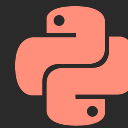




Arquivos temporários com tempfile

Live de Python # 201



1. O que são arquivos temporários

E por que precisamos deles?

2. A biblioteca tempfile

Um básico sobre o uso

3. Performance

Não enchendo a memória de porcaria

4. Segurança

Quando as coisas puderem, elas darão errado



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Acássio Anjos, Ademar Peixoto, Alexandre Harano, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rocha, Aquiles Coutinho, Arnaldo Turque, Bloquearsites Farewall, Bruno Barcellos, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, César Almeida, Christiano Morais, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, Dartz Dartz, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Douglas Braga, Douglas Martins, Douglas Zickuhr, Eli Júnior, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Fernando Sousa, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Dettenborn, Heitor Fernandes, Henrique Junqueira, Igor Taconi, Ismael Ventura, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, João Lugão, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, Jorge Plautz, Jose Mazolini, Juan Gutierrez, Juliana Machado, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Botassio, Leandro Miranda, Leonardo Cruz, Leonardo Mello, Leonardo Nazareth, Lucas Adorno, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Praciano, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mario Deus, Mateus Lisboa, Matheus Silva, Matheus Vian, Mírian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Patrick Brito, Patrick Felipe, Paulo Tadei, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Rafael Lopes, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renan Moura, Renato Veirich, Riverfount Riverfount, Robson Maciel, Rodrigo Brandao, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago S, Tiago Minuzzi, Tony Dias, Victor Wildner, Vinícius Bastos, Vinicius Figueiredo, Vítor Gomes, Vitor Luz, Vlademir Souza, Vladimir Lemos, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Yury Barros



Obrigado você



Por que precisamos
disso?

Temp
files

Arquivos temporários



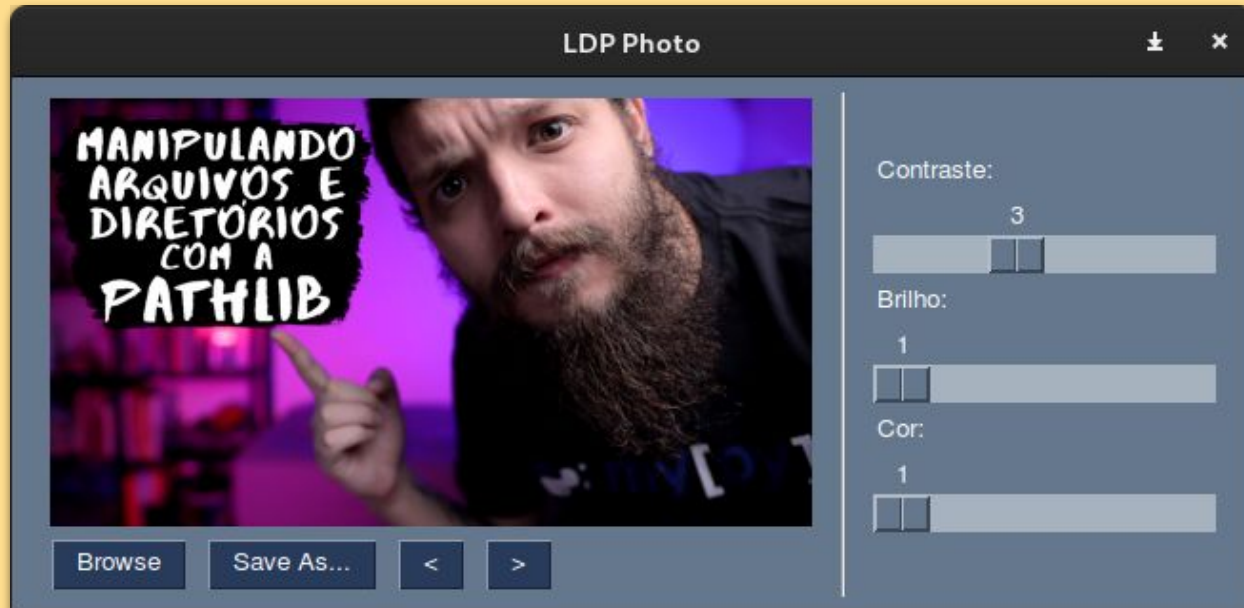
Um arquivo temporário é um arquivo criado para armazenar informações temporariamente.

