

Python



Otávio Vieira, Bruno Isaac e Enzo Rodrigues



O que é Python?

O Python é uma linguagem de programação interpretada, interativa e orientada a objetos. Há suporte a vários paradigmas de programação além da programação orientada a objetos, tal como programação procedural e funcional.

Criado por Guido van Rossum. Guido nasceu em 31 de Janeiro de 1956. Realizou seu mestrado em Matemática e Ciência da Computação na Universidade de Amsterdam em 1982. Foi medalhista de Bronze na olimpíada internacional de matemática.

Origem e Motivação

Necessidade de uma Nova Linguagem

O Python foi desenvolvido devido à necessidade de uma linguagem interpretada, extensível e fácil de usar, combinando elementos positivos das linguagens ABC e Modula-3. Surgiu especificamente para facilitar a administração do sistema operacional distribuído Amoeba, superando limitações das linguagens existentes.

Origem do Nome

Na década de 70, Van Rossum era um grande fã de um programa de televisão chamado Fly Circus de Monty Python, da BBC, ou apenas Monty Python para os íntimos.

Assim, quando desenvolveu a linguagem, ele pensou que precisava de um nome curto, único e um pouco misterioso, e por algum motivo que só ele conhecia, decidiu chamar o projeto de "Python".

Primeiros Anos (1989-1994)

1989

A linguagem de programação python foi concebida na década de 1980 e começou a ser implementada em Dezembro de 1989 por Guido van Rossum no CWI (Centrum Wiskunde & Informatica), como um sucessor da linguagem de programação ABC. Guido continua sendo o autor principal do Python.

Impacto: Introdução de uma linguagem de programação com sintaxe clara e legível, facilitando a interação entre programadores e máquinas.

1994

Em Janeiro de 1994, a versão 1.0 do Python foi publicada. A nova versão contava com features de ferramentas de programação funcional como lambda, map, filter e reduce.

Impacto: Aumento da adoção do Python em projetos comerciais e acadêmicos, consolidando sua posição no mercado.

1

2

3

1991

Em fevereiro de 1991, Van Rossum publicou a primeira versão do Python, sendo a versão 0.9.0 em que já era presente classes com herança, tratamento de erros, funções, tipos de dados como listas, dicionários, strings e muitos mais recursos.

Ela foi liberada como um sistema modular emprestado da linguagem de programação Modula-3.

Impacto: Estabelecimento de uma base sólida para a comunidade de desenvolvedores, promovendo a colaboração e o crescimento da linguagem.

Crescimento e Reconhecimento (2000-2002)

2000: Python 2.0 e PythonLabs

A equipe principal do Python foi para a BeOpen.com, criando o BeOpen PythonLabs. O Python 2.0 foi o único lançamento da BeOpen. Após isso, Guido e os desenvolvedores do PythonLabs se juntaram ao Digital Creations.

Impacto: Expansão do uso do Python em aplicações internacionais e sistemas de grande escala.

2001: Criação da Python Software Foundation (PSF)

A Python Software Foundation (PSF), fundada em 2001, gerencia os direitos sobre Python e promove o PyCon US. Em 2019, concedeu mais de US\$ 460.000 em doações e foi pioneira na implementação de um Código de Conduta. Seu impacto fortalece a comunidade global e incentiva boas práticas.

Impacto: Seu impacto fortalece a comunidade global e impulsiona o crescimento de Python e boas práticas em eventos e projetos open-source.

2002: Primeira edição da EuroPython

A EuroPython começou em 2002 como uma iniciativa comunitária para reunir desenvolvedores e entusiastas de Python e Zope. Cresceu e se tornou o maior evento de Python na Europa, abrangendo áreas como web, ciência de dados, robótica e HPC.

Impacto: Fortaleceu a comunidade e consolidou Python como linguagem central em ciência, tecnologia e desenvolvimento na Europa.

Expansão (2003-2008)

2003: Primeira edição da PyCon

A PyCon é a maior conferência anual de Python, com edições em 40+ países, tutoriais e treinamentos. Pioneira em código de conduta, conta com grupos como PyLadies e Django Girls.

Organizada pela PSF, é patrocinada por Microsoft, Google e Facebook.

Impacto: Sua influência cresce com a popularidade de Python em ciência, big data e engenharia, com participação de Guido van Rossum.

