O tabu dos executáveis

Live de Python # 173

Roteiro

1. O tabu dos executáveis
Por que eu deveria transformar meu código em executável?

2. Executável VS Interpretador
Como o interpretador funciona?

3. Compilação
Sim, dá pra compilar python

4. Os executáveis
Como distribuir meu código?

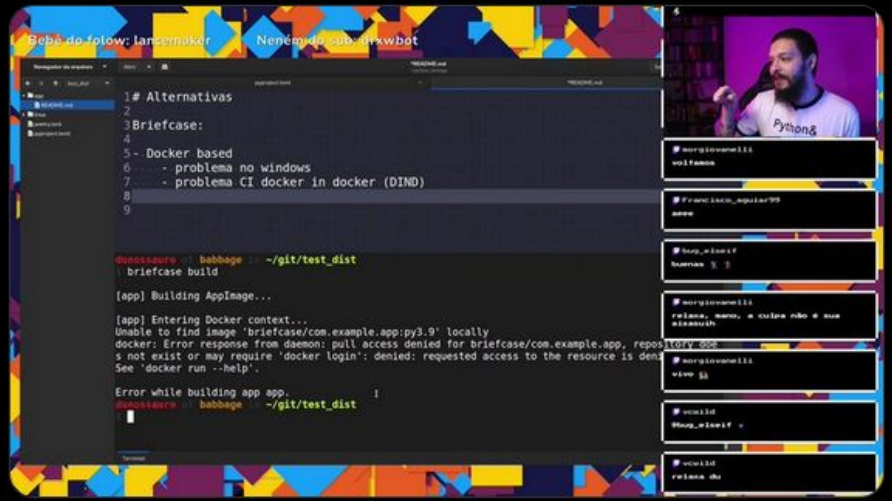
LiveDivulgador
@LiveDivulgador

➡️ @dunossauro está em Live neste momento!🔴

[Python] Estudando distribuição de binários

Entra aí: twitch.tv/dunossauro

#Twitch #live



12:22 AM · 26 de fev de 2021 · erased18375832

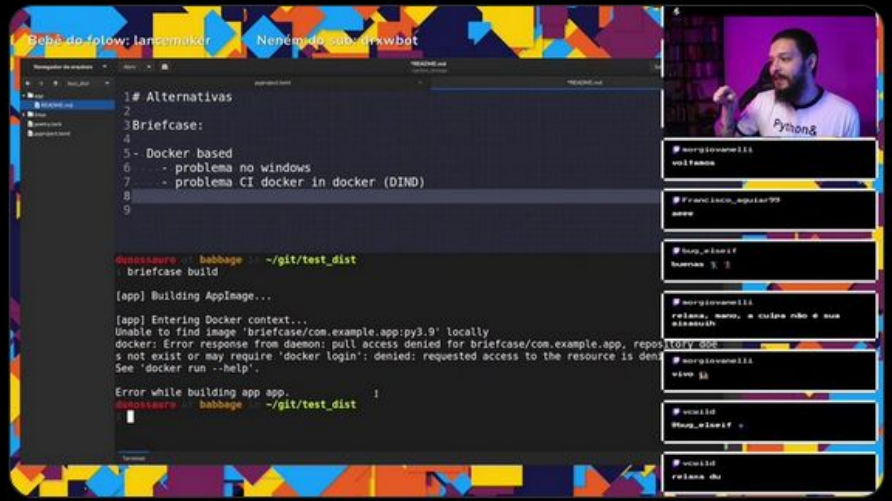
LiveDivulgador
@LiveDivulgador

➡️ @dunossauro está em Live neste momento!🔴

[Python] Estudando distribuição de binários

Entra aí: twitch.tv/dunossauro

#Twitch #live



12:22 AM · 26 de fev de 2021 · erased18375832

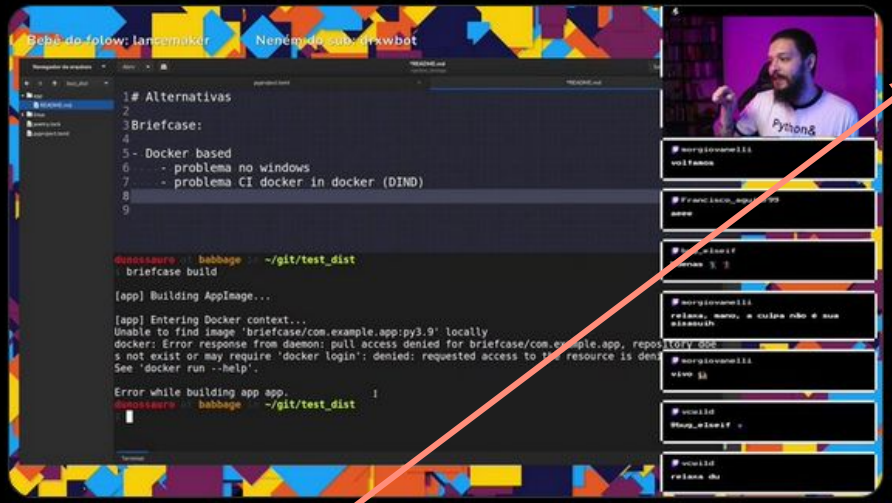
LiveDivulgador
@LiveDivulgador

@dunossauro está em Live neste momento!

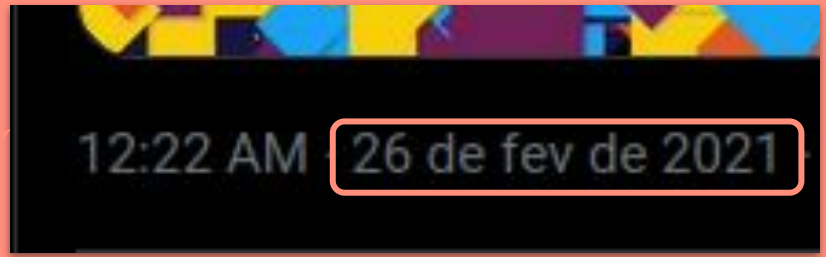
[Python] Estudando distribuição de binários

Entra aí: twitch.tv/dunossauro

#Twitch #live



12:22 AM · 26 de fev de 2021 erased18375832



5 meses depois...

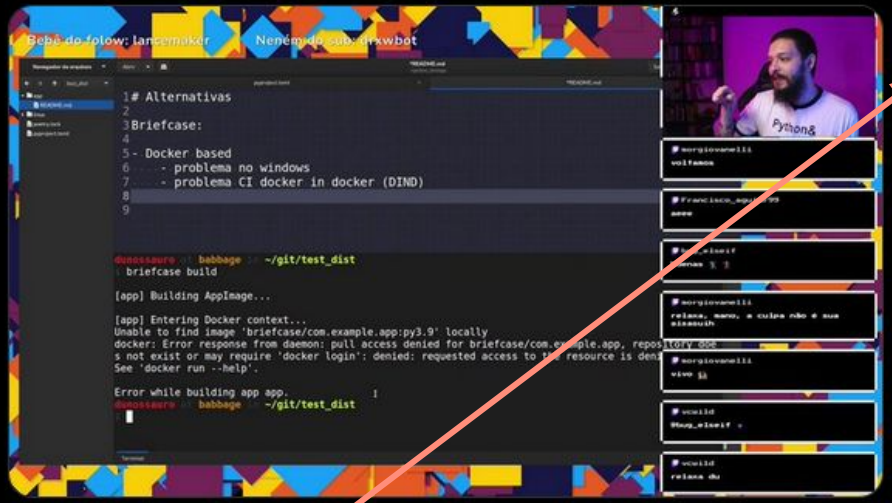
LiveDivulgador
@LiveDivulgador

@dunossauro está em Live neste momento!

