

魏先宇的程序人生

多读, 多写, 多思, 方有多得!

登录 | 注册

目录视图摘要视图RSS 订阅

个人资料



more2010wei

访问: 900252次
积分: 9899
等级: 6
排名: 第815名

原创: 123篇 转载: 115篇
译文: 10篇 评论: 93条

文章搜索

文章分类

- iphone开发之视图部分 (40)
- iphone开发之视图部分-scollview (3)
- iphone开发之数据部分 (38)
- iphone开发之内存部分 (19)
- iphone开发之框架部分 (16)
- iphone开发之功能实现部分 (18)
- iphone开发之多线程 (7)
- iphone开发之事件响应 (11)
- iphone开发之语法部分 (20)
- iphone开发之资料收集 (13)
- iphone开发之文件操作 (7)
- iphone开发之动画部分 (5)
- Quartz 2D (7)
- 关于调试 (8)

想听课? 来发话题吧 CSDN APP 博客上线 双11赚"双喜", 奖励多多 有奖征文: 云服务器使用初体验

NSString / NSMutableString 字符串处理, 常用代码 (实例)

分类: **iphone开发之数据部分** 2012-04-01 09:15 21965人阅读 评论(2) 收藏 举报

c C encoding Encoding extension path PATH Path string String

Objective-C 中核心处理字符串的类是 NSString 与 NSMutableString , 这两个类最大的区别就是 NSString 创建赋值以后该字符串的内容与长度不能在动态的更改, 除非重新给这个字符串赋值。而NSMutableString 创建赋值以后可以动态在该字符串上更改内容与长度。

NSString 常用方法总结

+(id)stringWithContentsOfFile:path encoding:enc err	创建一个新字符串并将其设置为 path 指定文件的内容, 使用字符编码 enc, 在err 上返回错误
+(id)stringWithContentsOfURL:url encoding:enc err	创建一个新字符串并将其设置为 url 所指向的内容, 使用字符编码 enc, 在err 上返回错误
+(id)string	创建一个新的空字符串
+(id)stringWithString:nsstring	创建一个新字符串, 将其内容设置为 nsstring 内容
-(id)initWithString:nsstring	将字符串内容设置为 nsstring 内容
-(id)initWithContentsOfFile:path encoding:enc error:err	将字符串设置为 path 指定文件的内容, 使用字符编码 enc, 在err 上返回错误
-(id)initWithContentsOfURL:url encoding:enc error:err	将字符串设置为 url 所指向的内容, 使用字符 err 上返回错误
-(NSUInteger)length	返回字符串中字符数目
-(unichar)characterAtIndex:i	返回索引i 所在Unicode 字符
-(NSString*)substringFromIndex:i	返回从索引 i 开始到结尾的子字符串
-(NSString*)substringWithRange:range	根据指定范围返回子字符串
-(NSString*)substringToIndex:i	返回从字符串开始到 索引 i 的子字符串
-	
(NSComparator*)caseInsensitiveCompare:nsstring	比较两个字符串大小, 忽略大小写
-(NSComparator*)compare:nsstring	比较两个字符串大小
-(BOOL)hasPrefix:nsstring	测试字符串是否以 nsstring 开始
-(BOOL)hasSuffix:nsstring	测试字符串是否以 nsstring 结尾
-(BOOL)isEqualToString:nsstring	测试两个字符串是否相等
-(NSString*)capitalizedString	返回字符串, 串中的每个单词的首字母大写, 其余字母小写
-(NSString*)lowercaseString	返回转换为小写的字符串

关于面向对象 (1)

IOS6 (11)

数据库 (6)

XCode (18)

资讯 (6)

C和C++ (5)

生活的智慧 (1)

证书 (2)

IOS5 (2)

越狱功能和私有API (4)

CoreText (2)

UIBezierPath (2)

NSCoding (2)

UIMenuController (1)

UIPageControl (1)

KVC (2)

KVO (2)

NSLock (1)

CALayer (2)

网络 (4)