- Quando um programa precisa armazenar algo por um tempo
- Quando é preciso juntar arquivos para fornecer um novo
- Quando não existe memória o suficiente para trabalhar com os dados na RAM
- Quando 2 processos diferentes precisam se comunicar
- Quando alguma biblioteca precisa de um arquivo e você não tem
- ...

Categorização



- **Cache**
 - Quando um programa precisa armazenar algo por um tempo
 - Quando não existe memória o suficiente para trabalhar com os dados na RAM
- **Transição / Cache**
 - Quando é preciso juntar arquivos para fornecer um novo
 - **Testes**
 - Quando alguma biblioteca precisa de um arquivo e você não tem
- **IPC**
 - Quando 2 processos diferentes precisam se comunicar
- ...



Exemplificando o uso



Temp
file

A biblioteca

A biblioteca



A biblioteca **tempfile** é nativa no python e conta com formas de criar:

- Arquivos temporários
- Diretórios temporários

Cria arquivos de forma "**segura**"

Como funciona?



```
1  from tempfile import TemporaryFile
2
3  with TemporaryFile() as file:
4      file.write(b'Batatinhas Fritas')
5      file.seek(0)
6      print(file.read())
```

Como funciona?



Mesma forma de
uso do open()

```
1  from tempf
2
3  with TemporaryFile() as file:
4      file.write(b'Batatinhas Fritas')
5      file.seek(0)
6      print(file.read())
```

Como funciona?



Modo default:
w+b
(Escrita em bytes)

```
1 from tempfile
2
3 with TemporaryFile() as file:
4     file.write(b'Batatinhas Fritas')
5     file.seek(0)
6     print(file.read())
```

Para escrever com strings



```
1  from tempfile import TemporaryFile
2
3  with TemporaryFile(mode='w+') as file:
4      file.write('Batatinhas Fritas')
5      file.seek(0)
6      print(file.read())
```

Diretórios temporários



```
1  from pathlib import Path
2  from tempfile import TemporaryDirectory
3
4  with TemporaryDirectory() as temp_dir:
5
6      file_in_temp = Path(temp_dir) / 'exemplo.txt'
7      with file_in_temp.open(mode='w+') as file:
8          file.write('Batatas!')
```

Diretórios temporários



```
1 from pathlib import Path
2 from tempfile import TemporaryDirectory
3
4 with TemporaryDirectory() as temp_dir:
```

```
5
6     file_in_temp = Path(temp_dir) / 'exemplo.txt'
7     with file_in_temp.open(mode='w+') as file:
8         file.write('Batatas!')
```

Será deletado,
mesmo não sendo
temporário

Arquivos com Nomes



```
1  from tempfile import NamedTemporaryFile
2
3  with NamedTemporaryFile() as temp_file:
4      print(temp_file.name)
```



Existem diversas formas de uso que se pode dar para esses objetos e por isso, podemos usar diferentes sufixos e prefixo:

- **prefix:** Início padrão para os nomes
 - `prefix='batata_'`
 - Todos os arquivos criados temporariamente terão o seu nome começado em 'batata_'
- **suffix:** Mesma coisa para o final do arquivo.
 - Pode ser útil para extensões de arquivo
 - **OBS: arquivos temporários não tem extensão por padrão**
- **remove:** Você pode não mover o arquivo de forma automática
 - `remove=False`

Juntando tudo



```
1  from tempfile import NamedTemporaryFile
2
3  with NamedTemporaryFile(
4      prefix='ldp',
5      suffix='.txt',
6      mode='w+'
7  ) as temp_file:
8      print(temp_file.name)
```

Utilitários do tempfile



```
1  import tempfile
2
3  # Retorna o nome da pasta temporária
4  tempfile.gettempdir() # /tmp
5
6  # Muda o path para .cache
7  tempfile.tempdir = '.cache'
8
9  tempfile.gettempdir() # .cache
```

Chamadas baixo nível



Temos duas chamadas de baixo nível

- **mkstemp**: Para criar arquivos
- **mkdtemp**: Para criar diretórios

Essas chamadas criam arquivos e diretórios seguros, mas não os deletam.