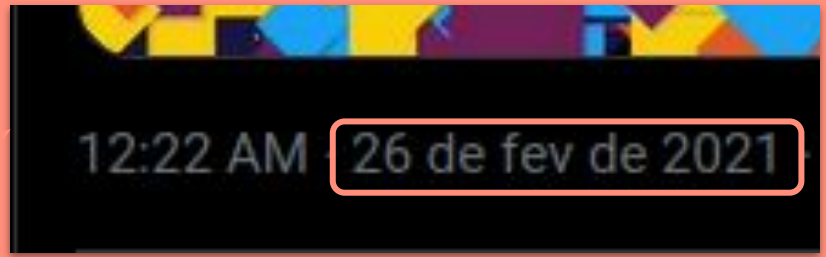
[Python] Estudando distribuição de binários

Entra aí: twitch.tv/dunossauro

#Twitch #live



12:22 AM · 26 de fev de 2021 erased18375832



5 meses depois...
Ainda não tenho uma resposta

picpay.me/dunossauro

apoia.se/livedepython

PIX

Ajude o projeto

Ademar Peixoto, Alex Lima, Alex Lopes, Alexandre Harano, Alexandre Santos, Alexandre Tsuno, Alexandre Villares, Alynne Ferreira, Alysson Oliveira, Amaziles Carvalho, André Rocha, Arnaldo Turque, Bruno Batista, Bruno Oliveira, Caio Nascimento, Carlos Chiarelli, Cleber Santos, César Almeida, Davi Ramos, David Kwast, Diego Guimarães, Dilenon Delfino, Elias Soares, Eugenio Mazzini, Everton Alves, Fabiano Gomes, Fabio Barros, Fabio Castro, Fabrícia Diniz, Fabrício Coelho, Flavkaze, Francisco Alencar, Fábio Serrão, Gabriel Simonetto, Gabriel Soares, Gabriela Santiago, Geandreson Costa, Guilherme Castro, Guilherme Felitti, Guilherme Ostrock, Gustavo Chacon, Henrique Machado, Israel Fabiano, Italo Silva, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jorge Plautz, Jose Mazolini, José Prado, João Lugão, João Schiavon, Juan Gutierrez, Julio Silva, Jônatas Silva, Júlia Kastrup, Kaneson Alves, Leonardo Cruz, Leonardo Galani, Leonardo Mello, Lidiane Monteiro, Lorena Ribeiro, Lucas Barros, Lucas Mello, Lucas Mendes, Lucas Teixeira, Lucas Valino, Luciano Ratamero, Maiquel Leonel, Marcela Campos, Marcelo Rodrigues, Maria Clara, Marina Passos, Matheus Vian, Melissa Mendonça, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Patric Lacouth, Patricia Minamizawa, Patrick Gomes, Paulo Tadei, Pedro Pereira, Peterson Santos, Rafael Lino, Reinaldo Silva, Renan Moura, Revton Silva, Rodrigo Ferreira, Rodrigo Mende, Rodrigo Vaccari, Ronaldo Silva, Sandro Mio, Silvio Xm, Thiago Araujo, Thiago Borges, Thiago Bueno, Tyrone Damasceno, Victor Geraldo, Vinícius Bastos, Vinícius Ferreira, Vítor Gomes, Wendel Rios, Wesley Mendes, Willian Lopes, Willian Rosa, Wilson Duarte, Érico Andrei

Obrigado você

Por que esse assunto é polêmico?

O taboo

Como gerar executáveis com Python?

A primeira pesquisa!

Sem tempo pra perguntas ...

Se objetivo é gerar o executável ...

```
1  from tkinter import Tk, Label
2
3  root = Tk()
4
5  Label(
6    text='Live de Python',
7    font=('Arial', 25)
8  ).pack()
9
10 root.mainloop()
```

```
1  pip install pyinstaller
2  pyinstaller --onefile exemplo_tk.py
```

```
1 @@@@@@@@@@   @@@@@@@@@@
```

Por que eu quero gerar Executáveis?

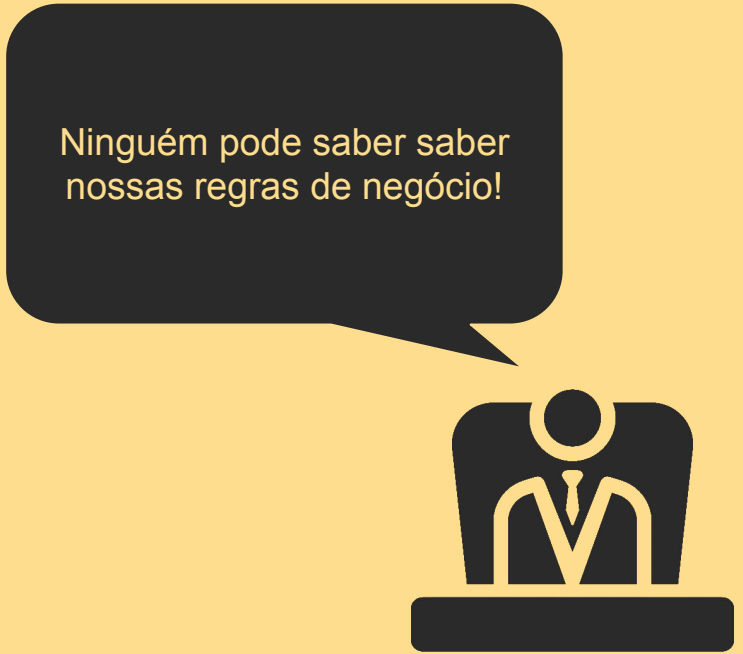
A pergunta deveria ser essa!

Motivos pelos quais você pode querer

- Ofuscação de código
- Velocidade
- Compilados
- Distribuição
  - Plataformas

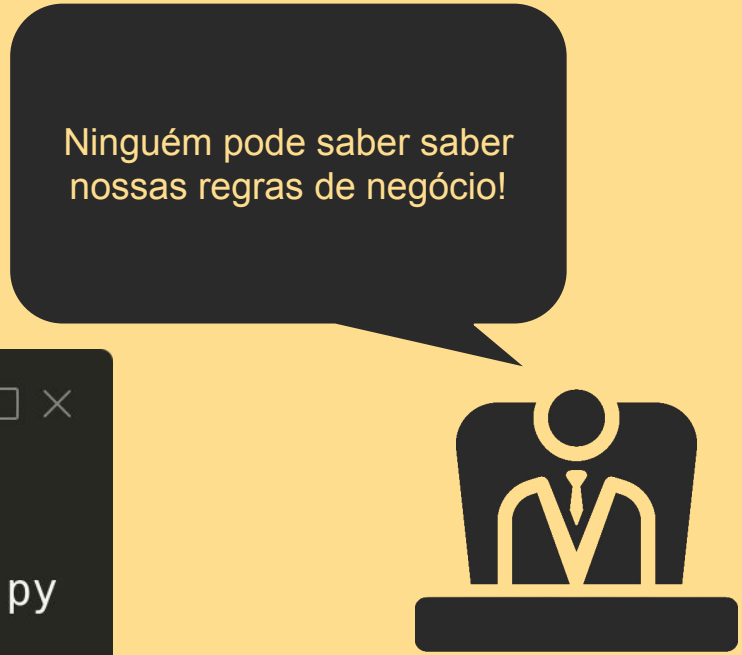
Motivos pelos quais você pode querer

- Ofuscação de código



Motivos pelos quais você pode querer

- Ofuscação de código




```
1   pip install pyarmor
2   pyarmor obfuscate exemplo_tk.py
```

