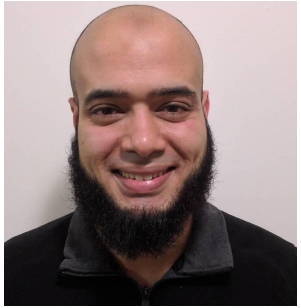
Python Programming

Pickle Module

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Little about binary mode

- rb and wb modes are for reading and writing binary
  - It writes bytes (8 0s/1s).
- It is not so convenient, so we use modules that makes our life easier

```
lst = [120, 255, 100]

with open("data.binary", "wb") as writer:
    binary_format = bytearray(lst)  # must be in range(0, 256)
    writer.write(binary_format)
    str_encoded = bytearray('abc', 'utf-8')
    writer.write(str_encoded)

with open("data.binary", "rb") as reader:
    lst2 = list(reader.read())
    print(lst2)     # [120, 255, 100, 97, 98, 99]
    # a integer code is 97
```

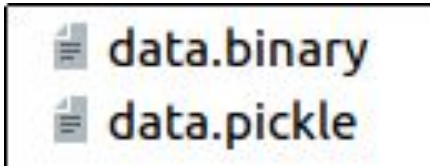
Pickle module

- We can use to trivially create binary files of arbitrary objects
- We can also use it with our user-defined classes
  - Future: we can use special methods __setstate__, __getstate__ or __reduce__
    - E.g. Pickle don't know how to handle your opened file!

```python
import pickle
# Pickle serializes objects in a file.
# Serialization is the process of converting an object into a stream of bytes

data = (2021, '4444', ((7, 'wow'), [4, 5]))
lst = [1, 251221, 30000]    # > 256

with open("data.pickle", "wb") as pickle_file:
    pickle.dump(data, pickle_file)
    pickle.dump(lst, pickle_file)
```



Reading pickle file

- We can read in a trivial way
- Just remember rb mode
- Overall, easy read & write

```python
import pickle

with open("data.pickle", "rb") as pickle_file:
    data = pickle.load(pickle_file)
    lst = pickle.load(pickle_file)
    print(data)
    print(lst)

"""
(2021, '4444', ((7, 'wow'), [4, 5]))
[1, 2, 3]
"""

# Observe: we read/write full thing
# Always overwrite
# Try to corrupt and read
```

What is wrong with pickle?

- **Performance**: Full file loading, which is not efficient for huge files
- **Security**: if a hacker replaced your pickle file (or give), his pickle file can contain commands to be run (e.g. **os.system** to delete your system files)
- If your class variables **restructured** ⇒ old pickle file is useless!
- **No control** on how to serialize things that might be saved in different ways
- It seralizes everything by default, which might be a problem (e.g. File object)
  - You need to be more careful or do workarounds
- **__init__** isn't called for objects creation
- Mainly a **python** binary file (dependent). Also as binary = Unreadable
- When to use? Personnel local projects. Security issue is very critical one
- There are other alternatives (shelve, json, etc). Each has pros/cons

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."