



PYTHON  
ACADEMY

# PYPY: EXECUTE CÓDIGOS PYTHON ATÉ 4X MAIS RÁPIDO!

Quer saber como executar seu código 4x mais rápido? Então leia esse ebook  
AGORA!

[PYTHONACADEMY.COM.BR](https://pythonacademy.com.br)

Este ebook foi gerado por



# Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

 Syntax Highlight

 Adicione Banners Promocionais

 Edite em Markdown em Tempo Real

 Infográficos feitos por IA

**TESTE AGORA** 

 PRIMEIRO CAPÍTULO 100% GRÁTIS

Olá Pythonistas!

O post de hoje vai mostrar como seu projeto pode rodar **mais rápido com PyPy!**

PyPy é um compilador alternativo ao compilador padrão do Python: o CPython.

Seu foco é **velocidade**: sua documentação traz a informação que o PyPy é, em média, **4,2x mais rápido** que a implementação padrão!

*Está ansioso para ver essa belezinha rodando seu código?*

*Prepara seu café e vamos nessa!*

## Introdução

Por padrão, a linguagem Python vem com o CPython por baixo dos panos.

*Mas, espere aí... O que é CPython?*

Talvez muitos de vocês não saibam o que ele é.

Basicamente, o CPython é a implementação de referência do Python. É o responsável por interpretar seu código e executá-lo.

CPython é um interpretador e um compilador de código Python (ele compila o código em bytecode). Também é onde temos a gramática da linguagem Python.

*Ok, mas o que é PyPy e de onde vem?*

PyPy é uma implementação alternativa ao CPython que segue a especificação oficial da linguagem Python e faz quase o mesmo trabalho do CPython, mas com algumas diferenças interessantes.

O PyPy é um compilador *Just in Time* (JIT), ele usa uma técnica conhecida como *meta-tracing*, que é responsável por transformar um interpretador em um compilador JIT.

Em resumo: quando você executa seu código Python baixado via `apt-get`, ou compilado do próprio site do Python, você provavelmente está usando a forma padrão, que utiliza o CPython.

Contudo, você pode fazer isso de uma maneira diferente, executando seu código Python com o PyPy!

***Ok... Mas por “djabos” eu usaria algo diferente da implementação padrão?***

Porque talvez você precise de **velocidade!**

Se esse for o seu caso, o PyPy pode ser uma boa solução para você!

Nas palavras do criador do Python, **O** cara, Guido van Rossum:

*“Se você deseja que seu código rode magicamente mais rápido, provavelmente deve apenas usar o PyPy.”*

*Disponível em sua apresentação sobre Type Hints em 2015 ([YouTube](#), [minuto 16m52s](#))*

***Animado(a) para usar o PyPy? Então vamos aos detalhes!***

# Compatibilidade

A maioria das bibliotecas Python são executadas no PyPy, exceto aquelas que dependem de extensões CPython, que não funcionam ou incorrem em alguma sobrecarga ao serem executadas no PyPy.

Neste [link](#) temos os 1000 pacotes mais baixados do [pypi.python.org](#), mostrando se é compatível com PyPy ou não.

## Quando não usar PyPy?

Existem casos onde utilizar o PyPy não é aconselhado.

Não opte por utilizá-lo quando: - Desempenho não é uma necessidade; - For necessário usar as funcionalidades mais recentes da linguagem, pois PyPy não está atualizado com as versões do Python; - Seu projeto depender de pacotes que utilizam extensões escritas em C, pois PyPy pode ter problemas de desempenho.

Outro detalhe importante é a arquitetura onde seu projeto vai ser executado.

PyPy oferece compatibilidade com as seguintes arquiteturas: - x86 (IA-32) e x86\_64; - Plataformas ARM (ARMv6 ou ARMv7, com VFPv3); - AArch64; - PowerPC 64 bits, tanto *Little Endian* quanto *Big Endian*; - System Z (s390x)

Portanto, se seu projeto vai rodar em outra arquitetura, o PyPy **não** é opção.

