České vysoké učení technické v Praze FIT

Programování v Pythonu

Jiří Znamenáček

Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.

Praha & EU: Investujeme do vaší budoucnosti

Python – Serializace

Úvod

Motivace: Po dnešní hodině již umíte uložit/načíst textová a binární data. Ale co kdybyste chtěli totéž spáchat například s datovými strukturami Python'u?

> → Například proto, že provádíte složitý a dlouhý výpočet a ukládat průběžná data v nějakém binárním nebo textovém formátu, který byste při načtení museli „přeložit“, by představovalo zbytečně velkou penalizaci.
> → Nebo třeba pokládáte za zbytečné ukládat na disk (nebo třeba posílat po síti) průběh hry v nějakém čitelném tvaru ^_~

Řešení: V Python'u je k dispozici modul `pickle`, který umí zaserializovat pythonovskou datovou strukturu (téměř libovolnou) pomocí příslušného pickle-protokolu a uložit jako **binární** objekt buď do proměnné nebo do souboru. A samozřejmě ji pak zase umí z proměnné/souboru načíst a rozserializovat zpátky do datové struktury.

Protokol

(*by default* vždy posledního dostupného, což pro Python 3.x znamená verzi 3) (*by default* vždy posledního dostupného, což pro Python 3.x znamená verzi 3)

Pickle-protokolu jsou dnes již čtyři verze:

- protokol **0** – původní *human-readable* textová verze; zpětně kompatibilní
- protokol **1** – původní binární serializace
- protokol **2** – od verze Python'u 2.3
- protokol **3** – od verze Python'u 3.0

> → A další varování před verzí Python'u 3.0: Protokol 2 je v ní naimplementován jinak, než v pozdějších verzích řady 3.X.

Pokud výslovně neřeknete jinak, používá se jako výchozí hodnota příslušející

vaší verzi Python'u, takže pro Python 3.x je to verze protokolu 3.

Metody modulu „pickle“

Modul `pickle` umí piklit a rozpiklovávat v zásadě do/z dvou různých míst – paměti nebo proudu (*stream*u):

- do/z **paměti**:
  - `po = pickle.dumps(o)` – zapiklí objekt *o* do proměnné *po* typu *bytes()*
  - `o = pickle.loads(po)` – rozpiklí bajt-objekt *po* do datové struktury *o*
- do/z **proudu**:
  - `pickle.dump(o, file)` – zapiklí objekt *o* do binárního souboru *file*
  - `o = pickle.load(file)` – rozpiklí do objektu *o* zapiklenou strukturu v binárním souboru *file*

→ Všimněte si koncového *s* u metod pro piklení přes paměť.
→ *file* je objekt typu *stream*, tj. `file = open('soubor', 'rwb')`, nikoli cesta k souboru.
→ *dump*-metody mají několik dalších nepovinných parametrů, ale pokud nebudete sdílet zapiklené struktury mezi různými verzemi Python'u, může vám to být celkem jedno.

Co lze zapiklit

Zaserializovat je možné následující typy:

- `None`, `True`, `False`
- celá, reálná a komplexní čísla
- řetězce (unicodové), bajtové řetězce a bajtová pole
- n-tice, seznamy, množiny a slovníky, pokud obsahují piklitelné typy
- funkce definované na globální úrovni modulu
- vestavěné funkce definované na globální úrovni modulu
- třídy definované na globální úrovni modulu

- instance tříd, jejichž atributy `__dict__` a `__setstate__()` jsou piklitelné

Přitom:

- Pokus o zapiklení nepiklitelného objektu vyhodí výjimku *PicklingError*, blíže neurčený počet bajtů však mezitím už může být zapsán do otevřeného proudu. Podobně při pokusu o piklení velmi rekurzivních dat můžete narazit na hranice rekurze (výjimka *RuntimeError*).
- Piklení probíhá na úrovni *plně kvalifikovaných jmen*, nikoli příslušných zdrojových kódů. Tzn. že funkce a třídy jsou zapikleny pod svým jménem a jménem rodičovského modulu. Při odpiklení pak musí být příslušný modul a v něm příslušná funkce či třída k dispozici.
- Ze tříd jsou piklena pouze data instancí. To proto, aby byl zapiklený objekt použitelný i tehdy, pokud základní třídu nějak (rozumně) upravíme.

Poznámky

I. Piklící formát není nijak zabezpečen proti nebezpečným datům => **nikdy neodpiklovávejte cizí data!** Úpravou *Unpickler.find_class()* však můžete omezit, které globální funkce a třídy bude povoleno odpiklit.

II. Nejčastější použití asi najde piklení do souboru, ale stejně tak dobře můžete posílat zapiklené struktury po síti nebo ukládat do databáze.