

C2184 Úvod do programování v Pythonu

Povinné domácí úkoly

Úkoly v této sadě řešte do připravených souborů `sort.py`, `recursive_find.py`, `hydrolase_cofactors.py`, `select_columns.py`, `convert_contacts.py`, které pak odevzdáte do odevzdáárny. Doctesty v těchto souborech považujte za součást zadání.

V každé úloze je naším úkolem napsat program, který budeme moct volat z příkazového řádku. Základní struktura je už připravena (blok `if __name__=='__main__': main()`). Naším úkolem je doplnit samotnou funkci `main` (nahradit `raise NotImplementedError` za řešení). Samozřejmě lze definovat další pomocné funkce a ty pak volat v `main`. Neměňte ale blok `if __name__=='__main__'` ani hlavičku funkce `main`.

V tomto zadání je vždy uvedený příklad volání programu - řádek začínající `$` popisuje, jak budeme program volat z příkazového řádku; další řádky jsou očekávaný výstup, který se má vypsát na terminál. Tento příklad je pouze ilustrační - více příkladů je vždy doctestech ke konkrétnímu programu. Doctesty využívají pomocnou funkci `cli.run`, která simuluje volání našeho programu z příkazového řádku.

Příklad doctestu:

```
>>> cli.run("python foo.py x 5")
xxxxx
yyyy
```

Znamená: Program `foo.py` voláme z příkazového řádku s argumenty `x` a `5`. Očekávaný výstup jsou dva řádky `xxxxx` a `yyyy`.

Doctesty spustíte z příkazového řádku:

```
$ python -m doctest *.py          # default mode
$ python -m doctest -v *.py       # verbose mode
```

Všechny soubory ve složce `data` jsou v kódování UTF-8, stejné kódování použijte i při zápisu výstupních souborů.

Při spouštění testů je důležité, abyste byli přímo ve složce, kde máte uložené programy, jinak nebudou sedět relativní cesty a testy neprojdou.

DÚ 10.1: Sort

Doplňte program `sort.py`, který bude volán z příkazového řádku s jedním argumentem – názvem vstupního souboru. Z tohoto souboru načte všechny řádky a vypíše je na standardní výstup abecedně seřazené.

Dále bude možné spustit program s volbou `--reverse` (nebo `-r`) – pak bude výpis řádků v opačném pořadí.

Když bude program spouštěn s nesprávnými argumenty, pak má vypsat `usage:...` (Toto automaticky zaručí modul `argparse`, nepokoušejte se používat `sys.argv`. Příklady použití `argparse` najdete v `pizza.py`).

Příklad spuštění z příkazového řádku:

```
$ python sort.py data/muses.txt -r
```

```
Urania  
Thalia  
Terpsichore  
Polyhymnia  
Melpomene  
Euterpe  
Erato  
Clio  
Calliope
```

DÚ 10.2: Hledá se Dory

Doplňte program `recursive_find.py`, který bude volán z příkazového řádku se dvěma argumenty – hledaným slovem a názvem prohledávané složky. Program vypíše všechny soubory ve složce, které obsahují hledané slovo. Složku bude prohledávat rekurzivně (tj. včetně podsložek, podpodsložek atd.).

Doctesty jsou udělané tak, že nahradí ve vašem výpisu `\` za `/`, takže je jedno, jestli je spouštíte na Unixu nebo Windowsu.

Tip: Použijte metodu `.glob` nebo `.rglob`. Tyto metody nezaručují pořadí souborů, proto na jejich výsledek aplikujte funkci `sorted`.

Příklad spuštění z příkazového řádku:

```
$ python recursive_find.py Dory data/CSFD  
data/CSFD/Hleda_se_Dory.txt  
data/CSFD/Hleda_se_Nemo.txt
```

DÚ 10.3: Kofaktory

V souboru `data/cofactors.json` jsou různé sloučeniny a k nim informace o tom, jestli jsou kofaktorem nějakého enzymu. Konkrétně každá sloučenina má v "EC" seznam enzymů, pro které je kofaktorem. (Pro přehledné zobrazení souboru ve VSCode klikněte pravým tlačítkem na text a zvolte `Format document`.)

EC čísla enzymů vyjadřují zařazení enzymů do tříd, např. enzymy začínající trojkou se nazývají *hydrolázy*.

Doplňte program `hydrolase_cofactors.py`, který načte soubor `data/cofactors.json` a vypíše všechny sloučeniny, které jsou kofaktorem nějaké hydrolázy (tj. které mají v "EC" uvedené aspoň jedno číslo začínající trojkou).

Příklad spuštění z příkazového řádku:

```
$ python hydrolase_cofactors.py data/cofactors.json
```

```
Phosphopantetheine
Nicotinamide-adenine dinucleotide
Adenosylcobalamin
Flavin adenine dinucleotide
Tetrahydrofolic acid
Coenzyme A
Flavin Mononucleotide
Menaquinone
Heme
Glutathione
S-adenosylmethionine
Thiamine diphosphate
Pyridoxal 5'-phosphate
```

(Zdroj dat: <https://www.ebi.ac.uk/pdbe/api/doc/compounds.html>, část Cofactors)

DÚ 10.4: Výběr sloupců

Doplňte program `select_columns.py`, který bude volán z příkazového řádku se několika argumenty – první argument je název vstupního CSV souboru, druhý argument je název výstupního CSV souboru a všechny další argumenty jsou názvy sloupců, které chceme vybrat ze vstupního souboru a zapsat do výstupního (tip: pro načtení názvů sloupců lze použít: `parser.add_argument('columns', nargs='*', type=str)`).

Program tedy nebude nic vypisovat na standardní výstup, pouze do výstupního souboru (viz vzorové výstupní soubory `data/auta-selected-expected.csv`, `data/contacts2-selected-expected.csv`).

Příklad spuštění z příkazového řádku:

```
$ python select_columns.py data/auta.csv data/auta-selected.csv Cena Model
```

DÚ 10.5: Kontakty

Doplňte program `convert_contacts.py`, který bude volán z příkazového řádku se dvěma argumenty – první argument je cesta ke vstupnímu JSON souboru, druhý argument je cesta k výstupnímu CSV souboru. Dále bude možné volat program s volbou `-d`

(`--delimiter`), která nastavuje oddělovač sloupců ve výstupním souboru (defaultně čárka).

Program načte ze vstupního souboru kontakty a vypíše je do výstupního souboru ve formátu CSV se sloupci `name`, `surname`, `email`, `phone`, `address`. Pokud u některé osoby některý údaj chybí nebo je `null`, do výstupního CSV se zapíše prázdné políčko.

Program tedy nebude nic vypisovat na standardní výstup, pouze do výstupního souboru (viz vzorové výstupní soubory `data/contacts1-expected.csv`, `data/contacts2-expected.csv`).

Příklady spuštění z příkazového řádku:

```
$ python convert_contacts.py data/contacts1.json data/contacts1.csv
```

```
$ python convert_contacts.py data/contacts2.json data/contacts2.csv -d  
' ; '
```

```
$ python convert_contacts.py data/contacts2.json data/contacts2.csv  
--delimiter ' ; '
```

(Všimněte si, že znak středník musí být v uvozovkách. To proto, že některé znaky (`;` `|` `<` `>` `?` `*` atd.) jsou speciálními znaky příkazového řádku (bashu/powershellu). Aby se jejich speciální funkce zrušila, zadáme je do příkazového řádku v uvozovkách. Tyto uvozovky se automaticky odstraní, tj. zadáme-li např. `' ; '`, tak Python dostane už jen řetězec o jednom znaku – středník.)