Como fica o código ofuscado?

```python
from tkinter import Tk, Label

root = Tk()

Label(
  text='Live de Python',
  font=('Arial', 25)
).pack()

root.mainloop()
```

```python
from pytransform import pyarmor_runtime
pyarmor_runtime()
__pyarmor__(__name__, __file__, b'\x50\x59\x41\x52\x4d\x4f\x52\x00\x00\x03\x09\x00\x61\x0d\x0d\x0a\x07\x2c\xa0
\x01\x00\x00\x00\x00\x01\x00\x00\x00\x40\x00\x00\x00\x4c\x01\x00\x00\x00\x00\x00\x18\x1f\xe5\xec\xd0\x58\x5e\x7e
\x76\x3e\xe5\x6b\x7b\x64\xcc\xa9\x3e\x00\x00\x00\x00\x00\x00\x00\x00\x69\x1e\x26\x17\x73\x1d\xae\x32\xb0\x3e\x0c
\x21\x49\x67\xc9\x5d\xfc\x17\xa2\x26\x12\xeb\x33\x6f\x3c\x1b\xd9\x4a\x65\x3e\xb5\x38\xcc\x1b\x5e\x42\x41\xb9\x3a
\x3a\xaa\x20\x3d\xaf\xab\x8c\xf0\xaf\x95\x96\xae\xaa\xc3\x0f\x98\x32\x10\x44\xcb\xee\xf8\xb0\xeb\x60\xf1\x7f\x35
\x41\xc0\xd1\xce\x4a\xc1\x5b\x9c\xc4\x0e\x55\x9e\xe5\x49\x38\xeb\x42\x36\x64\x00\xfa\xc5\x4e\x3a\xb9\x74\x34\xd7
\x6e\x2e\x73\x21\xf4\xa5\x0d\x68\xab\x19\xe8\x56\xb6\x53\xcb\x3d\x5c\x05\x4e\x9d\x19\x8a\xa2\x1d\xea\xe1\x7a\x73
\x0e\x2d\x36\x18\x7d\x2b\x5e\x10\xdb\x9b\x8f\x50\xe2\xfd\x0e\x09\x2b\x21\x15\x99\xb4\xe6\x1d\x4f\x0a\xcc\xea\x3d
\x3e\x3b\x0c\x18\xa8\x7c\xbb\x9a\x3a\x01\x43\xab\xca\xc0\x55\x84\x8c\x95\xec\xa3\x7a\xef\xd6\x0f\x76\xcd\x61\x90
\xb5\xbd\xdc\x38\x42\x5a\x5c\xfc\x0f\x9d\xde\x3c\x14\x3b\x7b\x6c\x8a\xfa\x34\xd9\xea\xcf\x0b\x81\x99\x29\x09\x1e
\xa8\xbd\x2f\xc1\x70\x08\x45\x1d\xfc\xad\x94\x0a\x66\x69\x17\xda\x25\x72\xfb\x2f\x95\xf1\x37\x6b\x33\x4a\xcb\xd1
\x9b\x91\x88\x65\x2c\x97\x59\x62\xf9\x1d\x4d\x81\x95\xaf\x2e\x03\xfd\x22\xd5\xfc\x6f\x26\x8c\x96\x58\x7d\x26\x8b
\x2b\xfc\x6e\xfd\x31\xce\x69\x8d\xd8\xf8\x64\x5f\xd4\xce\xd0\x86\x91\xed\x9f\x42\xf6\x1c\x2a\x34\xba\xfd\x20\x0f
\xb7\xee\x00\xea\x0d\xe3\xa2\xd9\xdc\x24\x54\xbe\xd3\x6a\x90\xd0\x58\xed\xe3\xcc\x6b\xaa\x60\xc7\xc9\xfe\x00\x5f
\xb4\x40\x77\x4b\x01\x42\x14\xb0\x37\xe2\x4a\x70\xf2', 2)
```

"Vantagens" de ofuscar

- Não dá pra ler
- AINDA não existe engenharia reversa
- O pyarmor não precisa estar instalado no cliente

Outras vantagens

- O código pode der data de expiração
- Bloqueio de máquinas para execução (MAC, Serial do HD, ...)

"Vantagens" de ofuscar

- Não dá pra ler
- AINDA não existe engen
- O pyarmor não precis

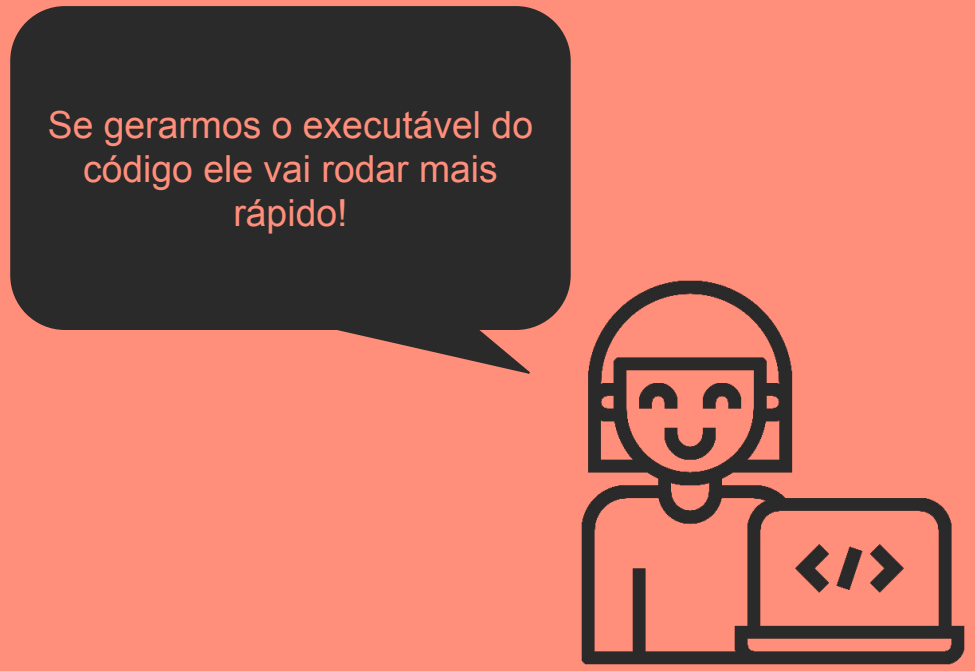
Outras vantagens

- O código pode de
- Bloqueio de máqui

Querem uma live sobre pyarmor?

Motivos pelos quais você pode querer

- Ofuscação de código
- Velocidade



Motivos pelos quais você pode querer

- Ofuscação de código
- Velocidade

JIT
(Just-In-Time)

Ferramentas de JIT

PyPy

Numba

A ideia do jit é compilar códigos "quentes"

Imagine em um bloco de código "caro" em tempo de execução. A partir do momento em que ele precisa ser executado diversas vezes, essa parte "cara" do código é compilada.

Exemplo mais simples de JIT - Numba

```python
def loop_caro(x):
    for x in range(x):
        if x ** 2:
            if x ** 3:
                x + 3
```