抓包 (1)

ios开发规范 (2)

ios面试题集锦 (1)

block (12)

数据结构与算法 (4)

关键字 (1)

UIPasteboard (1)

设计模式 (1)

NSDate的使用 (1)

ARC (4)

Push (3)

CoreData (4)

Cocos2d-x (1)

JavaScript (1)

cocos2d (4)

cocos2d-HTML (4)

一步一步学Cocos2d-html5 (0)

storyboard (2)

Swift (0)

cocos2d-x 3.x游戏开发学习笔记 (6)

pop开源库使用笔记 (1)

开源库使用笔记 (3)

性能优化 (3)

armv64 (1)

iphone6 (0)

iphone6的适配 (1)

文章存档

2015年10月 (1)

2015年03月 (7)

2014年12月 (2)

2014年11月 (3)

2014年10月 (2)

展开

阅读排行

-(NSString*)uppercaseString

返回转换为大写的字符串

-(const char*)UTF8String

返回UTF8编码格式的字符串

-(double)doubleValue

返回转换为double 类型的字符串

-(float)floatValue

返回转换为 float 类型的字符串

-(NSInteger)integerValue

返回转换为 NSInteger 类型的新字符串

-(int)intValue

返回转换为 int 的字符串

NSMutableString 可修改字符串常用方法

+(id)stringWithCapacity:size

创建一个字符串, 容量为size大小

-(id)initWithCapacity:size

初始化一个字符串, 容量为size

-(void)setString:nsstring

将字符串设置为 nsstring

-(void)appendString:nsstring

在字符串末尾追加字符串 nsstring

-(void)deleteCharatersInRange:range

删除指定range 中的字符

-(void)insertString:nsstring atIndex:i

以索引 i 为起始位置插入 nsstring

-(void)replaceCharatersInRange:range
withString:nsstring

使用 nsstring 替换 range 指定的字符

-(void)replaceOccurrencesOfString:nsstring
withString:nsstring2 options:opts range:range

根据选项 opts, 使用指定 range 中的 nsstring2 替换所有的 nsstring

来源:

<http://blog.csdn.net/xys289187120/article/details/6777283><http://www.cocoachina.com/iphonedev/sdk/2010/0607/1634.html>

1.创建NSString字符串

NSString 与 char* 最大的区别就是 NSString是一个objective对象,而char* 是一个字节数组。@"字符串" 这个符号为objective-c NSString 字符串常量的标准用法, char* 创建的时候 无需添加@

C代码 ☆

```

01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //经典的字符串赋值
06.     NSString *str0 = @"my name is justcoding!";
07.
08.     //字符串格式化合并并分别包括
09.     //NSString*类型 int类型 char*类型
10.     NSString *str1 = [NSString stringWithFormat:@"我的名字: %@ 我的年龄: %d 我的邮箱: %s", @"justcoding", 25, "justcoding@gmail.com"];
11.
12.     //字符串赋值 参数中只可以写一个字符串 和第一种很像
13.     NSString *str2 = [NSString stringWithString:@"我是字符串"];
14.
15.     //字符串转换为utf-8格式 参数为char*类型
16.     NSString *str3 = [NSString stringWithUTF8String:"字符串转换utf-8格式"];
17.
18.

```

ios的手势操作之UIGestu
(65458)

解决EXC_BAD_ACCESS:
(33461)

关于userInteractionEnab
(23510)

NSString / NSMutableSti
(21932)

Xcode的控制台调试命令
(21129)

关于removeFromSuperv
(20342)

iOS 多线程 锁 互斥 同步
(19224)

关于iphone开发中的@pr
(17105)

IOS-应用之间调用
(16516)

IOS文件操作的两种方式
(16130)

评论排行

Xcode的控制台调试命令 (5)

关于userInteractionEnab (4)

正确理解Navigations的Ba (4)

关于iphone开发中的@pr (4)

Object-C的函数调用机制 (3)

从UIImage的矩阵变换看 (3)

关于addSubview需要注 (3)

视图之UITabBarControll (3)

NSCoding协议 (3)

ios的手势操作之UIGestu (3)

推荐文章

*Spark源码阅读: 第十节
Standalone运行模式解析

* Android基础入门教程——
8.3.16 Canvas API详解(Part 1)

*Android 一个改善的okHttp封装
库

* 美团Android DEX自动拆包及动
态加载简介

*Android实战技巧之四十四:
Hello,Native!

