

Automação web com Python

Uma introdução



Antes de tudo





Nome:

Eduardo Mendes

Instituição:

Unicamp / Diebold Nixdorf

Contatos:

{facebook, github, gist
instagram, linkedin,
telegram, twitter}/dunossauro



Eduardo Mendes

1.969 inscritos

PERSONALIZAR O CANAL

ESTÚDIO DE CRIAÇÃO

INÍCIO

VÍDEOS

PLAYLISTS

CANAIS

DISCUSSÃO

SOBRE



Envios

REPRODUZIR TODOS



Live de Python #48 - Criando logs para aplicações

271 visualizações •

Transmitido 1 dia atrás



Live de Python #47 - Esses dados são viciados?

326 visualizações •

Transmitido 1 semana atrás



Live de Python #46 - O que aprendi ensinando Python

435 visualizações •

Transmitido 1 semana atrás



Live de Python #45 - Rows (Com Turicas)

570 visualizações •

Transmitido 2 semanas atrás

<https://github.com/dunossauro/qaninja-liveclass>

<https://bit.ly/2HMZTCN>

Algumas palavrinhas sobre Python





História do Python

Python foi criado em **1982** por Guido Van Rossum, carinhosamente chamado somente de **Guido** enquanto ele trabalhava no centro de matemática e ciência da computação em Amsterdã. Nesse mesmo centro foram criadas diversas linguagens como o ABC (Pai do Python) e o Algol68.



História do Python

Em 2001 foi criada a PSF e todas as releases lançadas desde então não sofrem alterações de licença.

Em 2001 também foram criadas as PEPs (Python Enhancement Proposals)

<https://www.python.org/dev/peps/pep-0020/>

<https://github.com/python/>

Python e ferramentas populares

- Web (django, flask ...)
- Cientistas de dados (pandas, numpy, matplotlib, tensorflow ...)
- Scraping (requests, selenium, scrapy ...)
- QA (**behave**, radish, **selenium**, specs, mamba, **pageobjects** ...)

