Python Persistence

Python 3.x 3.5

Examples

Python Persistence

Objects like numbers, lists, dictionaries,nested structures and class instance objects live in your computer's memory and are lost as soon as the script ends.

pickle stores data persistently in separate file.

pickled representation of an object is always a bytes object in all cases so one must open files in `wb` to store data and `rb` to load data from pickle.

the data may may be off any kind , for example,

```
data={'a':'some_value',
     'b':[9,4,7],
     'c':['some_str','another_str','spam','ham'],
     'd':{'key':'nested_dictionary'},
     }
```

Store data

```
import pickle
file=open('filename','wb')  #file object in binary write mode
pickle.dump(data,file)      #dump the data in the file object
file.close()                #close the file to write into the file
```

Load data

```
import pickle
file=open('filename','rb')  #file object in binary read mode
data=pickle.load(file)      #load the data back
file.close()

>>>data
{'b': [9, 4, 7], 'a': 'some_value', 'd': {'key': 'nested_dictionary'},
 'c': ['some_str', 'another_str', 'spam', 'ham']}
```

The following types can be pickled

1. None, True, and False
2. integers, floating point numbers, complex numbers
3. strings, bytes, bytearrays
4. tuples, lists, sets, and dictionaries containing only picklable objects
5. functions defined at the top level of a module (using def, not lambda)
6. built-in functions defined at the top level of a module
7. classes that are defined at the top level of a module
8. instances of such classes whose **dict** or the result of calling **getstate** ()

Function utility for save and load

Save data to and from file

```
import pickle
def save(filename,object):
    file=open(filename,'wb')
    pickle.dump(object,file)
    file.close()

def load(filename):
    file=open(filename,'rb')
    object=pickle.load(file)
    file.close()
    return object


>>>list_object=[1,1,2,3,5,8,'a','e','i','o','u']
>>>save(list_file,list_object)
>>>new_list=load(list_file)
>>>new_list
[1, 1, 2, 3, 5, 8, 'a', 'e', 'i', 'o', 'u'
```

Syntax

```
pickle.dump(obj, file, protocol=None, *, fix_imports=True)
```

```
pickle.load(file, *, fix_imports=True, encoding="ASCII", errors="strict")
```

Parameters

Parameter	Details
obj	pickled representation of obj to the open file object file
protocol	an integer, tells the pickler to use the given protocol, `0` -ASCII, `1` - old binary format
file	The file argument must have a write() method `wb` for *dump* method and for loading read() method `rb`

Remarks