* 如何面试Python后端工程师?

最新评论

关于iphone开发中的@property和
more2010wei: @jstarlet: 其实有
说的, self和不使用self的区别就
是调用和不调用对应的setter和...

关于iphone开发中的@property和
jstarlet: 要是能说下11.-
(id)initWithName:(NSString*)n{
if(self ==...

NSCoding协议
jstarlet: 学习了。我也仅以此, 表
谢谢!2026-08-17 13:07:45.339
OC6_文件归档 202...

关于addSubview需要注意的几个
赵荷: 初学者过来看看, 看明白
了, ths

关于addSubview需要注意的几个
千里马君: 谢谢 终于搞懂这个
addSubview了

NSCoding协议
tongyuhu: 学习了。我也仅以
此, 表谢谢!2015-08-17
13:07:45.339 OC6_文件归档
201...

```
19. //字符串合并
20. int i = 100;
21. char*c = "xuanyusong";
22. NSString *temp = @"我是临时字符串";
23.
24. //在字符串temp的基础继续添加 int i 与 char* c 组成一个新的字符串
25. NSString *str4 = [temp stringByAppendingFormat:@"%d 字
    符型: %s", i, c];
26.
27. //在字符串temp的基础继续添加temp 并组成一个新的字符串
28. NSString *str5 = [temp stringByAppendingString:temp];
29.
30. //字符串输出
31. NSLog(@"str0 = %@", str0);
32. NSLog(@"str1 = %@", str1);
33. NSLog(@"str2 = %@", str2);
34. NSLog(@"str3 = %@", str3);
35. NSLog(@"str4 = %@", str4);
36. NSLog(@"str5 = %@", str5);
37.
38. }
```

2.字符串的遍历

每一个字符串其实是由若干个char字符组成, 字符串的遍历实际上就是将字符串中的每一个字符提取出来。

C代码 ☆

```
01. - (void)viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //经典的字符串赋值
06.     NSString *str = @"YUSONGMOMO";
07.
08.     //字符串的长度
09.     int count = [str length];
10.
11.     NSLog(@"字符串的长度是%d", count);
12.
13.     //遍历字符串中的每一个字符
14.     for(int i = 0; i < count; i++)
15.     {
16.         char c = [str characterAtIndex:i];
17.         NSLog(@"字符串第 %d 位为 %c", i, c);
18.     }
19.
20. }
```

3.字符串的比较

isEqualToString 比较字符串是否完全相等, 大小写不一样也无法完全匹配。

hasPrefix 匹配字符串头

hasSuffix 匹配字符串的尾巴

C代码 ☆

```
01. - (void)viewDidLoad
02. {
03.     [super viewDidLoad];
04. }
```

让Mysql支持Emoji表情
 疯儿的小菜园:
 XCode修改公司名和作者名
 idlqy: 试了没用
 IOS 整体框架类图
 童真的烂漫: 原文在哪啊?
 视图之UITabBarController结构详
 刘小壮: 非常好, 谢谢

```

05.
06.     NSString *str0 = @"justcoding";
07.     NSString *str1 = @"justcoding";
08.
09.
10.     //字符串完全相等比较
11.     if([str0 isEqualToString:str1])
12.     {
13.         NSLog(@"字符串完全相等");
14.     }
15.
16.     //字符串以开头比较
17.     if([str0 hasPrefix:@"just"])
18.     {
19.         NSLog(@"字符串str0以just开头");
20.     }
21.
22.     //字符串以结尾比较
23.     if([str1 hasSuffix:@"coding"])
24.     {
25.         NSLog(@"str1字符串以coding结尾");
26.     }
27. }
  
