


ХОЧУ ПОМОЧЬ
ПРОЕКТУ

Модуль pickle, упаковка объектов Python



prom-steklo.ru

РЕКЛАМА

Жидкое стекло от производителя

Жидкое стекло натриевоe и калиевоe для вашего бизнеса по лучшей цене с доставкой

Перейти на сайт

Бесплатное
занятие
английским
в Яндекс
Практиуме

Полноценное занятие
с преподавателем, а не
презентация курсов

Устный тест на уровень
языка

Практика английского

Узнать больше

Модуль pickle, упаковка объектов Python

ализация структуры объекта Python

е протоколы для сериализации и десериализации структуры объекта Python. "Pickling" - го иерархия объектов Python преобразуется в поток байтов, а "unpickling" - обратная оток байтов, из двоичного файла или объекта, подобного байту преобразуется обратно в

ъектов, вы просто вызываете [функцию pickle.dumps\(\)](#). Аналогично, для десериализации [цию pickle.loads\(\)](#). Если нужно больше контроля за упаковкой и особенно распаковкой [pickle.Pickler\(\)](#) или [pickle.Unpickler\(\)](#) соответственно.

которых оканчиваются на символ 's' (dumps, loads) - работают со строками.

- Модуль pickle не защищен. Распаковывайте данные только те, которым вы доверяете. Можно сериализовать данные, которые будут выполнять произвольный код во время распаковки. Никогда не извлекайте данные, которые [могли прийти из ненадежного источника или могли быть подделаны](#).
- Подписывайте данные с помощью модуля hmas для того, чтобы быть уверенным, что они не были подделаны.
- Если часто приходится обрабатывать ненадежные данные, то используйте более безопасный формат сериализации, такие как json.

В Python есть более примитивный модуль сериализации, называемый marshal. Этот модуль существует главным образом для поддержки файлов Python .рус и ввиду его не частого применения не будет рассматривается на этом сайте. В общем случае pickle всегда должен быть предпочтительным способом сериализации объектов Python.

Сравнение с JSON.

Существуют фундаментальные различия между протоколами Pickle и JSON:

- JSON - это текстовый формат сериализации, а pickle - это двоичный формат сериализации;
- JSON читается человеком, а pickle - нет;
- JSON широко используется за пределами экосистемы Python, в то время как Pickle зависит от Python;

По умолчанию JSON может представлять только подмножество встроенных типов Python, а не пользовательские классы. Pickle может представлять чрезвычайно большое количество типов Python. Сложные случаи могут быть решены путем реализации определенных объектных API.

В отличие от pickle, десериализация ненадежного JSON сама по себе не создает уязвимости при выполнении произвольного кода.

Формат потока данных.

Формат данных, используемый [модулем pickle](#) зависит от Python. Это имеет то преимущество, что нет никаких ограничений, налагаемых внешними стандартами, такими как JSON или XDR, которые не могут представлять совместное использование указателей. Это означает, что программы, не являющиеся Python, могут не иметь возможности реконструировать выбранные объекты Python.

Вверх

По [этому](#) формат данных pickle использует относительно компактное двоичное представление. Если вам нужны оптимальные характеристики размера, вы можете [эффективно сжимать](#) упакованные данные.

В настоящее время существует 6 различных протоколов, которые можно использовать для сериализации. Чем выше используется протокол, более поздние версии Python требуется, что бы прочитать сериализованные данные.

- Протокол версии 0 является исходным "читаемым человеком" протоколом и обратно совместим с более ранними

РЕКЛАМА · 18+

practicum.yandex.ru

Бесплатное
занятие
английским
в Яндекс
Практиуме

Полноценное занятие с преподавателем, а не презентация курсов

Устный тест на уровень языка

>

Практика английского

>

Узнать больше

Первый двоичный формат, который также совместим с более ранними версиями Python. Введен в Python 2.3. Это обеспечивает намного более эффективную сборку классов

Введен в Python 3.0. Он имеет явную поддержку байтовых объектов и не может быть использован в Python 3.0–3.7.

Введен в Python 3.4. Добавлена поддержка очень больших объектов, выборка большего количества объектов и некоторые оптимизации форматов данных. Это протокол по умолчанию, начиная с Python 3.4.

Введен в Python 3.8. Добавлена поддержка внеполосных данных и ускорение внутрисписочных

Это не понятие, чем постоянство. Хотя pickle читает и записывает файловые объекты, он не может обрабатывать объекты, а также, даже более сложную проблему одновременного доступа к объектам. pickle может преобразовать сложный объект в поток байтов и может преобразовать поток байтов в объект с внутренней структурой. Возможно, наиболее очевидная вещь, которую нужно сделать с этими объектами, это записать их в файл, но также возможно отправить их по сети или сохранить их в базе данных. pickle предоставляет простой интерфейс для выбора и удаления объектов в файлах базы данных в стиле DBM.

pickle.dump() и pickle.load() функцию.

```
>>> import pickle
# используемый протокол по умолчанию
>>> pickle.DEFAULT_PROTOCOL
# 3

# создадим данные для записи
>>> data1 = {'a': [1, 2.0, 3, 4+6j],
...          'b': ('string', u'Unicode string'),
...          'c': None}
>>> selfref_list = [1, 2, 3]
>>> selfref_list.append(selfref_list)
# открываем файл для записи
>>> output = open('data.pkl', 'wb')
>>> pickle.dump(data1, output)
# список запишем по протоколу 4
>>> pickle.dump(selfref_list, output, 4)
# закрываем файл
>>> output.close()
```

В следующем примере считываются упакованные данные.

```
>>> import pprint, pickle
>>> pkl_file = open('data.pkl', 'rb')
>>> data1 = pickle.load(pkl_file)
>>> pprint.pprint(data1, width=60)
# {'a': [1, 2.0, 3, (4+6j)],
#  'b': ('string', 'Unicode string'),
#  'c': None}
>>> data = pickle.load(pkl_file)
>>> pprint.pprint(data2, width=60)
# [1, 2, 3, <Recursion on list with id=139710613355592>]
>>> pkl_file.close()
```

- [КРАТКИЙ ОБЗОР МАТЕРИАЛА.](#)
- [Какие данные можно сохранять при помощи pickle](#)
- [Функции dump\(\) и dumps\(\) модуля pickle](#)

РЕКЛАМА · 18+



 practicum.yandex.ru

**Бесплатное
занятие
английским
в Яндекс
Практиуме**
[DOCS-Python.ru™](#), 2023 г.
полноценное занятие
с преподавателем, а не
презентация курсов

Устный тест на уровень
языка >

Практика английского >

[Узнать больше](#)

[pickle](#)

[надежных источников](#)

[pickle](#)

[pickle](#)

[именам в pickle](#)

(Внимание! При копировании материала ссылка на источник обязательна)

[@docs_python_ru](#)