



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)



# Introdução ao Cython





# Introdução ao Cython

A principal implementação da linguagem Python, chamada de cPython, é escrita com a linguagem de programação C.

Inclusive se você sabe programar em C ou C++ você pode estender a linguagem Python criando seus próprios módulos.



# Introdução ao Cython

A principal implementação da linguagem Python, chamada de cPython, é escrita com a linguagem de programação C.

Inclusive se você sabe programar em C ou C++ você pode estender a linguagem Python criando seus próprios módulos.

## 1. Estendendo Python com C ou C++

É muito fácil adicionar novos módulos embutidos ao Python, se você souber programar em C. Você pode adicionar 'módulos de extensão' para fazer duas coisas que não podem ser feitas diretamente no Python: eles podem implementar novos tipos de objetos embutidos e eles podem chamar funções da biblioteca C e chamadas do sistema.

To support extensions, the Python API (Application Programmers Interface) defines a set of functions, macros and variables that provide access to most aspects of the Python run-time system. The Python API is incorporated in a C source file by including the header `"Python.h"`.

A compilação de um módulo de extensão depende do uso pretendido e da configuração do sistema; detalhes serão dados nos próximos capítulos.

**Nota:** The C extension interface is specific to CPython, and extension modules do not work on other Python implementations. In many cases, it is possible to avoid writing C extensions and preserve portability to other implementations. For example, if your use case is calling C library functions or system calls, you should consider using the `ctypes` module or the `cffi` library rather than writing custom C code. These modules let you write Python code to interface with C code and are more portable between implementations of Python than writing and compiling a C extension module.



# Introdução ao Cython

Várias bibliotecas importantes do mundo Python são escritas em linguagem C, como é o caso da Numpy, ou mesmo partes da SQLAlchemy.

The screenshot shows the GitHub interface for the `numpy/numpy` repository. The file `array_assign.c` is selected, showing its commit history and code. The commit is by `charris` with the message "MAINT: Replace PyUString\_ConcatAndDel in array\_assign.c". The file has 5 contributors. The code is 170 lines (148 sloc) and 5.07 KB. The code includes a license header and various preprocessor directives.

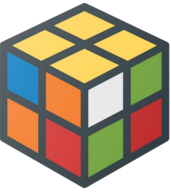
```
1 /*
2  * This file implements some helper functions for the array assignment
3  * routines. The actual assignment routines are in array_assign*.c
4  *
5  * Written by Mark Wiebe (mwwiebe@gmail.com)
6  * Copyright (c) 2011 by Enthought, Inc.
7  *
8  * See LICENSE.txt for the license.
9  */
10
11 #define PY_SSIZE_T_CLEAN
12 #include <Python.h>
13
14 #define NPY_NO_DEPRECATED_API NPY_API_VERSION
15 #define _MULTIARRAYMODULE
16 #include <numpy/ndarraytypes.h>
17 #include "npconfig.h"
18 #include "npypcompat.h"
19
```



# Introdução ao Cython

Isso porque em alguns tipos de problemas, Python não é tão performático.

Mas a forma com que Python foi criado, de forma modular, permite que os propósitos de uso da linguagem sejam infinitos, pois podemos mesclar Python com C/C++ e até mesmo outras linguagens como Rust.



# Introdução ao Cython

Mas as linguagens C, C++ ou Rust não são as mais amigáveis e você pode não ser muito familiarizado com elas.

É aí que entra o **Cython**.



# Introdução ao Cython

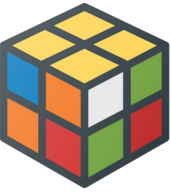
Cython é um compilador estático otimizado para a linguagem de programação Python, tornando a escrita de extensões C para Python tão fácil quanto o próprio Python.



Podemos instalar facilmente com: *pip install Cython*

<https://cython.org/>





# Introdução ao ython

Escrevendo código com Cython, ao compilarmos o programa, o código é transformado em linguagem C e então compilado para linguagem de máquina, da mesma forma que ocorre quando escrevemos código em linguagem C.

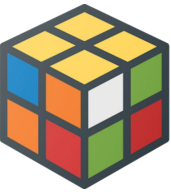
Ou seja, você não precisa conhecer a linguagem C para escrever código extremamente performático compilado para a máquina.



# Introdução ao ython

## Por quê Cython?

- Podemos escrever código Python que facilmente se comunica com códigos nativos em C/C++;
- Podemos facilmente adicionar performance extrema comparável à linguagem C ao nosso código bastando adicionar algumas declarações estáticas de tipo de dados;
- É possível integrar facilmente o debug de código, tanto da parte em Python, quanto Cython e linguagem C;
- Pode fazer uso eficiente de quantidades gigantes de dados, assim como Numpy consegue fazer;
- A biblioteca Cython é madura e amplamente usada no mundo Python;
- Você pode integrar Cython nativamente com qualquer outra biblioteca Python;



# Introdução ao ython

## Por quê Cython?

- Podemos escrever código Python que facilmente se comunica com códigos nativos em C/C++;
- Podemos facilmente adicionar performance extrema comparável à linguagem C ao nosso código bastando adicionar algumas declarações estáticas de tipo de dados;
- É possível integrar facilmente o debug de código, tanto da parte em Python, quanto Cython e linguagem C;
- Pode fazer uso eficiente de quantidades gigantes de dados, assim como Numpy consegue fazer;
- A biblioteca Cython é madura e amplamente usada no mundo Python;
- Você pode integrar Cython nativamente com qualquer outra biblioteca Python;

Na próxima aula iremos comparar e compreender a sintaxe de Python vs Cython...



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)