Можливості пакету JSON

JSON (JavaScript Object Notation) - простий формат обміну даними, заснований на підмножині синтаксису JavaScript. Модуль json дозволяє кодувати і декодувати дані в зручному форматі.

Основні функції

json.dump - серіалізуются obj як форматований JSON потік в fp.

Якщо skipkeys = True, то ключі словника не базового типу (str, unicode, int, long, float, bool, None) будуть проігноровані, замість того, щоб викликати виключення TypeError.

Якщо ensure\_ascii = True, все не-ASCII символи у висновку будуть екрановані послідовностями \ uXXXX, і результатом буде рядок, що містить тільки ASCII символи. Якщо ensure\_ascii = False, рядки запишуться як є.

Якщо check\_circular = False, то перевірка циклічних посилань буде пропущена, а такі посилання будуть викликати OverflowError.

Якщо allow\_nan = False, при спробі серіалізувати значення з коми, що виходить за допустимі межі, буде викликатися ValueError (nan, inf, -inf) в суворій відповідності зі специфікацією JSON, замість того, щоб використовувати еквіваленти з JavaScript (NaN, Infinity, -Infinity ).

Якщо indent є невід'ємним числом, то масиви і об'єкти в JSON будуть виводитися з цим рівнем відступу. Якщо рівень відступу 0, негативний або "", то замість цього просто використовуватимуться нові рядки. Значення за замовчуванням None відображає найбільш компактне уявлення. Якщо indent - рядок, то вона і буде використовуватися в якості відступу.

Якщо sort\_keys = True, то ключі виведеного словника будуть відсортовані.

json.dumps - серіалізуются obj в рядок JSON-формату.

Аргументи мають те ж значення, що і для dump ().

json.load - десеріалізует JSON з fp.

object\_hook - опціональна функція, яка застосовується до результату декодування об'єкта (dict). Використовуватися буде значення, яке цієї функцією, а не отриманий словник.

object\_pairs\_hook - опціональна функція, яка застосовується до результату декодування об'єкта з певною послідовністю пар ключ / значення. Буде використаний результат, що повертається функцією, замість справжнього словника. Якщо заданий так само object\_hook, то пріоритет віддається object\_pairs\_hook.

parse\_float, якщо визначений, буде викликаний для кожного значення JSON з плаваючою точкою. За замовчуванням, це еквівалентно float (num\_str).

parse\_int, якщо визначений, буде викликаний для рядка JSON з числовим значенням. За замовчуванням еквівалентно int (num\_str).

parse\_constant, якщо визначений, буде викликаний для наступних рядків: "-Infinity", "Infinity", "NaN". Може бути використано для порушення винятків при виявленні помилкових чисел JSON.

Якщо не вдасться десеріалізовать JSON, буде порушено виняток ValueError.

json.loads - десеріалізует s (екземпляр str, що містить документ JSON) в об'єкт Python.

Інші аргументи аналогічні аргументам в load ().

Як кодуються об’єкти Python:

>>> import json

>>> json.dumps(['foo', {'bar': ('baz', None, 1.0, 2)}])

'["foo", {"bar": ["baz", null, 1.0, 2]}]'

>>> print(json.dumps("\"foo\bar"))

"\"foo\bar"

>>> print(json.dumps('\u1234'))

"\u1234"

>>> print(json.dumps('\\'))

"\\"

>>> print(json.dumps({"c": 0, "b": 0, "a": 0}, sort\_keys=True))

{"a": 0, "b": 0, "c": 0}