2005: Criação do Django

Foi criado o Django, um framework de desenvolvimento web em Python, foi lançado. Ele simplificou o processo de construção de sites dinâmicos e escaláveis, tornando a criação de aplicativos web mais acessível.

Impacto: Python foi amplamente adotado para o desenvolvimento web, influenciando o design e a arquitetura da Web.

2008: Python 3.0

Lançado em dezembro de 2008, o Python 3.0 (ou "Python 3000") foi o primeiro lançamento intencionalmente incompatível com versões anteriores, visando corrigir falhas no design da linguagem e reduzir duplicação.

Impacto: Embora tenha exigido adaptações, a versão 3.0 trouxe melhorias que facilitaram a leitura e manutenção do código.

Consolidação (2010-2012)

2010: Python 2.7

Versão do python 2.7 foi liberada em 3 de Julho de 2010, ela introduziu diversas melhorias e foi a última da série 2.x.

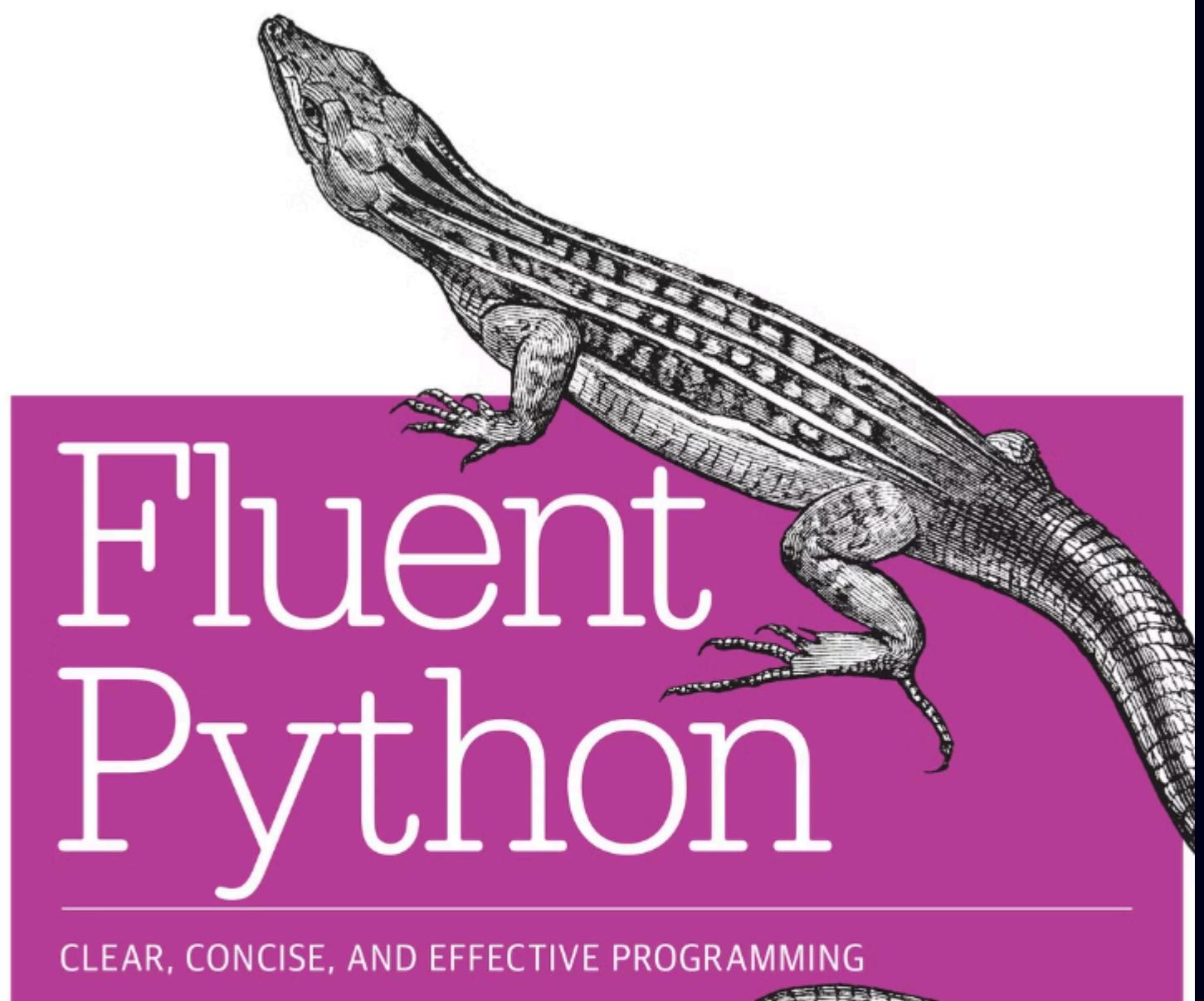
Impacto: Fornecimento de uma versão estável para projetos que ainda não migraram para o Python 3.