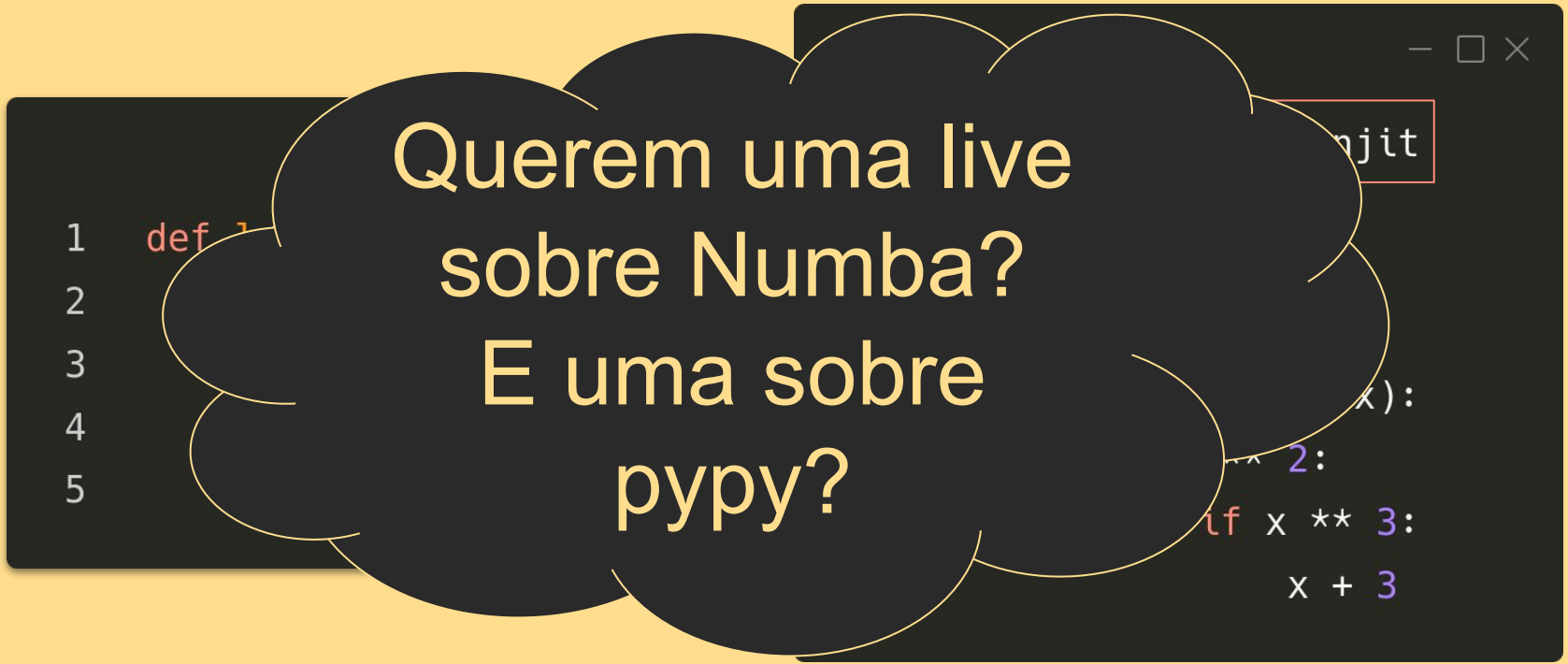
Exemplo mais simples de JIT - Numba

```python
def loop_caro(x):
    for x in range(x):
        if x ** 2:
            if x ** 3:
                x + 3
```

```python
from numba import njit

@njit
def loop_caro(x):
    for x in range(x):
        if x ** 2:
            if x ** 3:
                x + 3
```

Exemplo mais simples de JIT - Numba



Executável VS Interpretador

Como o interpretador funciona?

Arquivos executáveis

Existem 2 tipos de arquivos que precisamos conhecer. Os arquivos de texto e os arquivos binários executáveis.

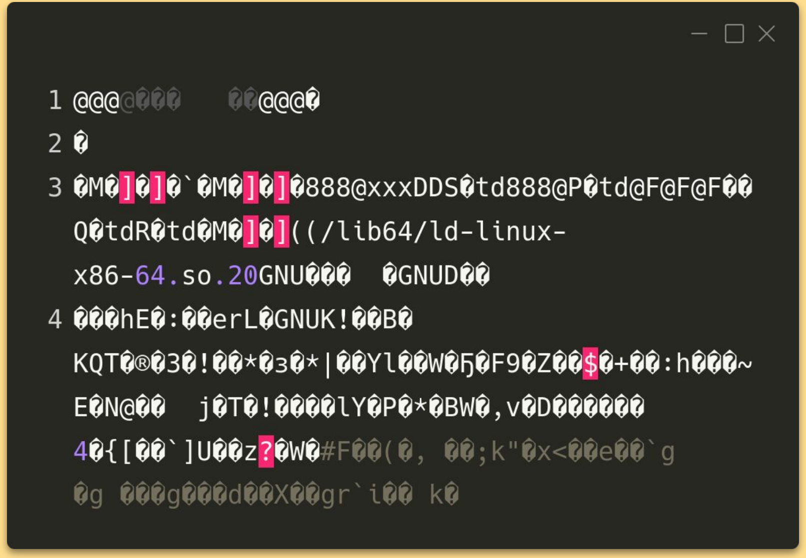
```python
from tkinter import Tk, Label

root = Tk()

Label(
  text='Live de Python',
  font=('Arial', 25)
).pack()

root.mainloop()
```

fonte -> meu_arquivo.py



binário -> meu_arquivo

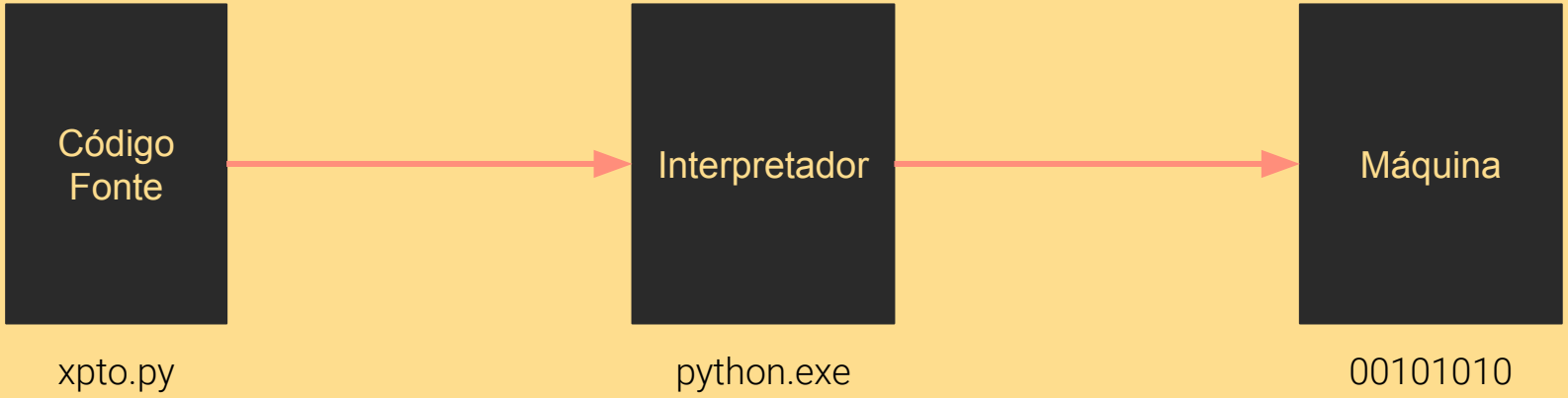
Como funciona um executável?

Um arquivo executável contém as instruções necessárias para dizer a máquina o que fazer.

