



Google Fire

Live de Python # 157



picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto



Ademar Peixoto, Alex Lima, Alexandre Harano, Alexandre Santos, Alexandre Souza, Alexandre Tsuno, Alysso Oliveira, Amaziles Carvalho, Anderson Araujo, Andre Rodrigues, André Rocha, Antonio Neto, Arnaldo Turque, Bianca Rosa, Bruno Oliveira, Caio Nascimento, Carlos Cardoso, Carlos Chiarelli, César Almeida, César Moreira, Davi Ramos, David Kwast, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Edgard Sampaio, Elias Soares, Eugenio Mazzini, Everton Alves, Fabio Barros, Fabio Castro, Fabrício Coelho, Flavkaze Flavkaze, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Felitti, Guilherme Marson, Guilherme Ostrock, Haroldo Júnior, Henrique Machado, Hélio Neto, Isaac Ferreira, Israel Fabiano, Italo Silva, Jeison Sanches, Johnny Tardin, Jonatas Leon, Jorge Plautz, José Prado, Jovan Costa, João Lugão, João Schiavon, Juan Gutierrez, Jônatas Silva, Júlia Kastrup, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lorena Ribeiro, Lucas Barros, Lucas Ferreira, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luiz Lima, Maiquel Leonel, Maiquel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Natan Cervinski, Nicolas Teodosio, Nilo Pereira, Otavio Carneiro, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Andrade, Pedro Pereira, Peterson Santos, Reinaldo Silva, Rodrigo Campos, Rodrigo Ferreira, Rodrigo Vaccari, Ronaldo Silva, Rubens Gianfaldoni, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Valdir Junior, Victor Geraldo, Vinícius Bastos, Vinícius Ferreira, Wesley Mendes, Willian Lopes, Willian Lopes, Willian Rosa, Wilson Duarte



Obrigado você





1. CLIs

Interfaces de linha de comando

2. Fire

Conhecendo o básico da biblioteca

3. Free-Style

Até cansar xD

CLIS

Alguns exemplos

O que são CLIs



Command Line Interfaces, ou interfaces de linha de comando, são programas que oferecem a opção de executar funções pelo terminal.

Um exemplo



```
dunossauro@babbage:~/git/live-de-python/stuff
$ ls
apoiase_label.png  logo.jpg          obs_background.png
apoiase.png        name_card.xcf     obs_background.xcf
apoiase.xcf        obs_background_2.png 'qr codes'
carbon.json        obs_background_2.xcf thumb_template.xcf
dunossauro at babbage in ~/git/live-de-python/stuff on master*
$ ls -l
total 64428
-rw-r--r-- 1 dunossauro dunossauro 21101 ago 24 2020 apoiase_label.png
-rw-r--r-- 1 dunossauro dunossauro 43373 ago 24 2020 apoiase.png
-rw-r--r-- 1 dunossauro dunossauro 111918 ago 24 2020 apoiase.xcf
-rw-r--r-- 1 dunossauro dunossauro 662 set 7 18:57 carbon.json
-rw-r--r-- 1 dunossauro dunossauro 31606 ago 24 2020 logo.jpg
-rw-r--r-- 1 dunossauro dunossauro 5256202 ago 24 2020 name_card.xcf
-rw-r--r-- 1 dunossauro dunossauro 35129 ago 24 2020 obs_background_2.png
-rw-r--r-- 1 dunossauro dunossauro 87971 ago 24 2020 obs_background_2.xcf
-rw-r--r-- 1 dunossauro dunossauro 28123 ago 24 2020 obs_background.png
-rw-r--r-- 1 dunossauro dunossauro 58916 ago 24 2020 obs_background.xcf
drwxr-xr-x 2 dunossauro dunossauro 4096 jan 3 19:40 'qr codes'
-rw-r--r-- 1 dunossauro dunossauro 60271846 dez 2 06:36 thumb_template.xcf
dunossauro at babbage in ~/git/live-de-python/stuff on master*
$
```

Fire

Conhecendo a lib

0 básico



Fire é uma biblioteca para criar Interfaces de linha de comando para qualquer tipo de objeto python.