Python na mídia

OPERAÇÃO

SERENATA
DE AMOR

INTELIGÊNCIA ARTIFICIAL
PARA CONTROLE SOCIAL DA
ADMINISTRAÇÃO PÚBLICA

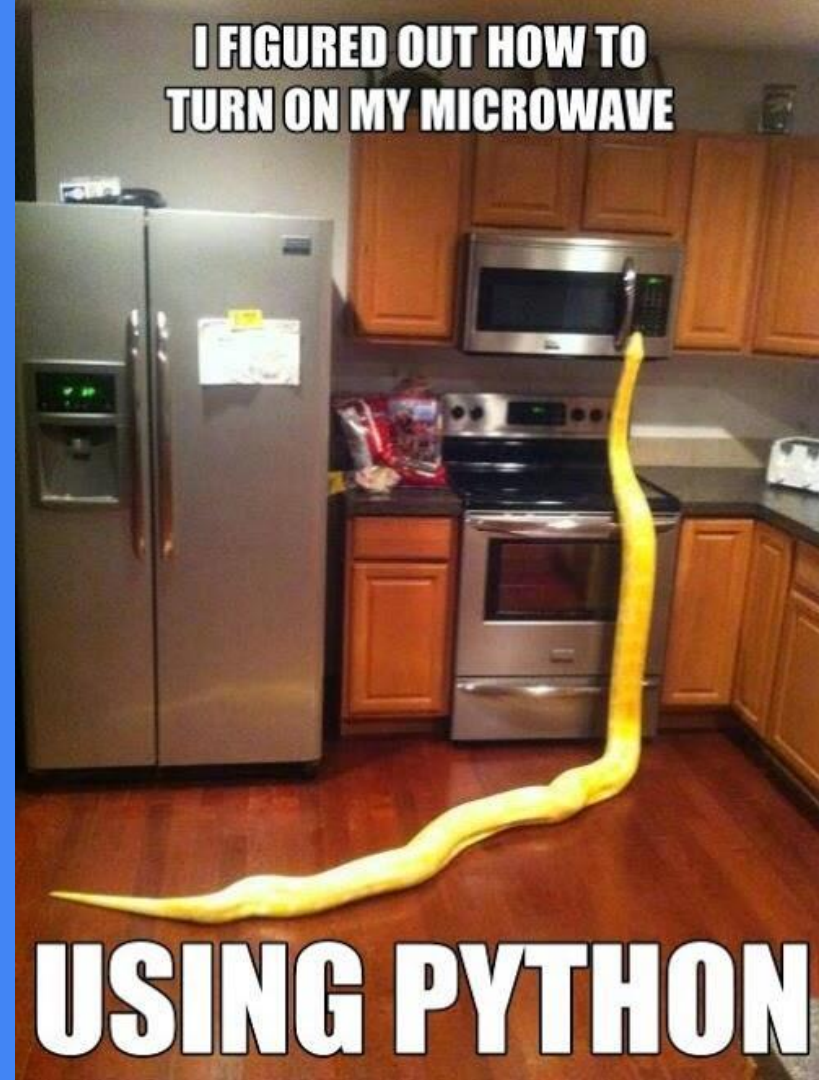


Comunidade



Python

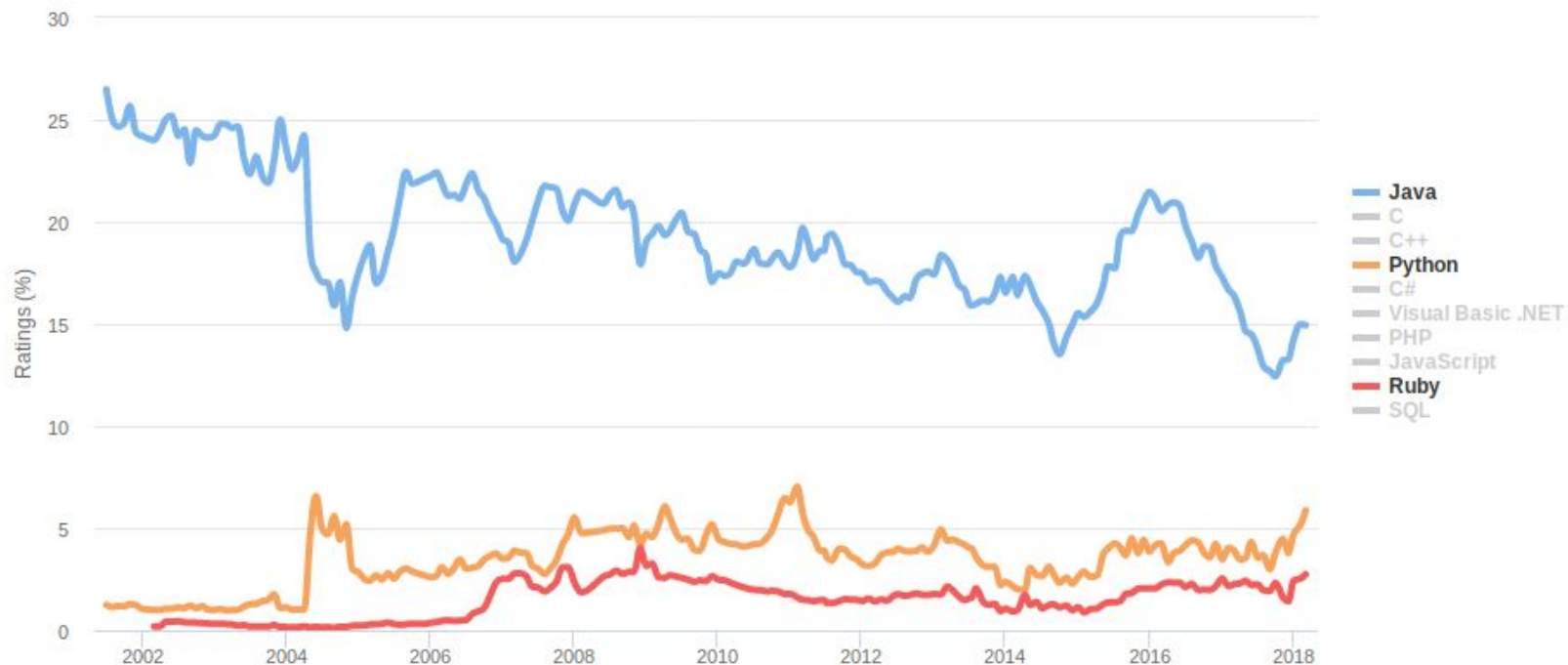
- Multiplataforma
- Multiparadigma
- Mantida pela comunidade
- Com uma licença não proibitiva



Por que falar de Python?