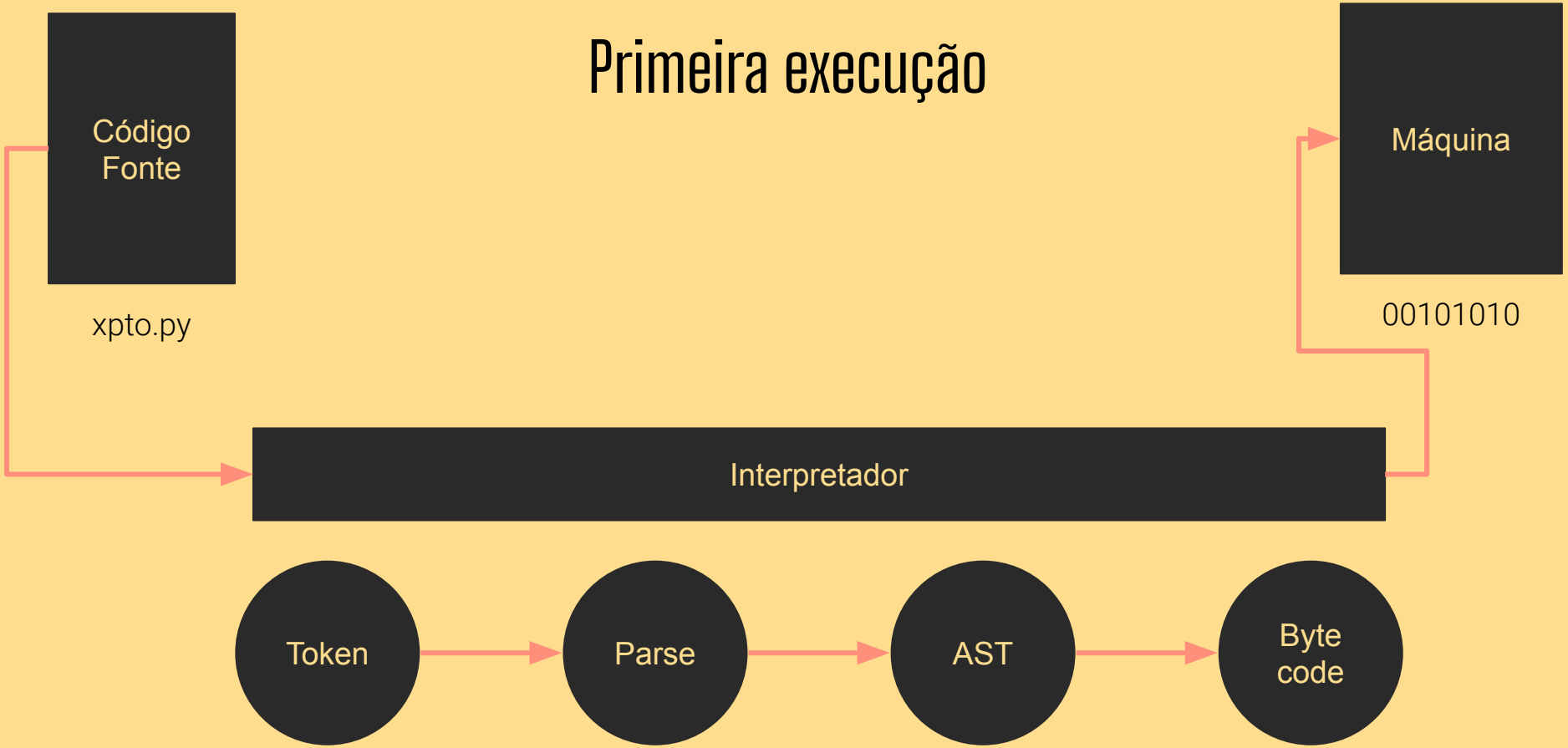
Embora ele não possa ser lido por pessoas, ele contém todas as instruções de máquina, que fazem o fluxo ser executado.

Como o código python é executado?

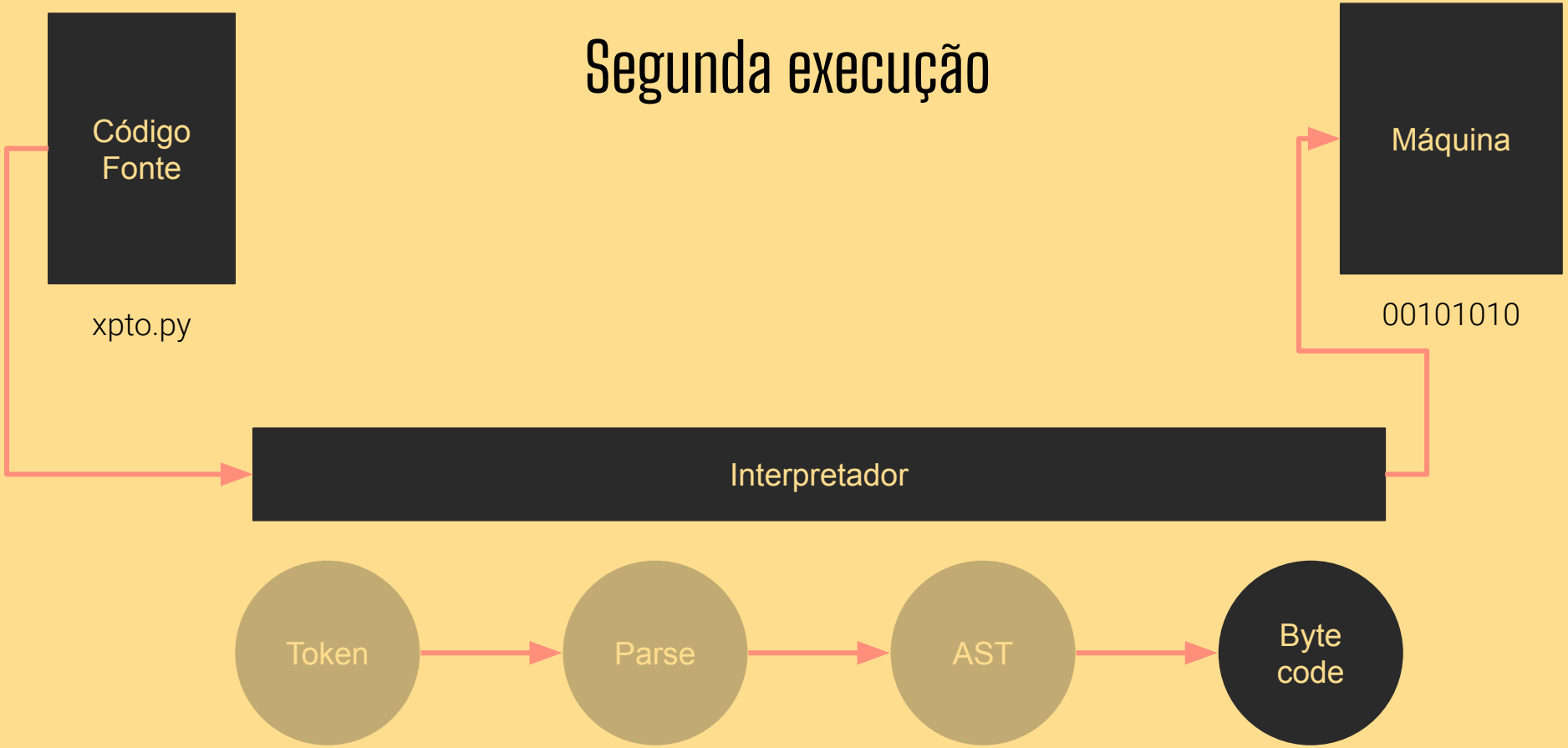
Bom, sabemos que python é uma linguagem interpretada. Mas o que isso quer dizer?



Primeira execução



Segunda execução



Vamos ver essa representação?

A tokenização

python -m tokenize code.py

Vamos ver essa representação?

Parse + ASTS

python -m ast code.py

Vamos ver essa representação?

Bytecode

pythom -m py_compile code.py

Sim, podemos compilar python

Compilação

Motivos pelos quais você pode querer

- Ofuscação de código
- Velocidade
- **Compilados**
- Distribuição
  - Plataformas

Motivos pelos quais você pode querer

- Ofuscação de código
- Velocidade
- **Compilados**
- Distribuição
  - Plataformas

Nuitka

Vamos gerar um executável?

Para a nossa primeira experiência, vamos transcrever nosso código para linguagem C usando **Cython**.

```
pip install cython
cython meu_script.py --embed
gcc -Os -I /usr/include/python3.9/ meu_script.c -lpython3.9 -o meu_script
```

Vamos gerar um executável?