Usando o tempfile a
nosso favor

Perfor
mance

io.BytesIO



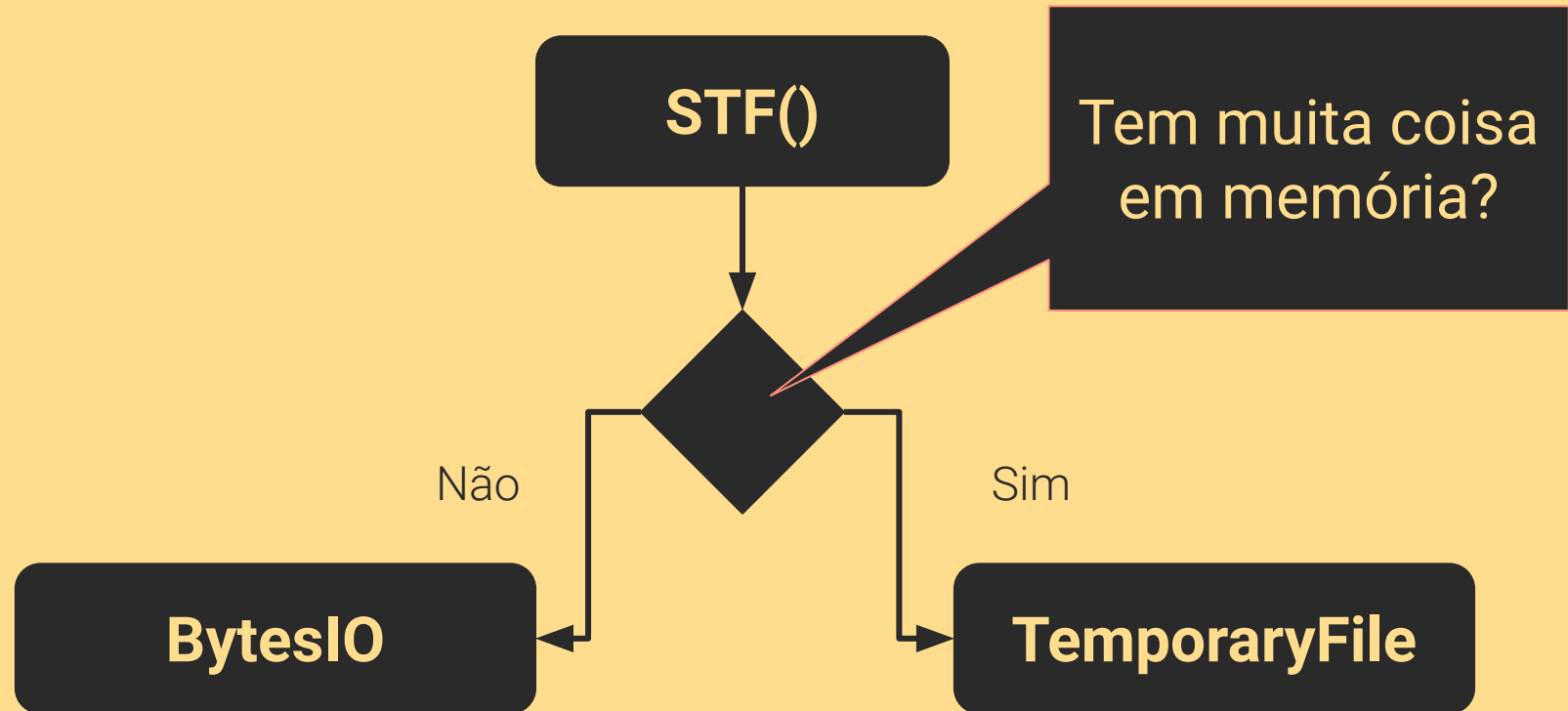
Alguém deve ter se perguntado, mas pra que arquivos? Posso botar tudo em memória!

```
1  from io import BytesIO
2
3  with open("image_test.jpg", 'rb') as file:
4      # Colocando 4.4mb em memória
5      em_memoria = BytesIO(file.read())
```

Como não colocar em memória?



