



Python<sup>Live de</sup>

Click - Ferramentas de linha de comando

# Obrigado!

# apoia.se/livedepython

```
z4r4tu5tr4 at babbage in ~/git/apoiadores
```

```
$ python apoiadores.py
```

|                   |                   |                   |                    |
|-------------------|-------------------|-------------------|--------------------|
| Amaziles Carvalho | Andre Machado     | Bruno Guizi       | Carlos Augusto     |
| Cleber Santos     | David Reis        | Dayham Soares     | Diego Ubirajara    |
| Edimar Fardim     | Eliabe Silva      | Elias Soares      | Emerson Lara       |
| Fabiano Silos     | Fabiano Teichmann | Fabiano Gomes     | Fernando Furtado   |
| Fábio Serrão      | Gleison Oliveira  | Humberto Rocha    | Jean Vetorello     |
| Johnny Tardin     | Jonatas Oliveira  | Jonatas Simões    | João Lugão         |
| Jucélio Silva     | Júlia Kastrup     | Leon Teixeira     | Magno Malkut       |
| Maria Boladona    | Matheus Francisco | Nilo Pereira      | Pablo Henrique     |
| Paulo Tadei       | Pedro Alves       | Rafael Galleani   | Regis Santos       |
| Renan Moura       | Renato Santos     | Rennan Almeida    | Rodrigo Vaccari    |
| Sérgio Passos     | Thiago Araujo     | Tiago Cordeiro    | Vergil Valverde    |
| Vicente Marcal    | Wander Silva      | Wellington Carlos | Wellington Camargo |
| Welton Souza      | William Oliveira  | Willian Gl        | Yros Aguiar        |

Python Pro

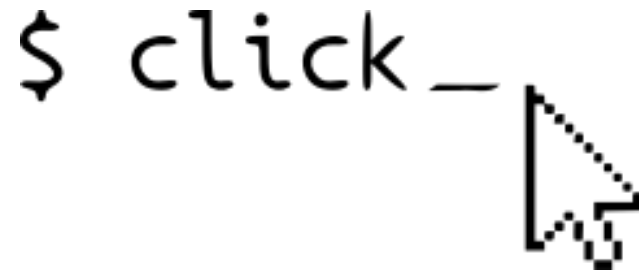
Python para Profissionais

# Roteiro

- CLI
  - Um exemplo
  - Ferramentas de terminal
- Ah.. o Click
  - Um exemplo básico
  - Argumentos
    - types
    - files
  - Opções
    - Estilo
  - Grupos

# Exemplo de CLI

# Ah.. o Click



“Click é um pacote python para criar lindas interfaces de comando (...) com mínimo de código necessário. É altamente configurável, mas vem com padrões sensatos prontos para uso”

<https://click.palletsprojects.com/en/7.x/>

# Exemplo da documentação

```
import click

@click.command()
@click.option('--count', default=1, help='Number of greetings.')
@click.option('--name', prompt='Your name', help='The person to greet.')
def hello(count, name):
    """Simple program that greets NAME for a total of COUNT times."""
    for x in range(count):
        click.echo('Hello %s!' % name)

if __name__ == '__main__':
    hello()
```

# Argumentos

(com pitadas de arquivos)

# @argument(\*args)

Argumentos são opções “fixas” onde o click vai forçar a gente a passar um determinado valor.

```
@click.command()
@click.argument('meu_argumento')
def xpto(meu_argumento):
    """Exemplo de argumento."""
    click.echo(meu_argumento)
```



## @argument(\*args)

Argumentos são opções “fixas” onde o click vai forçar a gente a passar um determinado valor.

```
@click.command()
@click.argument('meu_argumento')
def xpto(meu_argumento):
    """Exemplo de argumento."""
    click.echo(meu_argumento)
```

```
z4r4tu5tr4 at babbage in ~/live88
$ python exemplo_argumento.py 1
1
```

# @argument(\*args)

Argumentos são opções “fixas” onde o click vai forçar a gente a passar um determinado valor.

```
@click.command()
@click.argument('meu_argumento')
def xpto(meu_argumento):
    """Exemplo de argumento."""
    click.echo(meu_argumento)
```

```
z4r4tu5tr4 at babbage in ~/live88
```

```
$ python exemplo_argumento.py
```

```
Usage: exemplo_argumento.py [OPTIONS] MEU_ARGUMENTO
```

```
Try "exemplo_argumento.py --help" for help.
```

```
Error: Missing argument "MEU_ARGUMENTO".
```

# @argument(\*args)

Algumas variações são possíveis a nível de argumentos.

- nargs: Número de argumentos
- envvar: Variável de ambiente
- type: Tipo de dado
  - INT, STRING, FLOAT
  - File
  - Path
  - Choice
  - Ranges

## @argument(\*args)

```
@click.command()
@click.argument('argumento', envvar='BATATAS', type=click.STRING)
@click.argument('outro', nargs=-1)
def xpto(argumento, outro):
    """
    Exemplo de argumento.

    envvar: Vai usar a variável de ambiente BATATAS
            Caso não seja passada nada
    type: string
    nargs: Número infinito de argumentos
    """
    click.echo(argumento)
    click.echo(outro)
```

# Desafio 1

Implemente um programinha igual ao cat. Ele tem que ler um arquivo e imprimir o mesmo na tela

# Opções

(com pitadas de estilo)

# @option(\*args)

O decorador de option permite que você use parâmetros opcionais. Ou seja, ele expande os comando a mais do que argumentos

```
@click.command()
@click.argument('file', type=click.File('r'))
@click.option('-n', type=click.BOOL, default=False)
def cat(file, n):
    """Similar ao cat."""
    for number, linha in enumerate(file.readlines()):
        if n:
            linha = f'{number} {linha}'
        click.echo(linha)
```

## Desafio 2

Implemente um programinha igual ao grep. Ele tem que ler um arquivo e imprimir o mesmo na tela, se der match em alguma linha ele deve printar colorido a linha



# @option(\*args)

O legal das opções é que elas podem ter muitas variações

- podem ter mais de um nome
- default: Diz um valor default
- count: Conta quantas vezes o parâmetro aparece
- type: Também aceitam tipos
- prompt: Aceitam entradas do usuário

# Grupos

(com pitadas de facilidade)