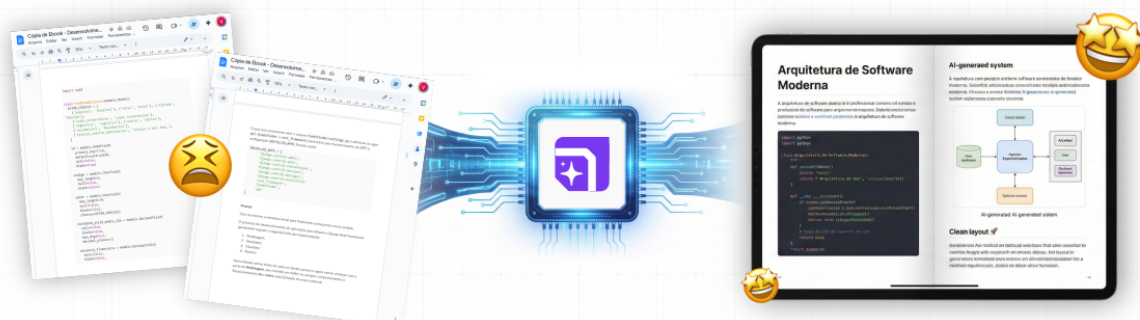


*Estou construindo o **DevBook**, uma plataforma que usa IA para criar ebooks técnicos — com código formatado e exportação em PDF. Depois de ler, dá uma passada lá!*



## Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

## Aumento de uso de memória durante tempo de execução

Tenha cuidado e preste atenção a isso: o *Garbage Collector do PyPy* é diferente do CPython.

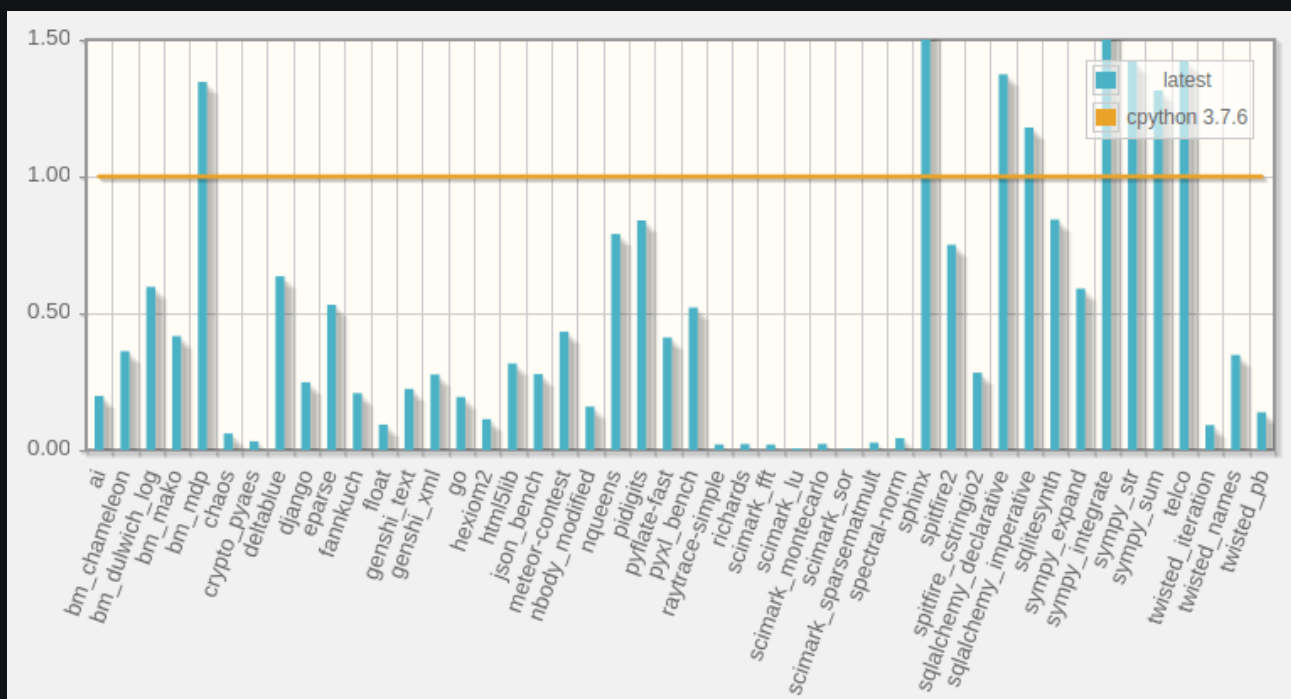
Existem *alguns relatórios* em que desenvolvedores têm relatado problemas de alto consumo de memória utilizando PyPy.

