# Модуль pickle

Модуль pickle реализует мощный алгоритм сериализации и десериализации объектов Python. "Pickling" - процесс преобразования объекта Python в [поток байтов](https://pythonworld.ru/tipy-dannyx-v-python/bajty-bytes-i-bytearray.html), а "unpickling" - обратная операция, в результате которой поток байтов преобразуется обратно в Python-объект. Так как поток байтов легко можно [записать в файл](https://pythonworld.ru/tipy-dannyx-v-python/fajly-rabota-s-fajlami.html), модуль pickle широко применяется для сохранения и загрузки сложных объектов в Python.

Не загружайте с помощью модуля pickle файлы из ненадёжных источников. Это может привести к необратимым последствиям.

Модуль pickle предоставляет следующие функции для удобства сохранения/загрузки объектов:

**pickle.dump**(obj, file, protocol=None, \*, fix\_imports=True) - записывает сериализованный объект в файл. Дополнительный аргумент protocol указывает используемый протокол. По умолчанию равен 3 и именно он рекомендован для использования в Python 3 (несмотря на то, что в Python 3.4 добавили протокол версии 4 с некоторыми оптимизациями). В любом случае, записывать и загружать надо с одним и тем же протоколом.

**pickle.dumps**(obj, protocol=None, \*, fix\_imports=True) - возвращает сериализованный объект. Впоследствии вы его можете использовать как угодно.

**pickle.load**(file, \*, fix\_imports=True, encoding="ASCII", errors="strict") - загружает объект из файла.

**pickle.loads**(bytes\_object, \*, fix\_imports=True, encoding="ASCII", errors="strict") - загружает объект из потока байт.

Модуль pickle также определяет несколько [исключений](https://pythonworld.ru/tipy-dannyx-v-python/isklyucheniya-v-python-konstrukciya-try-except-dlya-obrabotki-isklyuchenij.html):

* **pickle.PickleError**
  + **pickle.PicklingError** - случились проблемы с сериализацией объекта.
  + **pickle.UnpicklingError** - случились проблемы с десериализацией объекта.

Этих функций вполне достаточно для сохранения и загрузки встроенных типов данных.

>>>

**>>> import** **pickle**

**>>>** data = {

**...**  'a': [1, 2.0, 3, 4+6j],

**...**  'b': ("character string", b"byte string"),

**...**  'c': {None, True, False}

**...** }

>>>

**>>> with** open('data.pickle', 'wb') **as** f:

**...**  pickle.dump(data, f)

**...**

**>>> with** open('data.pickle', 'rb') **as** f:

**...**  data\_new = pickle.load(f)

**...**

**>>> print**(data\_new)

{'c': {False, True, None}, 'a': [1, 2.0, 3, (4+6j)], 'b': ('character string', b'byte