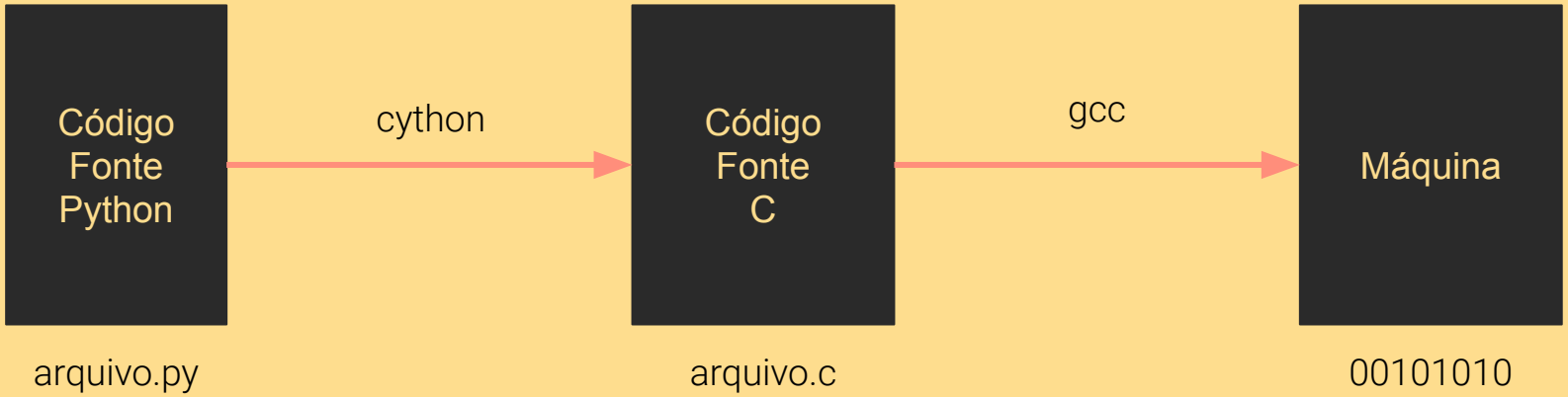
Para a nossa primeira experiência, vamos transcrever nosso código para linguagem C usando **Cython**.

```
1   pip install cython
2   cython meu_script.py --embed
3   gcc -Os -I /usr/include/python3.9/ meu_script.c -lpython3.9 -o meu_script
```

Os: Otimização
I: Include
l: Link

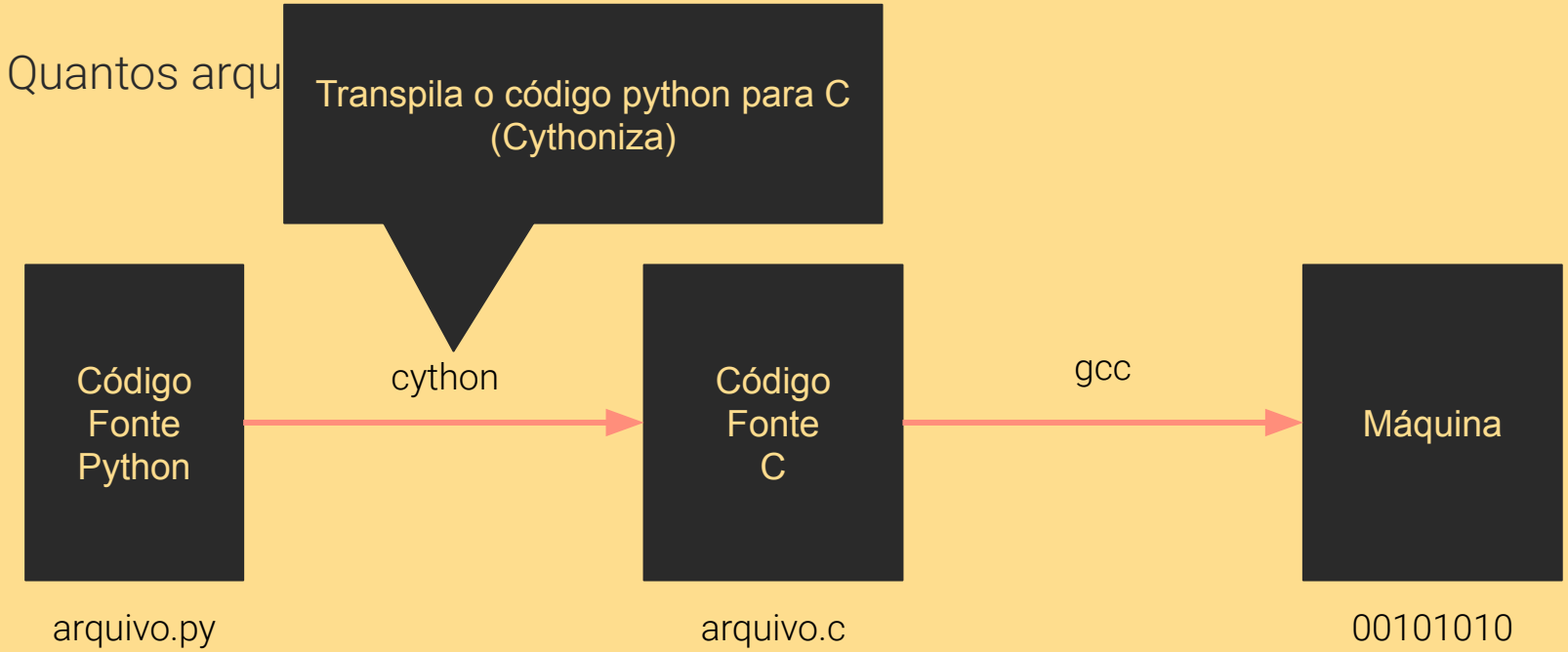
Agora vamos entender

Quantos arquivos e quantas coisas...



