Python

Семинар 3

Преподаватель: Дмитрий Косицин
BSU FAMCS (Fall'20)

Работа с Web

Urllib, Requests. lxml, BeautifulSoup. SQLAlchemy. Django, Flask.

Web-запросы и парсинг

В стандартной библиотеке есть модули для работы с web:

- **socket** – работа с *сокетами*
- html.parsers; **xml.etree**, xml.dom и xml.sax – html и xml *парсеры*
- **urllib** – http-*запросы*

В то же время есть множество библиотек (зачастую, надстроек либо над стандартными модулями, либо над другими библиотеками), которые использовать гораздо удобнее.

Для парсинга html / xml популярны библиотеки **lxml** и **BeautifulSoup**.

Web-запросы и парсинг

Для выполнения *запросов* (get / post), используют **requests** – использование в ней классов и менеджеров контекста упрощает работу. Библиотеку retry используют для повторения запросов (специфика соединений в web).

Замечание. При работе с запросами не забывайте про *timeout* и *retry*!

Также существует множество web-фреймворков (для написания сайтов). Полный список здесь, а вот дни из самых популярных:

- Django (и Django REST Framework для работы с REST API);
- Falcon (очень производительный);
- Flask

Для генерации страниц (шаблонов) используют Jinja2 и genshi.

Web-запросы и парсинг

Для работы с базами данных используют **SQLAlchemy**. Для работы, например, с PostgreSQL есть своя библиотека – Psycopg. В стандартной библиотеке также есть **sqlite3** для работы с SQLite databases.

Для работы с URL можно использовать библиотеку furl.

Для выполнения множества асинхронных запросов – gevent.

Для отслеживания падений сервиса используют sentry.

Замечание. Некоторые примеры см. в ipynb-приложении.

Python. Еще замечания

Интроспекция

Интерпретируемый язык позволяет легко создавать код «на лету», а также контролировать выполнение программы: проверить тип объекта, узнать сигнатуру функции, последний фрейм и все переменные в нем, стек вызовов.

Для работы со стеком и фреймами используется модуль sys, для работы с объектами – модуль inspect.

Замечание. В Python стек рекурсии ограничен (по умолчанию 1000, может быть изменена). Оптимизация хвостовой рекурсии отсутствует.

В Python есть собственный интерактивный отладчик – pdb (зачастую используется многими IDE).

Сериализация данных

Сериализация данных

- Для текста и простых объектов: json, yaml; для структур: struct.
- Для Python объектов (кроме потоков, lambda-функций, frame'ов и некоторых др.) и их передачи между процессами: pickle
- Другие библиотеки: bson, dill; protobuf

Python Zen

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Readability counts.
Special cases aren't special enough to break the rules.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
…

Spider-Man rule: With Great Power Comes Great Responsibility!