```

C代码 ☆

```

01. //isEqualToString方法
02. NSString *astring01 = @"This is a String!";
03. NSString *astring02 = @"This is a String!";
04. BOOL result = [astring01 isEqualToString:astring02];
05. NSLog(@"result:%d", result);
  
```

C代码 ☆

```

01. //compare方法 (comparer返回的三种值)
02. NSString *astring01 = @"This is a String!";
03. NSString *astring02 = @"This is a String!";
04. BOOL result = [astring01 compare:astring02] == NSOrderedSame;
05. NSLog(@"result:%d", result);
06. //NSOrderedSame 判断两者内容是否相同
  
```

C代码 ☆

```

01. NSString *astring01 = @"This is a String!";
02. NSString *astring02 = @"this is a String!";
03. BOOL result = [astring01 compare:astring02] == NSOrderedAscending;
04. NSLog(@"result:%d", result);
05. //NSOrderedAscending 判断两对象值的大小 (按字母顺序进行比较, astring02大于astring01为真)
  
```

C代码 ☆

```

01. NSString *astring01 = @"this is a String!";
02. NSString *astring02 = @"This is a String!";
03. BOOL result = [astring01 compare:astring02] == NSOrderedDescending;
04. NSLog(@"result:%d", result);
05. //NSOrderedDescending 判断两对象值的大小 (按字母顺序进行比较, astring02
  
```

小于astring01为真)

C代码 ☆

```
01. //不考虑大 小写比较字符串1
02. NSString *astring01 = @"this is a String!";
03. NSString *astring02 = @"This is a String!";
04. BOOL result = [astring01 caseInsensitiveCompare:astring02] == NS
05. NSLog(@"result:%d", result);
06. //NSOrderedDescending判断两对象值的大小(按字母顺序进行比较, astring02小
    于astring01为 真)
```

C代码 ☆

```
01. //不考虑大小写比较字符串2
02. NSString *astring01 = @"this is a String!";
03. NSString *astring02 = @"This is a String!";
04. BOOL result = [astring01 compare:astring02 options:NSCaseInsensit
05. NSLog(@"result:%d", result);
06. //NSCaseInsensitiveSearch:不区分大小写比较 NSLiteralSearch:进行完全
    比较, 区分大小写 NSNumericSearch:比较字符串的字符个数, 而不是字符值。
```

4. 文件字符串操作 (读写)

C代码 ☆

```
01. //从文件读取字符串:initWithContentsOfFile方法
02. NSString *path = @"astring.text";
03. NSString *astring = [[NSString alloc] initWithContentsOfFile:path
04. NSLog(@"astring:%@", astring);
05. [astring release]; //astring=nil;
```

C代码 ☆

```
01. //写字符串到文件:writeToFile方法
02. NSString *astring = [[NSString alloc] initWithString:@"This is a
03. NSLog(@"astring:%@", astring);
04. NSString *path = @"astring.text";
05. [astring writeToFile: path atomically: YES];
06. [astring release]; //astring=nil;
```

* ios5 不支持release iOS应用开发: 什么是ARC?

C代码 ☆

```
01. //扩展路径
02. NSString *Path = @"~/NSData.txt";
03. NSString *absolutePath = [Path stringByExpandingTildeInPath];
04. NSLog(@"absolutePath:%@", absolutePath);
05. NSLog(@"Path:%@",
    [absolutePath stringByAbbreviatingWithTildeInPath]);
```

C代码 ☆

```
01. //文件扩展名
02. NSString *Path = @"~/NSData.txt";
03. NSLog(@"Extension:%@", [Path pathExtension]);
```

5. 字符串的截取和大小写

C代码 ☆

```
01. //改变字符串的大小写
02. NSString *string1 = @"A String";
03. NSString *string2 = @"String";
04. NSLog(@"string1:%@", [string1 uppercaseString]); //大写
05. NSLog(@"string2:%@", [string2 lowercaseString]); //小写
06. NSLog(@"string2:%@", [string2 capitalizedString]); //首字母大小
```