O **SpooledTemporaryFile**. É um arquivo em tipo diferente de arquivo temporário na templib.



SpooledTemporaryFile



```
1  from tempfile import SpooledTemporaryFile
2
3  um_mb = 1_000_000
4
5  with SpooledTemporaryFile(max_size=um_mb) as stf:
6      em_memoria = stf.write(file.read())
7      print(stf.name)  # Não é nomeado
```

SpooledTemporaryFile



```
1  from tempfile import SpooledTemporaryFile
2
3  um_mb = 1_000_000
4
5  with SpooledTemporaryFile(max_size=um_mb) as stf:
6      em_memoria = stf.write(file.read())
7      print(stf.name)  # Não é nomeado
```

Bora fazer um teste só
pra ver o que rola?



`bytesio__case.py`



Segur
ança

Tendo código menos
previsível

/tmp e hardcode



NUNCA escreva temporários na mão.

```
1  from pathlib import Path
2
3  path = '/tmp'
4  file = Path(f'{path}/xpto.txt')
5
6  if not file.exists():
7      with file.open() as file:
8          file.write('Batata!')
```

```
1  from os.path import exists
2
3  path = '/tmp'
4  file = f'{path}/xpto.txt'
5
6  if not exists(file):
7      with open(file, 'w') as file:
8          file.write('Batata!')
```

Os arquivos criados com `open()` e `Path.open()`



- Não são arquivos seguros
- Outros users podem ver
- Processos podem escrever

```
1  from os.path import exists
2
3  file = f'xpto.txt'
4
5  with open(file, 'w') as file:
6      file.write('Batata!')
```

Os arquivos criados com open() e Path.open()



- Não são arquivos seguros
- Outros users podem ver
- Processos podem escrever

```
py-3.10.4 babbage in ~  
o → lla xpto.txt  
.rw-r--r-- 7 dunossauro 2 mai 20:37 xpto.txt
```



`pip install`



Rodando o bandit



```
|.venv|py-3.10.4 babbage in ~/live_201
o → bandit exemplo_0.py
[main] INFO profile include tests: None
[main] INFO profile exclude tests: None
[main] INFO cli include tests: None
[main] INFO cli exclude tests: None
[main] INFO running on Python 3.10.4
[node_visitor] WARNING Unable to find qualified name for module: exemplo_0.py
Run started:2022-05-02 23:41:23.708879

Test results:
>> Issue: [B108:hardcoded_tmp_directory] Probable insecure usage of temp file/directory.
Severity: Medium Confidence: Medium
CWE: CWE-377 (https://cwe.mitre.org/data/definitions/377.html)
Location: exemplo_0.py:4:7
More Info: https://bandit.readthedocs.io/en/1.7.4/plugins/b108\_hardcoded\_tmp\_directory.html
3
4     path = '/tmp/'
5     filename = f'{path}xpto.txt'
```

Permissões



```
1  from tempfile import NamedTemporaryFile
2
3  with NamedTemporaryFile(
4      prefix='permissoes',
5      delete=False
6  ) as file:
7      file.write(b'batatinha')
```

Permissões



```
1 from tempfile import NamedTemporaryFile
2
3 with NamedTemporaryFile(
4     prefix='permissoes',
5     delete=False
```

py-3.10.4 babbage in ~

o → lla /tmp/permissoes*

```
.rw----- 7 dunossauro 2 mai 20:45 /tmp/permissoesfcix8e4i
.rw----- 0 dunossauro 2 mai 20:45 /tmp/permissoeslbmbasyga
```

TOCTOU



Existe uma vulnerabilidade famosa a respeito de arquivos temporários chamada **TOCTOU** (Time to check, Time to Use).

Ela trava o modo de leitura de um arquivo e nada consegue escrever lá.
Race condition.

```
1  while True:
2      file = open('/tmp/blah', 'w+')
3      from time import sleep
4      sleep(.1) # só pra conseguir fechar!
```



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3

