

Python: do básico à autonomia intermediária

aula 7

Bernardo Alves Furtado

March 14, 2022

Menu do dia

Persistence

Files

Pickle

my_first_file.py

- ▶ `with open('nome.txt', 'w') as f:`
 `f.write('your text')`
- ▶ `with open('nome.csv/txt/png', 'w/a/r') as`
 `f/g/h/qq coisa:`
 `f.read/write(variable/text/str)`
- ▶ Vantagens:
 - ▶ open and closes
 - ▶ gives you `f`, an object that is a handle
 - ▶ simple
 - ▶ Note: tudo é string

Python persistence obj

Pickle é um serializador próprio do Python

Eficiente: salva (quase) qualquer objeto python

Não perde configuração

Desvantagem: como é serializado, não se pode confiar em pickles recebidos externamente (não é verificável)

Utilizado internamente (seu próprio programa), perfeito!

my_first_pickle.py

```
import pickle
with open('nome_arquivo', 'wb') as f:
    pickle.dump(objeto_a_salvar, f)
with open('nome_arquivo', 'rb') as f:
    novo_objeto = pickle.load(f)
```

my_first_json.py

É inspecionável (não é binário, como pickle)

Intercambiável entre linguagens

Mas, no fundo, parece um dicionário

```
import json
```

```
with open('nome_arquivo.json', 'w') as f:
```

```
    json.dump(objeto_a_salvar, f)
```

```
with open('nome_arquivo.json', 'r') as f:
```

```
    novo_objeto = json.load(f)
```

Exercise1

1. Busque alguma função feita que produza outputs e exporte em forma de arquivos
2. Teste os diferentes métodos.

Exercise2: dictionary_json_example.py

1. Especificamente: crie uma função que gera um dicionário que funciona como base de dados.
2. Cada vez que ele se alterar (você sair do programa), salve-o em arquivo.
3. Na próxima abertura do arquivo, load (leia o arquivo) e continue a formação da base.
4. Detalhes: use `from collections import defaultdict`
5. Use `defaultdict(dict)`
6. Use método `update d.update()`
7. `update` recebe uma key e um dicionário: `key_name, {key: value}`