- Qualquer objeto exposto em um módulo, vira invocável
- Facilita o debug

Antes de tudo, pq?



Eu sei que existem ótimas libs disponíveis para CLIs

- Argparse
- Click
- Typer

Então pq conhecer o fire?

Motivação



O fire é extremamente simples e facilita MUITO, não tem nenhuma complicação para usar. Não recomendado para aplicações grandes baseadas unicamente em CLI, mas ela é divertida de usar.

pip install fire



Bora testar, falei de mais





```
1  import fire
2
3
4  def olar(nome='Eduardo'):
5      """Batatinhas fritas."""
6      return f'Olar {nome}'
7
8
9  fire()
```

Usando



```
1 import fire
2
3
4 def olar(nome='Eduardo'):
5     """Batatinhas fritas."""
6     return f'Olar {nome}'
7
8
9 fire()
```

```
dunossauro@babbage:~/git/live_fire
$ python teste.py olar
Olar Eduardo
dunossauro@babbage in ~/git/live_fire
$ python teste.py olar "live de python"
Olar live de python
dunossauro@babbage in ~/git/live_fire
$ python teste.py olar live
Olar live
dunossauro@babbage in ~/git/live_fire
$
```

Explorando objetos



```
1 import fire
2
3 meu_dicionario = {
4     'nome': 'Eduardo',
5     'hobbies':
6         ['fazer lives', 'Andar de skate', 'tocar violão'],
7     'aniversario': {
8         'dia': 6,
9         'mes': 3
10    }
11 }
12
13 fire.Fire()
```

A coisa que acho mais fascinante no fire é a maneira como ele expõe os objetos

Explorando objetos

```
dunossauro@babbage:~/git/live_fire
$ python teste.py meu_dicionario
nome: Eduardo
hobbies: ["fazer lives", "Andar de skate", "tocar violão"]
aniversario: {"dia": 6, "mes": 3}
dunossauro at babbage in ~/git/live_fire
$ python teste.py meu_dicionario hobbies
fazer lives
Andar de skate
tocar violão
dunossauro at babbage in ~/git/live_fire
$ python teste.py meu_dicionario hobbies 0
fazer lives
dunossauro at babbage in ~/git/live_fire
$ python teste.py meu_dicionario hobbies 1
Andar de skate
dunossauro at babbage in ~/git/live_fire
$
```


Explorando objetos

```
dunossauro@babbage:~/git/live_fire
$ python teste.py meu_dicionario
nome: Eduardo
hobbies: ["fazer lives", "Andar de skate", "tocar violão"]
aniversario: {"dia": 6, "mes": 3}

dunossauro@babbage:~/git/live_fire
$ python teste.py meu_dicionario hobbies
fazer lives
Andar de skate
tocar violão

dunossauro@babbage:~/git/live_fire
$ python teste.py meu_dicionario hobbies 0
fazer lives

dunossauro@babbage:~/git/live_fire
$ python teste.py meu_dicionario hobbies 1
Andar de skate

dunossauro@babbage:~/git/live_fire
$
```

Dict

Lista

String

Grupos

Grupos podem ser criados com classes.

```
1 import fire
2
3 def _comando_1():
4     return 'Tchauzes 1'
5
6 class Olares:
7     def comando_1(self):
8         return 'Olares 1'
9
10
11 class Tchauzes:
12     def __init__(self):
13         self.comando_1 = _comando_1
14
15 fire.Fire()
```

Free- style

Criando uma
micro aplicação

Criando linhas de comando



Vamos criar alguns comandos legais?

Separei alguns links:

- <https://lotericas.io/>
- <https://docs.awesomeapi.com.br/>
- <https://meuguia.tv/programacao/>

```
pip install httpx parsel
```



Tá de sacanagem, né?





picpay.me/dunossauro



apoia.se/livedepython



PIX



Ajude o projeto

