

## 15.15 C字符串转换为Python字符串¶

### 问题¶

怎样将C中的字符串转换为Python字节或一个字符串对象？

### 解决方案¶

C字符串使用一对 `char *` 和 `int` 来表示， 你需要决定字符串到底是用一个原始字节字符串还是一个Unicode字符串来表示。 字节对象可以像下面这样使用 `Py_BuildValue()` 来构建：

```
char *s;      /* Pointer to C string data */
int  len;     /* Length of data */

/* Make a bytes object */
PyObject *obj = Py_BuildValue("y#", s, len);
```

如果你要创建一个Unicode字符串，并且你知道 `s` 指向了UTF-8编码的数据，可以使用下面的方式：

```
PyObject *obj = Py_BuildValue("s#", s, len);
```

如果 `s` 使用其他编码方式，那么可以像下面使用 `PyUnicode_Decode()` 来构建一个字符串：

```
PyObject *obj = PyUnicode_Decode(s, len, "encoding", "errors");

/* Examples */
obj = PyUnicode_Decode(s, len, "latin-1", "strict");
obj = PyUnicode_Decode(s, len, "ascii", "ignore");
```

如果你恰好有一个用 `wchar_t *`，`len` 对表示的宽字符串， 有几种选择性。首先你可以使用 `Py_BuildValue()`：

```
wchar_t *w;    /* Wide character string */
int len;       /* Length */

PyObject *obj = Py_BuildValue("u#", w, len);
```

另外，你还可以使用 `PyUnicode_FromWideChar()`：

```
PyObject *obj = PyUnicode_FromWideChar(w, len);
```

对于宽字符串，并没有对字符串数据进行解析——它被假定是原始Unicode编码指针，可以被直接转换成Python。

### 讨论¶

将C中的字符串转换为Python字符串遵循和I/O同样的原则。也就是说，来自C中的数据必须根据一些解码器被显式的解码为一个字符串。通常编码格式包括ASCII、Latin-1和UTF-8. 如果你并不确定编码方式或者数据是二进制的，你最好将字符串编码成字节。当构造一个对象的时候，Python通常会复制你提供的字符串数据。如果有必要的话，你需要在后面去释放C字符串。同时，为了让程序更加健壮，你应该同时使用一个指针和一个大小值，而不是依赖NULL结尾数据来创建字符串。