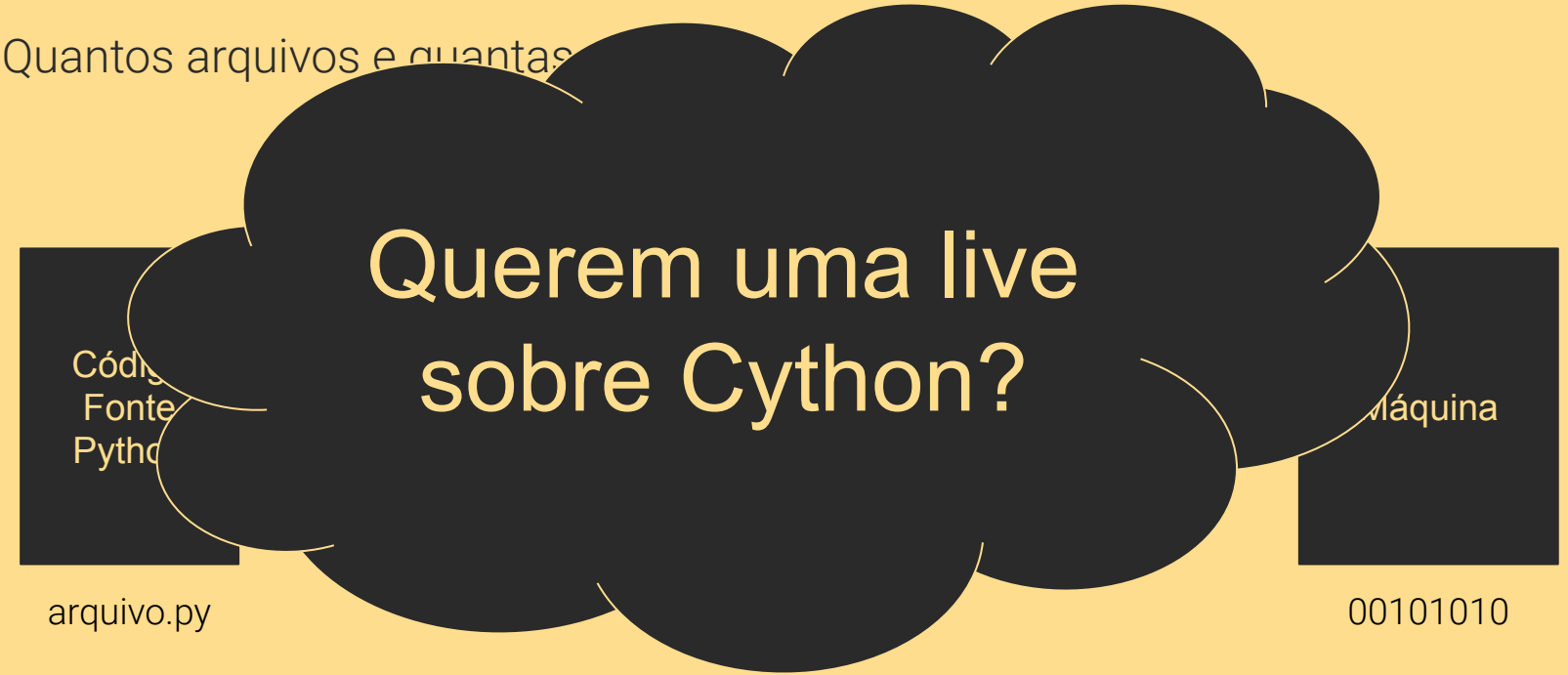
Agora vamos entender

Quantos arqu



Agora vamos entender

Quantos arquivos e quantas



Executáveis

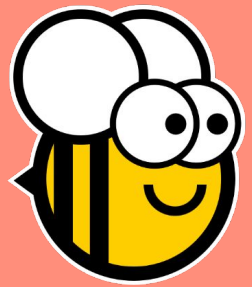
Distribuindo seu código

Motivos pelos quais você pode querer

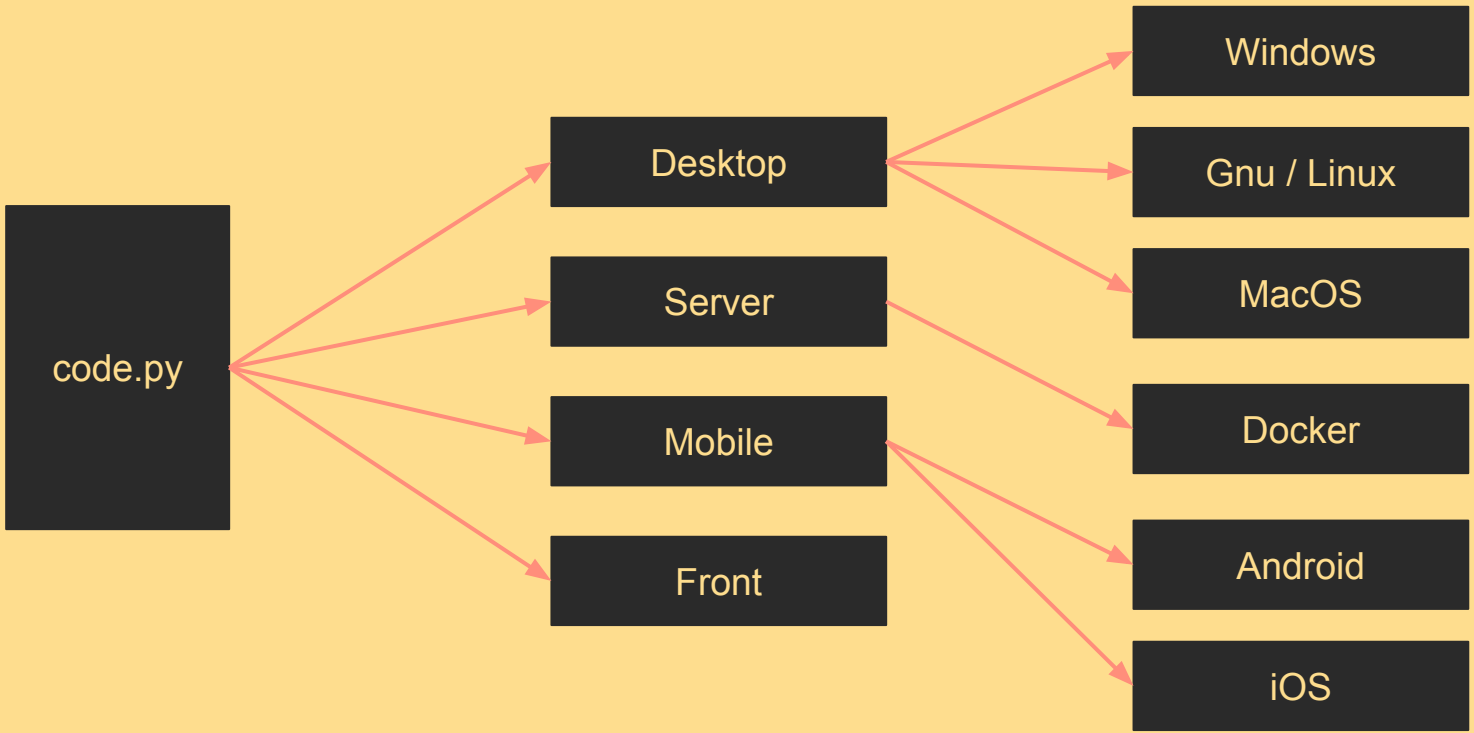
- Ofuscação de código
- Velocidade
- Compilados
- **Distribuição**
  - Plataformas

Motivos pelos quais você pode querer

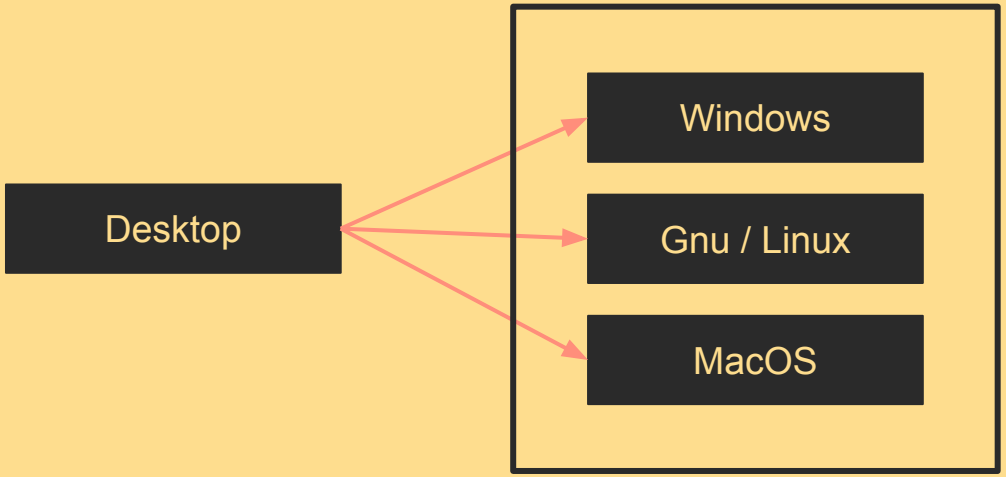
- Ofuscação de código
- Velocidade
- Compilados
- **Distribuição**
  - Plataformas



