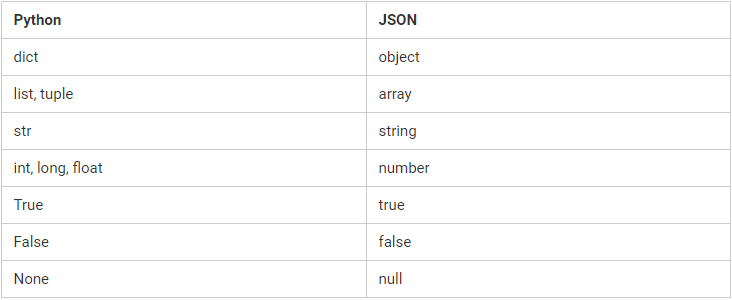
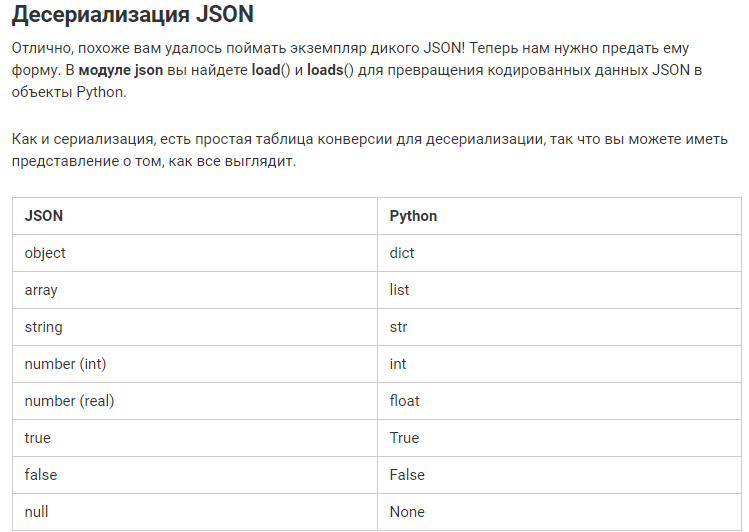
Использование JSON



# рассмотрим формат передачи данных json  
# в python json можно кодировать (сериализовать)  
# или декодировать (десериализовать)  
# команда dump(arg1, arg2) записывает в файл  
# команда dumps(arg1) записывает в переменную  
  
  
import json  
  
  
data = {  
 "name": "Kostan",  
 "age": 32,  
 "color": "red",  
 "specification": [1, 2, 3]  
}  
  
data\_clone = {  
 "name": "Tolan",  
 "age": 42,  
 "color": "red",  
 "specification": [7, 2, 3]  
}  
  
# первый способ записи в файл  
with open("data.json", "w") as file:  
 json.dump(data, file)  
  
# второй способ записи в файл  
temp\_file = open("data.json", "w")  
json.dump(data\_clone, temp\_file)  
  
# запись json как строки  
json\_temp = json.dumps(data)  
print(json\_temp, type(json\_temp), sep="\n"

# при дессериализации мы можем получить объект где  
# типы кортежа могут стать массивами и подобные преобразования  
# который глобально не изменят объект  
# есть команды loads для преобразования из строки  
# есть команда load для загрузки из файла  
  
import json  
  
data = {  
 "name": "Koral",  
 "age": 320,  
 "color": "green",  
 "mass": [12, 43, 211]  
}  
  
data = json.dumps(data)  
print(data, type(data), sep="\n")  
  
data = json.loads(data)  
print(data, type(data), sep="\n")



# в данном примере разберём пользовательское кодирование и декодирование данных  
# можно создать методы, которые будут считываться перед сериализацией и десериализацией данных  
  
  
import json  
  
  
class Magic(object):  
  
 def \_\_init\_\_(self, name, age, power):  
 self.name = name  
 self.age = age  
 self.power = power  
  
 # сериализуем данные  
 @staticmethod  
 def get\_data(self):  
 return {  
 "name": self.name,  
 "age": self.age,  
 "power": self.power  
 }  
  
 # тестовый метод класса  
 @classmethod  
 def get\_class(cls, age):  
 return cls(age, 32, 100)  
  
 # десериализуем данные  
 @staticmethod  
 def des\_data(data):  
 return [data["name"], data["age"], data["power"]]  
  
  
# сериализуем данные  
mag = Magic.get\_class(120)  
data = json.dumps(mag, default=Magic.get\_data)  
print(data, type(data))  
  
  
# десериализуем данные  
data = json.loads(data, object\_hook=Magic.des\_data)  
print(data, type(data))