

Cython概述



Azuki_N [关注](#) IP属地: 北京

2022.02.26 23:03:24 字数 446 阅读 5,510

Cython是什么？

1. [Cython](#)是一种编程语言，它使得为 **Python 语言编写 C 扩展**就像 Python 本身一样容易。
2. 它旨在成为[Python](#)语言的超集，**赋予它高级、面向对象、函数式和动态编程**。
3. 除此之外，它的主要功能是**支持作为语言一部分的可选静态类型声明**。源代码被翻译成优化的 C/C++ 代码并编译为 Python 扩展模块。这允许**非常快速的程序执行**和与外部 C 库的紧密集成，同时保持 Python 语言众所周知的高程序员生产力。

关于Cython运行的原理

[Cython之所以能加速的原理](#)

Cython的安装

[Cython的安装教程](#)

第一个Cython代码

首先编辑 `hello.pyx`：

```
1 def say_hello_to(name):
2     print("Hello %s!" % name)
```

因为Cython可以理解为Python的超集，所以Python的语法适用于Cython

- 现在我们将 `hello.pyx` 用Cython编译成 `.c` 文件
- 该 `.c` 文件由 C 编译器编译成一个 `.so` 文件（或 `.pyd` 在 Windows 上），该文件可以 `import` 直接编辑到 Python 会话中。[setuptools](#)负责这部分。
- 编写一个 `setuptools setup.py` 来构建Cython 代码

（虽然使用[Pyximport](#)可以使 `.pyx` 像 `.py` 文件一样导入 Cython文件，但是不推荐，所以之后也不进行太多记录）

`setup.py`：

```
1 from setuptools import setup
2 from Cython.Build import cythonize
3
4 setup(
5     name='Hello world app',
6     ext_modules=cythonize("hello.pyx"),
7     zip_safe=False,
8 )
```

之后需要在命令行进入 `.pyx` 路径使用 `python setup.py build_ext --inplace` 即可生成 `.pyd`，改文件可以直接被调用

```
1 import hello
2 hello.say_hello_to('Ning')
```

```
4  
5 | 运行结果为:  
   Hello Ning!
```
