

当我们编写Python代码时，我们得到的是一个包含Python代码的以.py为扩展名的文本文件。要运行代码，就需要Python解释器去执行.py文件。

## Python解释器

由于整个Python语言从规范到解释器都是开源的，所以理论上，只要水平够高，任何人都可以编写Python解释器来执行Python代码（当然难度很大）。事实上，确实存在多种Python解释器。

### CPython

当我们从[Python官方网站](#)下载并安装好Python 3.x后，我们就直接获得了一个官方版本的解释器：CPython，这个解释器是用C语言开发的，所以叫CPython，在命令行下运行python就是启动CPython解释器。

CPython是使用最广的Python解释器，教程的所有代码也都在CPython下执行。

### IPython

IPython是基于CPython之上的一个交互式解释器，也就是说，IPython只是在交互方式上有所增强，但是执行Python代码的功能和CPython是完全一样的，好比很多国产浏览器虽然外观不同，但内核其实都是调用了IE。

CPython用>>>作为提示符，而IPython用In (序号):作为提示符。

### PyPy

PyPy是另一个Python解释器，它的目标是执行速度，PyPy采用[JIT技术](#)，对Python代码进行动态编译（注意不是解释），所以可以显著提高Python代码的执行速度。

绝大部分Python代码都可以在PyPy下运行，但是PyPy和CPython有一些是不同的，这就导致相同的Python代码在两种解释器下执行可能会有不同的结果，如果你的代码要放到PyPy下执行，就需要了解[PyPy和CPython的不同点](#)。

### Jython

Jython是运行在Java平台上的Python解释器，可以直接把Python代码编译成Java字节码执行。

### IronPython

IronPython和Jython类似，只不过IronPython是运行在微软.Net平台上的Python解释器，可以直接把Python代码编译成.Net的字节码。

### 小结

Python的解释器很多，但使用最广泛的还是CPython，如果要和Java或.Net平台交互，最好的办法不是用Jython或IronPython，而是通过网络调用来交互，确保各程序之间的独立性。

本教程的所有代码只确保在CPython 3.x版本下运行，请务必在本地安装CPython（也就是从Python官方网站下载的安装程序）。