TIOBE Programming Community Index

Source: www.tiobe.com



Behave (Gherkin Files)



Behave é um framework de BDD baseado em arquivos de features e segue os padrões do cucumber e é ligeiramente parecido com ele.

Gherkin

- “É uma linguagem de domínio específico de negócios criada especialmente para descrições de comportamento”
- Tem como propósito servir de documentação e roteiro para testes automatizados

Gherkin example

Cenário: Transferir dinheiro para uma conta no exterior

Dado que um cliente possui uma conta corrente

Quando o cliente transfere fundos desta conta para uma conta no exterior

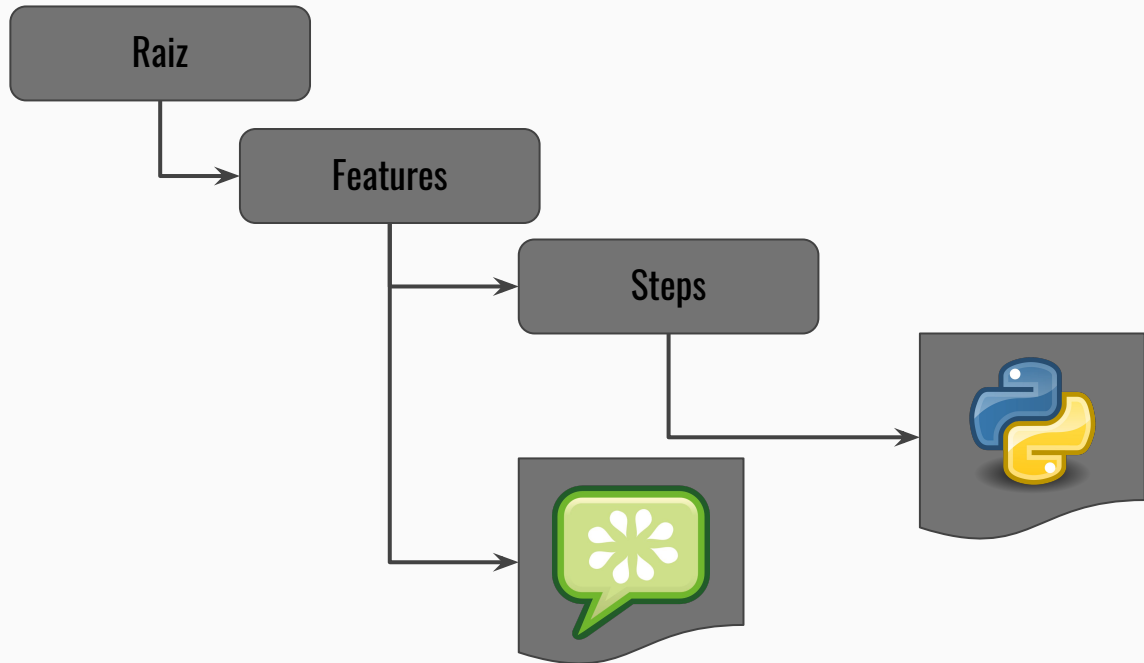
Então os fundos devem ser depositados na conta no exterior

E a taxa de transação deve ser deduzida da conta corrente

Código exemplo

```
@given('que um cliente possui uma conta corrente')  
def check_if_user_has_account(context):  
    pass  
  
@then('os fundos devem ser depositados na conta no exterior')  
def transfer_money_to_other_country(context):  
    pass
```

Estrutura de diretórios do behave







Dúvidas?

Nome:

Eduardo Mendes

Instituição:

Unicamp / Diebold Nixdorf

Contatos:

{facebook, github, gist
instagram, linkedin,
telegram, twitter}/dunossauro