2012: Raspberry Pi

Em 2012, o Raspberry Pi foi lançado, um microcomputador de baixo custo que adotou o Python como uma de suas principais linguagens de programação, incentivando a educação em informática e o desenvolvimento de projetos de hardware.

Impacto: Incentivo à educação em informática e ao desenvolvimento de projetos de hardware, ampliando a base de usuários do Python.





Luciano Ramalho

Evolução Técnica (2015-2016)



2015: Python Fluente

Publicado o livro Python fluente, publicado pela O'Reilly Media nos Estados Unidos.

Impacto: Fornecimento de recursos para desenvolvedores aprimorarem suas habilidades e compreensão da linguagem.



2016: Python 3.5

Lançamento do Python 3.5 em 13 de setembro de 2015, introduzindo recursos como o operador "@" para multiplicação de matrizes e a adição de corrotinas baseadas em `async/await`.

Impacto: Aprimoramento da linguagem para suportar programação assíncrona e operações matemáticas avançadas.

Mudança na Liderança (2018)

Transição de Liderança

Após quase três décadas liderando o desenvolvimento do Python, Guido decide passar o bastão para a comunidade.

Governança Comunitária

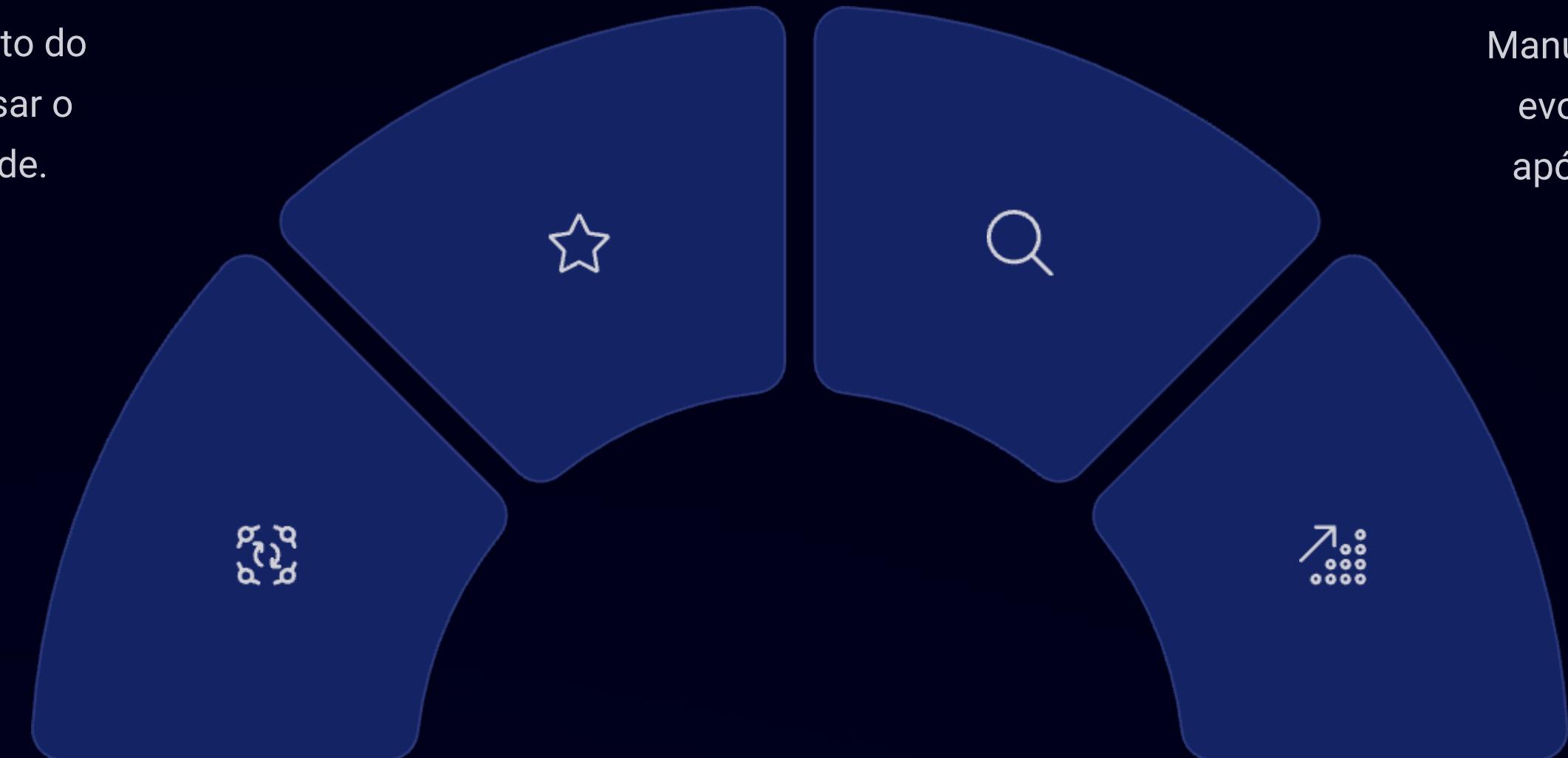
Fortalecimento da governança comunitária e incentivo à participação ativa de desenvolvedores no futuro da linguagem.

