# **python编码规范**

****(一)代码编码：****

python3在内存中是用unicode编码方式存储的，所以不能直接储存和传输，要转化为其他编码进行储存和传输。

字符串通过编码转换成字节码，字节码通过解码成为字符串

encode：str --> bytes

decode：bytes --> str

国际惯例，文件编码和 Python 编码格式全部为 utf-8 ，例如：在 Python 代码的开头，要统一加上 *#coding=utf-8 或 # -\*- coding: UTF-8 -\*-*

UTF-8（8位元，Universal Character Set/Unicode Transformation Format）是针对Unicode的一种可变长度字符编码。它可以用来表示Unicode标准中的任何字符，而且其编码中的第一个字节仍与[ASCII](https://baike.baidu.com/item/ASCII/309296" \t "https://baike.baidu.com/item/UTF-8/_blank)相容，使得原来处理ASCII字符的软件无须或只进行少部份修改后，便可继续使用。因此，它逐渐成为[电子邮件](https://baike.baidu.com/item/%E7%94%B5%E5%AD%90%E9%82%AE%E4%BB%B6/111106" \t "https://baike.baidu.com/item/UTF-8/_blank)、[网页](https://baike.baidu.com/item/%E7%BD%91%E9%A1%B5/99347" \t "https://baike.baidu.com/item/UTF-8/_blank)及其他存储或传送文字的应用中，优先采用的编码。

UTF-8编码把一个Unicode字符根据不同的数字大小编码成1-6个字节，常用的英文字母被编码成1个字节，汉字通常是3个字节，只有很生僻的字符才会被编码成4-6个字节。如果你要传输的文本包含大量英文字符，用UTF-8编码就能节省空间

****(二)命名规范:****  
1、包名、模块名、局部变量名、函数名

全小写+下划线式驼峰 示例：this\_is\_var

2、全局变量

全大写+下划线式驼峰 示例：GLOBAL\_VAR

3、类名

首字母大写式驼峰 示例：ClassName()

4、变量名命名

尽量体现变量的数据类型和具体意义

注：

变量名、类名取名必须有意义，严禁用单字母

变量名不要用系统关键字，如 dir type str等等

建议：

bool变量一般加上前缀 is\_ 如：is\_success

****(三)、****import 语句的模块顺序****:****

推荐所有的模块在 Python 模块的开头部分导入。

而且最好按照以下顺序:  
1，Python 标准库模块  
2，Python 第三方模块  
3，应用[程序](https://www.ctolib.com/" \l "/dev/" \t "https://www.ctolib.com/amp/_blank)自定义模块  
然后，使用一个空行分割这三类模块的导入语句。

这将确保模块使用固定的习惯导入, 有助于减少每个模块需要的 import 语句数目。