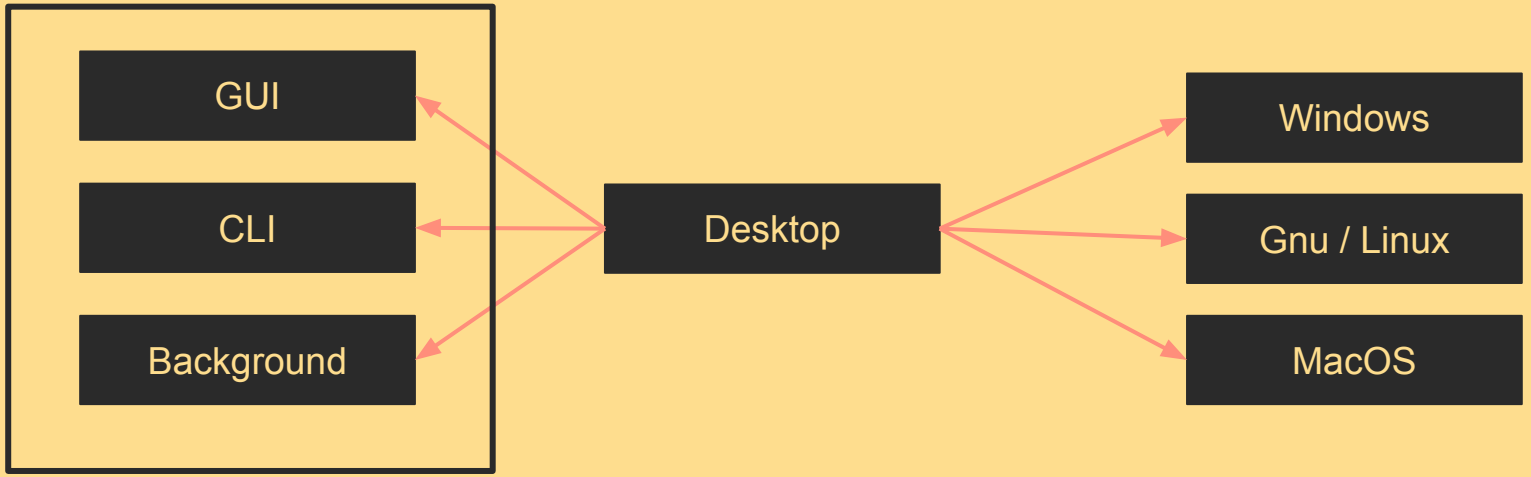
Plataforma?


code.py
Desktop
Server
Mobile
Front
Windows
Gnu / Linux
MacOS
Docker
Android
iOS

Plataforma?

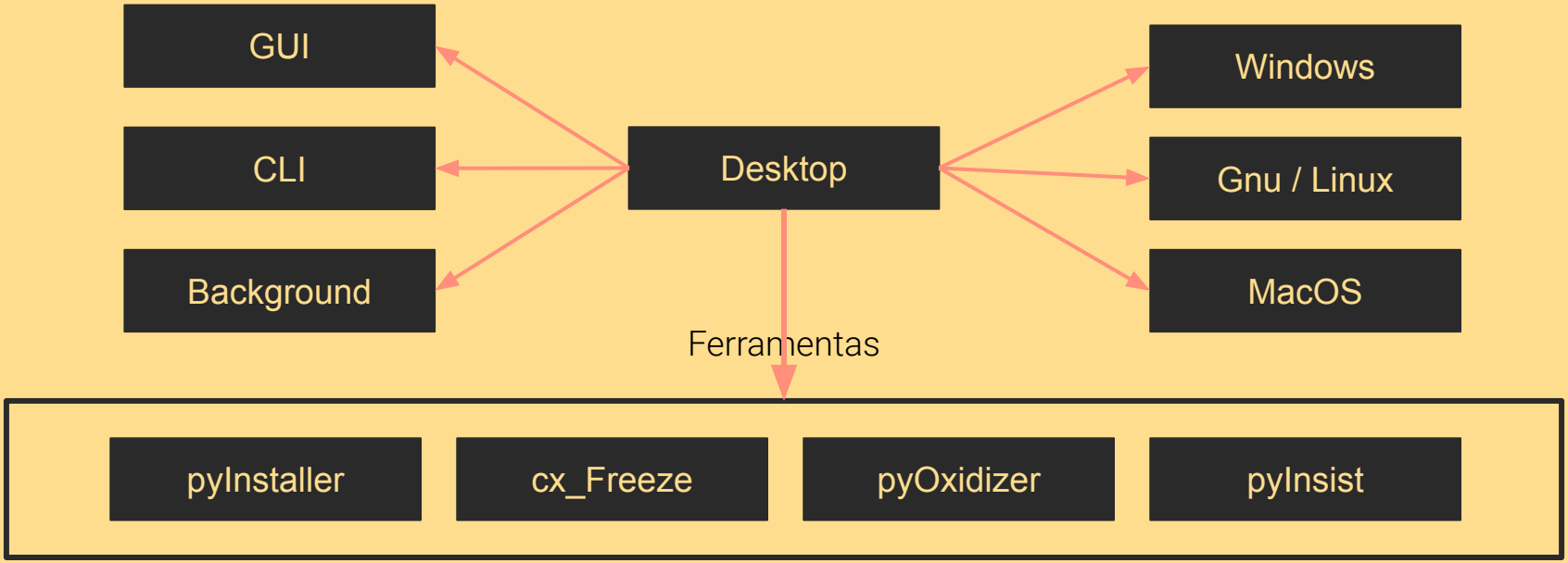


Plataforma?



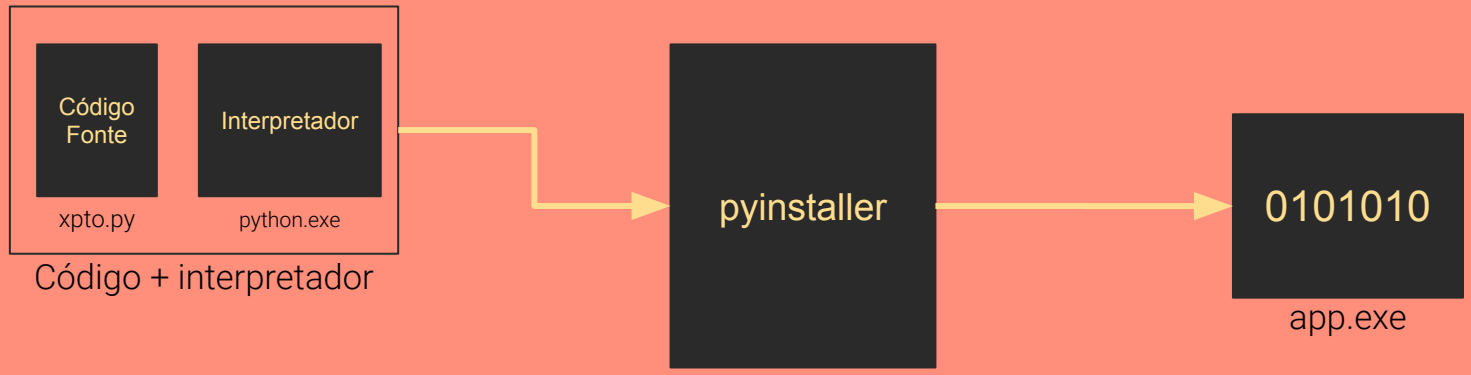
Formato

Plataforma?


GUI
CLI
Background
Desktop
Windows
Gnu / Linux
MacOS
Ferramentas
pyInstaller
cx_Freeze
pyOxidizer
pyInsist

Como funcionam os executáveis gerados?

Existem diversas ferramentas para isso. A mais tradicional talvez seja o **pyinstaller**



Qual ferramenta escolher?

DEPENDE!!!

Comparação simples

Vamos avaliar o quadro geral das ferramentas:

- **pyInstaller**
  - O estado da arte
  - multiplataforma
  - build spec
  - troca de bootloaders
  - reconhecimento de bibliotecas complexas (QT, GTK, ...)

* Muitos falsos positivos

Comparação simples

Vamos avaliar o quadro geral das ferramentas:

- pyInstaller
- **cx_Freeze**
  - Usa o setup.py
  - Multiplataforma
  - Não desempacota em memória

* Empacota a python.dll, o que trás muitos problemas no linux

Comparação simples

Vamos avaliar o quadro geral das ferramentas:

- pyInstaller
- cx_Freeze
- **pyOxidizer**
  - Ferramenta nova, Rust
  - multiplataforma
  - bezel scripts
  - Não desempacota em memória

* Só suporta python 3.8+; bezel scripts

Comparação simples

Vamos avaliar o quadro geral das ferramentas:

- pyInstaller
- cx_Freeze
- pyOxidizer
- **pyInsist**
  - **Gera instaladores** (SNIS)
  - Somente para windows

* Só funciona no windows, problemas com empacotamento

picpay.me/dunossauro

apoia.se/livedepython

PIX

Ajude o projeto