Desenvolvimento Colaborativo

Maior participação da comunidade nas decisões sobre o futuro da linguagem.

Crescimento Contínuo

Manutenção do desenvolvimento e evolução da linguagem mesmo após a saída do criador original.



Popularidade Global (2020-2021)

1º

Posição no GitHub

Python ultrapassa JavaScript em número de repositórios no GitHub. O Python se torna a linguagem com o maior número de repositórios ativos na plataforma de desenvolvimento colaborativo GitHub.

3.9

Versão Lançada

Em 2021, o Python 3.9 foi lançado, trazendo melhorias como a introdução de tipos genéricos e a função `zone info` para fusos horários.

2020

Ano de Consolidação

Impacto: Reflexo de sua popularidade crescente e versatilidade em diversos domínios, desde ciência de dados até desenvolvimento web.



Python Hoje (2024)

```
def __init__(self, path=None, debug=False):
    self.file = None
    self.fingerprints = set()
    self.logdups = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(path, 'fingerprint.log'), 'a')
        self.file.seek(0)
        self.fingerprints.update(self._read_file())
    else:
        self._log('No log file specified')

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('SUPERVISOR_DEBUG', False)
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```



2024 - Lançamento da versão 3.13.0. Continuação do aprimoramento da linguagem com novas funcionalidades e otimizações de desempenho. Impacto: Manutenção do Python como uma das principais escolhas para desenvolvedores, garantindo eficiência e modernidade no desenvolvimento de software. Forneceu aos desenvolvedores uma versão mais estável e confiável, garantindo melhor desempenho e segurança em aplicações existentes.

FONTES E REFERÊNCIAS:

<https://docs.python.org/pt-br/3/license.html>

https://en.wikipedia.org/wiki/History_of_Python

https://en.wikipedia.org/wiki/Guido_van_Rossum

https://pt.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica

https://en.wikipedia.org/wiki/Python_%28programming_language%29

<https://docs.python.org/pt-br/3/whatsnew/3.13.html>

<https://www.dio.me/articles/a-historia-do-python-da-criacao-a-popularidade-global>

<https://awari.com.br/python-version-a-evolucao-da-linguagem-de-programacao-python-2/>

<https://www.python.org/downloads/release/python-399/>

<https://docs.python.org/pt-br/3.8/faq/general.html#why-was-python-created-in-the-first-place>

<https://wiki.python.org/moin/EuroPython>

https://en.m.wikipedia.org/wiki/Python_Conference

https://pt.wikipedia.org/wiki/Python_Software_Foundation



PYTHON

FIM!

Obrigado pela atenção!