CH 99 : Object Serialization in python

Object Sterilization is a technique by which you can store and restore python object in your file system. It is very useful when you want to save, reuse, or send to someone else.

For example, you are playing a game. After some time, you want to stop the game and resume later on. To achieve that, you need to store current state of game and restore back when you resume.

For cases like this, the pickle module is ideal. It’s part of the Python standard library. So what can the pickle module store ?

1. All native datatype like  booleans, integers, floating point numbers, complex numbers, strings.
2. Lists, tuples, dictionaries, and sets containing any combination of native data types.
3. Function, Class and Instance of the class.

Here we use two term, Pickling and Unpicking:

* Pickling is the process whereby a Python object hierarchy is converted into a byte stream, and
* Unpickling - is the inverse operation, whereby a byte stream is converted back into an object hierarchy.

Pickling (and unpicking) is also known as serialization, marshalling, or flattening.

**99.1 Pickling Data**

Example #1:

Following example shows that I have created two objects: student and colors and save them in data.pk1 file.

import pickle

student = {'name': 'Python',

'roll': ('class roll', 10),

'subjects': ['python','djnago']

}

colors = ['red', 'green', 'yellow']

output = open('data.pkl', 'wb')

# Pickle our data

pickle.dump(student, output) # Dump First Objects

pickle.dump(colors, output) # Dump Second Objects

output.close() # Close the output file

**99.1 Unpickling Data**

Now, you can see data.pk1 file is generated. Next, Example shows how we can get the data back from that file:

import pickle

inputfile= open('data.pkl', 'rb')

student = pickle.load(inputfile)

print student # will print the student data

colors = pickle.load(inputfile)

print colors

inputfile.close()

Thus you can see how  pickle.dump() / pickle.load() can be used to save and store python Objects.

99.3: **Pickling Without a File**

It’s also possible to do picking and unpicking without using file by just creating picking file objects. For Example:

count = 100

a = pickle.dumps(count)

print type(a) #what is type of a

b = pickle.loads(a)

print b

99.4 Serializing Python Objects in JSON

Sometimes it requite to save the data in well-known format, such that it can be used by other programming languages (like Java). There are number of well-known format to specify data-set , like XML, JSON etc. Here we will see how we can store a python object in JSON format.

Below Code snappit shows how a python object is store in JSON file.

import json

student = {'name': 'Python',

'roll': ('class roll', 10),

'subjects': ['python','djnago']

}

colors = ['red', 'green', 'yellow']

output = open('data.json', 'w')

#Pickle our data

json.dump(student, output)

json.dump(colors, output)

output.close()

It will produce data.json file, which contains json representation of python Objcets. If you open this file in notepad, you can find:

{"subjects": ["python", "djnago"], "name": "Python", "roll": ["class roll", 10]}["red", "green", "yellow"]

Which is more readable, then pickle file and can be feed into any programming language.

Similarly we can feed any JSON file in a python program and can retrieve python object again. Following example illustrate this:

import json

inputfile= open('data.json', 'r')

student = json.load(inputfile)

print student

colors = json.load(inputfile)

print colors

inputfile.close()