chardet

字符串编码一直是令人非常头疼的问题，尤其是我们在处理一些不规范的第三方网页的时候。虽然Python提供了Unicode表示的str和bytes两种数据类型，并且可以通过encode()和decode()方法转换，但是，在不知道编码的情况下，对bytes做decode()不好做。

对于未知编码的bytes，要把它转换成str，需要先“猜测”编码。猜测的方式是先收集各种编码的特征字符，根据特征字符判断，就能有很大概率“猜对”。

当然，我们肯定不能从头自己写这个检测编码的功能，这样做费时费力。chardet这个第三方库正好就派上了用场。用它来检测编码，简单易用。

安装chardet

如果安装了Anaconda，chardet就已经可用了。否则，需要在命令行下通过pip安装：

* pip install chardet

如果遇到Permission denied安装失败，请加上sudo重试。

使用chardet

当我们拿到一个bytes时，就可以对其检测编码。用chardet检测编码，只需要一行代码：

* chardet.detect(b'Hello, world!')

{'encoding': 'ascii', 'confidence': 1.0, 'language': ''}

检测出的编码是ascii，注意到还有个confidence字段，表示检测的概率是1.0（即100%）。

我们来试试检测GBK编码的中文：

* data = '离离原上草，一岁一枯荣'.encode('gbk')
* chardet.detect(data)

{'encoding': 'GB2312', 'confidence': 0.7407407407407407, 'language': 'Chinese'}

检测的编码是GB2312，注意到GBK是GB2312的超集，两者是同一种编码，检测正确的概率是74%，language字段指出的语言是'Chinese'。

对UTF-8编码进行检测：

* data = '离离原上草，一岁一枯荣'.encode('utf-8')
* chardet.detect(data)

{'encoding': 'utf-8', 'confidence': 0.99, 'language': ''}

我们再试试对日文进行检测：

* data = '最新の主要ニュース'.encode('euc-jp')
* chardet.detect(data)

{'encoding': 'EUC-JP', 'confidence': 0.99, 'language': 'Japanese'}

可见，用chardet检测编码，使用简单。获取到编码后，再转换为str，就可以方便后续处理。

chardet支持检测的编码列表请参考官方文档[Supported encodings](https://chardet.readthedocs.io/en/latest/supported-encodings.html)。



小结

使用chardet检测编码非常容易，chardet支持检测中文、日文、韩文等多种语言。