínat zkratkou xml. Tyto požadavky se kontrolují regulárním výrazem, který je vytvořen v modulu `re`.

## Tisk XML souboru

Jakmile projdeme dvourozměrné pole můžeme nově vytvořený xml objekt vytisknout pomocí funkce `toprettyxml()`, která výsledek vytiskne i s odsazením a základním naformátováním, pro lepší přehlednost při testování a odhalování chyb ve skriptu. Tato funkce se nachází v modulu `dom` proto musíme xml objekt převést do tohoto prostředí pomocí funkce `ElementTree.tostring()`. Pro překódování speciálních znaků v xml využíváme

funkci `HTMLParser.unescape()`, která automaticky převede všechny nechtěné znaky do HTML (XML) entit. Poté jen stačí otevřít výstupní soubor a zapsat výsledný řetězec.