

# C2184 Úvod do programování v Pythonu

## Nepovinné úkoly

Nepovinné úkoly řešte do připravených souborů `convert_cars.py`, `minmax.py`, `monthly_temperatures.py`. 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`u. 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
yyyyy
```

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

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.

## Cvičení 10.1: Konverze CSV na JSON

Doplňte program `convert_cars.py`, který bude volán z příkazového řádku se dvěma argumenty – první argument je cesta ke vstupnímu CSV souboru, druhý argument je cesta k výstupnímu JSON souboru. Program konvertuje CSV soubor na JSON, přičemž hodnoty ze sloupce `Model` použije jako klíče do objektu ve výstupním JSONu (viz vzorový výstupní soubor `data/auta-expected.json`). Hodnoty, které lze konvertovat na `int`, konvertujte.

Příklad spuštění z příkazového řádku:

```
$ python convert_cars.py data/auta.csv data/auta-output.json
```

## Cvičení 10.2: Minimum maximum

Doplňte program `minmax.py`, který bude volán z příkazového řádku se dvěma argumenty – názvem vstupního a výstupního CSV souboru.

Program načte vstupní soubor ve formátu CSV, spočítá minimum a maximum z každého sloupce a výsledek uloží do výstupního CSV souboru (na standardní výstup nevypisuje nic).

Čísla načítejte a vypisujte jako `float`. Vzorový výstupní soubor je `data/minmax1-expected.csv`.

(Na uznání 10.2 stačí, když vám projdou první 3 doctesty. Na 10.3 vám musí projít všech 7 doctestů.)

Příklad spuštění z příkazového řádku:

```
$ python minmax.py data/table1.csv data/minmax1.csv
```

## Cvičení 10.3: Minimum maximum - rozšíření

Rozšiřte program `minmax.py` tak, aby ho bylo možné volat i s těmito volbami:

- `--header` nebo `-H` znamená, že první řádek tabulky ve vstupním souboru se má brát jako hlavička a tato hlavička se má zkopírovat i do výstupního souboru
- `--delimiter` nebo `-d` nastavuje, jaký oddělovač se má použít ve vstupním a výstupním souboru (default: čárka)

Dále doplňte toto chování: Pokud se některé hodnoty v sloupci nepodaří převést na `float` (např. první sloupec v `table2.csv`), nechte buňky odpovídající tomuto sloupci ve výstupní tabulce prázdné.

Vzorové výstupní soubory jsou `data/minmax1-expected.csv`, `data/minmax2-expected.csv`, `data/minmax3-expected.csv`.

Příklad spuštění z příkazového řádku:

```
$ python minmax.py data/table3.csv data/minmax3.csv --header  
--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.)

### **Cvičení 10.4: Teploty v Brně**

Doplňte program `monthly_temperatures.py`, který bude volán z příkazové řádky se dvěma argumenty. První argument bude název vstupního CSV souboru s tabulkou denních teplot. Druhý argument bude název výstupního souboru, do kterého program zapíše průměrné měsíční teploty (zaokrouhlené na 1 des. místo), opět ve formátu CSV.

První 4 řádky vstupního souboru jsou hlavička, v dalších řádcích je vždy rok, měsíc a teploty v jednotlivých dnech (pozor, buňky na konci mohou být prázdné, když má měsíc méně než 31 dní).

Ve výstupním souboru bude každý řádek odpovídat jednomu roku a každý sloupec jednomu měsíci. Vzorový výsledek najdete v souboru `data/Brno_monthly-expected.csv`.

```
$ python monthly_temperatures.py data/Brno_daily.csv  
data/Brno_monthly.csv
```