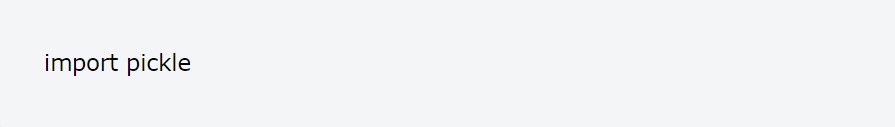
**Pickling and Unpickling in Python**

If we wish to write a structure such as a list or a dictionary to a file and read it subsequently, we need to use the Python module pickle. **Picking** refers to the process of converting structure to a byte stream before writing to the file. While reading the contents of the file a reverse process called **Unpickling** is used to convert the byte stream back to the original structure.

Pickle is used for serializing and de-serializing Python object structures, also called marshalling or flattening. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network. Later on, this character stream can be retrieved and de-serialized back to a Python object. Pickling is not to be confused with compression! The former is the conversion of an object from one representation (data in Random Access Memory(RAM)) to another (text on disk), while the latter is the process of encoding data with fewer bits, in order to save disk space.

“Pickle” method can be used to store any kind of object in a file as it allows us to store Python objects with their structures. So for storing data in binary format we will use the “pickle” method.

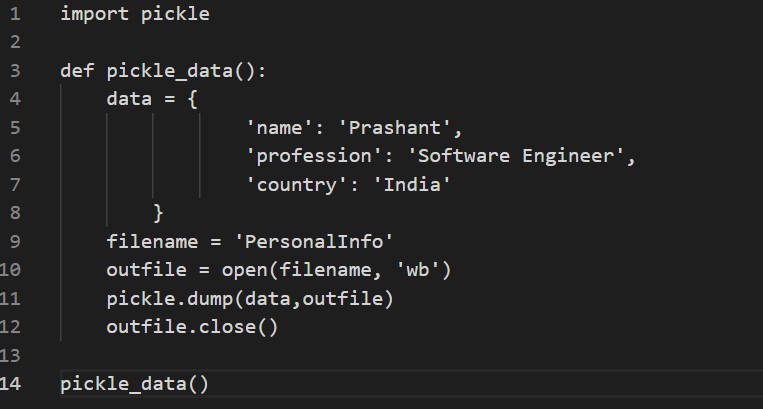
Firstly, we need to import the module. It provides two main methods for the purpose- dump and load.



**dump:** For creation of a binary file we will use pickle.dump() to write object in the file, which is opened in binary access mode. Syntax of dump() method: dump(object,fileobject)

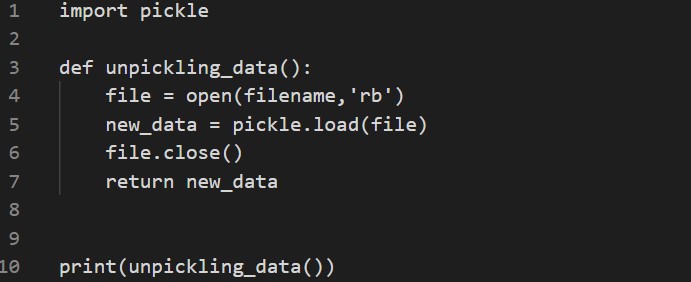
**load:** Once data is stored in the file using dump(), then we can read the stored data using pickle.load() method. Syntax of load() method: object=load(fileobject)

**Example of Pickling**

****

To Open the file we have used the ‘open’ function, The first argument should be the name of the file and the second argument is ‘wb’, wb refers to writing in binary mode. This means that the data will be written in the form of byte objects.

**Example of Unpickling**

****

Here **r** stands for reading mode and **b** stands for binary mode. The output of the Unpickling function is the same as that of the pickled function :



**What can be done with Pickle?**

Pickling is useful for applications where we need some degree of persistence in our data. The program’s state data can be saved to disk, so we can continue working on it later. It can also be used to send data over a Transmission Control Protocol(TCP) or socket connection, or to store python objects in a database. Pickle is very useful for when working with Machine Learning Algorithms, where we want to save them to be able to make new predictions at a later time, without having to rewrite everything or train the model all over again.

## **What can be pickled?** The following data types can be pickled:

## Booleans

## Integers

## Floats

## Complex numbers

## Normal and Unicode Strings

## Tuples

## Lists

## Sets

* Dictionaries that contain pickable objects

All the above can be pickled(easily), though: examples of this are generators, inner classes,lambda functions and defaultdicts. In the case of lambda function, we need to use an additional package named ‘dill’ . With defaultdicts, we need to create them with a module-level function.

## 

## **Common use cases of Pickling**

1. To save a program’s state data to disk so that it can carry on where it left off when restarted (persistence)
2. Sending Python data over a TCP connection in a multi-core or distributed system (marshalling)
3. Storing Python objects in a database

## **Dangers of Pickling**

The documentation of the Pickle module states:

“The Pickle module is not secure against erroneous or maliciously constructed data. Never unpickle data received from an untrusted or unauthenticated source.”

## **Safe implementation of Pickle**

* The Pickle should never be used between unknown parties.
* Ensure the parties exchanging Pickle have an encrypted network connection.
* In case of unsecured connection, any alteration in Pickle can be verified by using a cryptographic signature. Pickle can be signed before the transmission, and its signature can be verified before loading it on the receiver side.