Portanto, antes de decidir por usar o PyPy, estude o `garbage collector` e a maneira como deve configurá-lo e usá-lo em seu projeto.

# Benchmarks

A documentação do PyPy traz diversos gráficos comparativos de velocidade que eles apelidaram de **Speed Center!**

Abaixo um exemplo comparando o PyPy com o CPython em diversos scripts de *benchmark*:



De acordo com o [site oficial do PyPy](#), a diferença de velocidade entre o CPython e o PyPy:

*Depende muito do tipo de tarefa que está sendo realizada. A média geométrica de todos os benchmarks é 0,23 ou 4,4 vezes mais rápida que o CPython*

Se você quiser saber mais detalhadamente, eu recomendo esse incrível post do Hackernoon: [Which is the fastest version of Python?](#)

# Como instalar e usar PyPy?

*E aí, ficou intrigado para ver seu projeto voando baixo?*

Que tal dar uma chance ao PyPy?

```
$ python
Python 2.7.13 (default, Dec 18 2016,
07:03:39)
>>> print ("Hello World")
Hello World
>>>
>>> [x+1 for x in xrange(0,20, 2)]
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
>>>
```

```
$ python your-application.py
I'm a python application
```

```
$ pypy
Python 2.7.13[PyPy 5.8.0 with GCC 4.2.1]
on darwin
>>>> print ("Hello World")
Hello World
>>>>
>>>> [x+1 for x in xrange(0,20, 2)]
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
>>>>
```

```
$ pypy your-application.py
I'm a python application
```

Se você deseja testar o PyPy e está usando o Ubuntu, basta usar o bom e velho `apt-get` :

```
sudo apt-get install -y pypy3
```

E para utilizá-lo, digite em seu terminal:

```
pypy3 seu_script.py
```

Ou para abrir o terminal iterativo do Python, basta digitar `pypy3` .

Observação: não se esqueça das arquiteturas compatíveis com o PyPy!



# Conclusão

Espero ter deixado claro que o PyPy não é para todos os projetos!

Você precisa entender suas necessidades e se faz sentido para o seu cenário.

Se nunca sentiu necessidade de melhoria de velocidade em seu projeto provavelmente não precisa do PyPy. Agora, se já tiver sentido essa necessidade pode ser que o PyPy te ajude.

Faça testes de performance com PyPy e CPython, monitore o consumo de recursos e analise se os ganhos valem a pena em seu projeto.

***E você? Já conhecia o PyPy?***

***Já trabalhou em um projeto que usa o PyPy? Notou diferença de velocidade?***

***Deixe aqui nos comentários!*** 🙏 🙏 🙏

Espero que tenham gostado e que eu tenha contribuído um pouco com o estudo de vocês.

Vejo vocês na próxima!

Não se esqueça de conferir!



DevBook

# Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



 Syntax Highlight

 Infográficos feitos por IA

 Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

 Edite em Markdown em Tempo Real

**TESTE AGORA** 

 PRIMEIRO CAPÍTULO 100% GRÁTIS