

TI-Lua String Library Extension

ustring function库使用说明

简介

此库提供了一些基本函数，用于增强TI-Lua处理UTF8字符串的能力。

如果需要大量的比较，插入，连接，查找等操作或者希望使用类C++的字符串类，建议使用ustring class库。

此库所有函数均支持利用assert进行错误处理。

成员列表

- 此库包含以下函数/异常:
- 异常列表:

__ustring.exception.invType	--输入数据类型非法
__ustring.exception.outRange	--输入数据超出范围
__ustring.exception.invPos	--无效的字符位置
 - 函数列表:

string.ubyte(uchar)	string.byte的UTF8版本
string.ulen(ustr)	string.len的UTF8版本
string.UTF8CharLen(unum)	返回用UTF8编码一个Unicode字符所需字节数
string.uRealPos(ustr, pos)	返回某个字节所在UTF8字符在UTF8字符串中的真正位置
string.MbtoUTF8table(mbstr)	将一个字符串转换为一个包含每个字符Unicode编码的table
string.UTF8tabletoMb(UTF8table)	将一个包含每个字符Unicode编码的table转换为一个字符串

使用说明

P.S:由于TI-LUA的限制，此库仅能处理U+0000~U+FFFF范围内的字符， 所以一个UTF8字符最多需要3字节的空间。

函数功能

string.ubyte(uchar)

- 形参列表:

uchar	待转换的UTF8字符，类型必须为string， 否则抛出__ustring.exception.invType异常
-------	---
- 返回值列表:

res	转换的结果(0 ~ 65535)，若为nil则表明发生异常
err	异常内容，若为nil表示无异常

返回uchar的Unicode编码。与string.byte不同的是，这个函数仅支持单个UTF8字符。即总是处理输入字符串的第一个UTF8字符。

- e.g:

string.uchar("a")	--返回 97 nil
string.uchar("测试")	--返回 27979 nil
assert(string.uchar(nil))	--引发__ustring.exception.invType异常

string.ulen(ustr)

- 形参列表:

ustr 待处理的UTF8字符串，类型必须为string， 否则抛出__ustring.exception.invType异常

- 返回值列表：

res 字符串的实际长度，若为nil则表明发生异常
err 异常内容，若为nil表示无异常

返回UTF8字符串的实际长度，例如字符串“这是一个测试”，如果使用string.len会得到18(即18个字节)，而使用本函数则会得到真实长度6。

- e.g:

```
string.ulen("test")           --返回4 nil
string.ulen("这是一个测试test") --返回10 nil
assert(string.ulen(123456))    --引发__ustring.exception.invType异常
assert(string.ulen(tostring(123456))) --返回6
```

string.UTF8CharLen(unum)

- 形参列表：

unum 待转换的UTF8字符，类型必须为number， 否则抛出__ustring.exception.invType异常
若范围超出0 ~ 65535则抛出__ustring.exception.outRange异常

- 返回值列表：

res 所需的字节数(1 ~ 3)，若为nil则表明发生异常
err 异常内容，若为nil表示无异常

返回用UTF8编码一个Unicode字符所需字节数。

- e.g:

```
string.UTF8CharLen(65)    --返回 1 nil
string.UTF8CharLen(176)   --返回 2 nil
string.UTF8CharLen(27979) --返回 3 nil

assert(string.UTF8CharLen(123456)) --引发__ustring.exception.outRange异常
```

string.uRealPos(ustr, pos)

- 形参列表：

ustr 待处理的UTF8字符串，类型必须为string， 否则抛出__ustring.exception.invType异常
pos 字节所在位置， 类型必须为number，否则抛出__ustring.exception.invType异常
若pos不是整数则抛出__ustring.exception.invPos异常
若pos超出字符串范围则抛出__ustring.exception.outRange异常

- 返回值列表：

res 字符串的实际长度，若为nil则表明发生异常
err 异常内容，若为nil表示无异常

返回某个字节所在UTF8字符在UTF8字符串中的真正位置。pos可以为负数，此时会从末尾起进行计数。

- e.g:

```
string.uRealPos("中文测试", 2) --返回1 nil, 即第2个字节为"中"字的第二个字节，而"中"在整个字符串中为第1个字符

string.uRealPos("中文测试", -1) --返回4 nil, 即最后一个字节为"试"字的第3个字节，而"试"在整个字符串中为第4个字符

string.uRealPos("中文test测试", 7) --返回3 nil, 即第7个字节为"t", 而"t"在整个字符串中为第3个字符
```

string.MbtoUTF8table(mbstr)

- 形参列表：

mbstr 待处理的UTF8字符串，类型必须为string， 否则抛出__ustring.exception.invType异常

- 返回值列表：

res 含有每个字符Unicode编码的table
err 异常内容，若为nil表示无异常

将一个字符串转换为一个包含每个字符Unicode编码的table。

- e.g:

```
assert(string.MbtoUTF8table("中文测试test测试")) --返回{20013,25991,27979,35797,116,101,115,116,27979,35797}
```

string.UTF8tabletoMb(UTF8table)

- 形参列表:

UTF8table 含有每个字符的Unicode编码的table, 类型必须为table, 否则抛出__ustring.exception.invType异常

- 返回值列表:

res 转换后的UTF8字符串
err 异常内容, 若为nil表示无异常

将一个包含每个字符Unicode编码的table转换为一个字符串。

- e.g:

```
string.UTF8tabletoMb({20013,25991,27979,35797}) --返回 "中文测试" nil
```

演示样例

样例文件为\samples\lstring\lsssample.lua

在这个样例中, 可以粘贴任意字符串, 之后会根据Unicode编码的数值大小进行排序然后显示。

版权信息

TI-Lua String Library Extension
Copyright 2014 wtof1996
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

P.S: 若使用本库请在你的源代码中保留上述信息或在关于信息中注明上述信息(若太长可做简单标注, 但要求能体现作者、名称、许可等主要信息, 如:"使用了wtof1996 编写的TI-Lua String Library Extension库, 此库按 Apache 2.0许可进行授权")

作者信息

设计&开发:wtof1996

英文校对:chsi

参与测试:

Email: wtof1996 at gmail.com

BUG&建议请一律发送至上述Email, 欢迎提交。