

Python Programming

Shelve 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

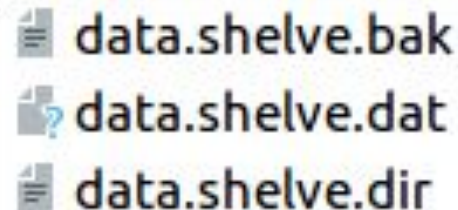
Ex-(Software Engineer / ICPC World Finalist)



Writing

- It is like-dictionary. Values are pickled / unpickled
- Behind the scene, like a database based on key-value (key is string)

```
3 import shelve
4
5 data = (2021, '4444', ((7, 'wow'), [4, 5]))
6 lst = [1, 251221, 30000] # > 256
7
8 # By default, the underlying database
9 # file is opened for reading and writing
10 with shelve.open('data.shelve') as shelf:
11     # Think like a dictionary. Key/value
12     shelf['data'] = data
13     shelf['lst'] = lst
14     # Use strings as keys
15     # shelf[10] = 20 # 'int' object has no attribute 'encode'
```



data.shelve.bak
data.shelve.dat
data.shelve.dir

Reading

- You can get keys similar to a dictionary
- Use it to access all or a specific keys
 - **Only** accessed values are **loaded** = memory efficient
 - Recall: Pickle loads all

```
3 import shelve
4
5 with shelve.open('data.shelve', 'r') as shelf:
6     for key in shelf.keys():
7         # load this specific value
8         print(key, shelf[key])
9
10
11 data (2021, '4444', ((7, 'wow'), [4, 5]))
12 ⚡ [1, 251221, 30000]
13
```

Updating shelf

- To update, just use the shelf[key] = value
 - Now these entries are updated/added

```
3 import shelve
4
5 data = (2021, '4444', ((7, 'wow'), [4, 5]))
6 lst = [1, 251221, 30000] # > 256
7
8 # let's open the same file.
9 # but we will use different keys
10 with shelve.open('data.shelve') as shelf:
11     # Think like a dictionary. Key/value
12     shelf['data_cusomter'] = data
13     shelf['numbers'] = lst
```

Mistake 1: Updating shelve *the wrong way*

```
1  import shelve
2
3  with shelve.open('data.shelve') as shelf:
4      shelf['numbers'].append(1111)
5
6  with shelve.open('data.shelve', 'r') as shelf:
7      for key in shelf.keys():
8          print(key, shelf[key])
9
10 """
11 data_cusomter (2021, '4444', ((7, 'wow'), [4, 5]))
12 numbers [1, 251221, 30000]
13
14 NO UPDATE
15
16 Right way: get the list. update. assign
17 """
```

Mistake 2: Only new items are there!

```
5      #.let's read again
6
7      with shelve.open('data.shelve', 'r') as shelf:
8          for key in shelf.keys():
9              #.load this specific value
10             print(key, shelf[key])
11
12     """
13     data (2021, '4444', ((7, 'wow'), [4, 5]))
14     lst [1, 251221, 30000]
15     data_cusomter (2021, '4444', ((7, 'wow'), [4, 5]))
16     numbers [1, 251221, 30000]
17
18     Surprise! Old keys exist
19     - the open command in writing one, load the saved files
20     - It doesn't remove them. just load. so old keys exist!
21
22     """
```

Deleting keys

- Explicitly delete the keys

```
1  import shelve
2
3  # let's open and delete
4  with shelve.open('data.shelve') as shelf:
5      del shelf['data'] # make sure it exists!
6      del shelf['lst']
7
8  with shelve.open('data.shelve', 'r') as shelf:
9      for key in shelf.keys():
10         # load this specific value
11         print(key, shelf[key])
12
13  """
14  data_cusomter (2021, '4444', ((7, 'wow'), [4, 5]))
15  nmbers [1, 251221, 30000]
16  """
```

Shelve Cons

- Shelve files also have the **same security issue** as pickle
- They might be slower: pickle and unpickle the values
- It shouldn't be used with **concurrent access**
 - Don't open the same file with 2 apps in same time, probably may fail
 - Databases are the way to go (You should study later)
- Might be convenient for your local apps
- Overall, very similar limitations to pickle,
 - but more flexible access
 - It doesn't load whole data in memory.
 - Behind the scene, file-based like a database

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”