Java代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.
06.     NSString *str0 = @"中文my name is xuanyusong";
07.
08.     //截取字符串起始点到index为4之间的内容
09.     NSString * to = [str0 substringToIndex:4];
10.
11.     NSLog(@"to = %@", to);
12.
13.     //截取字符串index为2 到结尾之间的内容
14.     NSString * from = [str0 substringFromIndex:2];
15.
16.     NSLog(@"from = %@", from);
17.
18.
19.     //设置截取字符串的范围
20.     //从第二位开始, 长度为十
21.     NSRange rang = NSMakeRange(2, 10);
22.     NSString * strRang = [str0 substringWithRange:rang];
23.     NSLog(@"rang = %@", strRang);
24.
25.     //设置字符串首字母大写
26.     NSLog(@"str0首字母大写:%@", [str0 capitalizedString]);
27.     //设置字符串全部内容为大写
28.     NSLog(@"str0大写:%@", [str0 uppercaseString]);
29.     //设置字符串全部内容为小写
30.     NSLog(@"str0小写:%@", [str0 lowercaseString]);
31.
32. }
```

6. 搜索字符串与替换字符串

C代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04. }
```

```

05.
06.     NSString *str0 = @"中文my name is xuanyusong";
07.     NSString *temp = @"is";
08.     NSRange rang = [str0 rangeOfString:temp];
09.
10.     NSLog(@"搜索的字符串在str0中起始点的
index 为 %d", rang.location);
11.     NSLog(@"搜索的字符串在str0中结束点的
index 为 %d", rang.location + rang.length);
12.
13.     //将搜索中的字符串替换成为一个新的字符串
14.     NSString *str = [str0 stringByReplacingCharactersInRange:rang
    咋咋卡卡咋"];
15.     NSLog(@"替换后字符串为%@", str);
16.
17.
18.     //将字符串中" " 全部替换成 *
19.     str = [str0 stringByReplacingOccurrencesOfString:@" " withSt
20.     NSLog(@"替换后字符串为%@", str);
21.
22.
23. }

```

拓展：使用下面这个方法整体替换字符串还可以设置替换的区域。

stringByReplacingOccurrencesOfString:(NSString *)withString:(NSString *) options:
(NSStringCompareOptions) range:(NSRange)

7. 字符串尾部添加

使用alloc在内存中创建字符串对象后边可以动态的操作这个字符串，修改与添加等。

appendstring 方法：向字符串尾部添加一个字符串。

appendFormat方法：向字符串尾部添加多个类型的字符串，可以添加任意数量与类型的字符串。

C代码 ☆

```

01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.
06.     NSMutableString * str = [[NSMutableString alloc] init];
07.
08.     //添加普通的字符串
09.     [str appendString:@"aaa"];
10.
11.     //添加字符串 整型 字符型
12.     [str appendFormat:@"我的名字：%@ 我的年龄：%d 我的邮箱：
    %s", @"justcoding", 25, "justcoding@gmail.com"];
13.
14.     NSLog(@"str = %@", str);
15.
16. }

```

8. 字符串中删除元素

stringWithString 方法：用于创建字符串初始化赋值

rangeOfString方法：传入字符串返回一个在该字符串中的范围 也可以写 NSRange(0, 3) 意思是范围在字符串0位到第3位

`deleteCharactersInRange`:删除字符串 参数为Range 就是删除的范围。

C代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //创建字符串
06.     NSMutableString *str = [NSMutableString stringWithString:@"ju
07.
08.     //删除字符串中含"justcoding"的字符
09.     [str deleteCharactersInRange: [str rangeOfString:@"justcodin
10.
11.     NSLog(@"str = %@", str);
12.
13. }
```

9. 字符串插入

`stringWithString` 方法: 用于创建字符串初始化赋值

`insertString`方法: 第一个参数 插入的字符串对象, 第二个参数 插入的位置。

C代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //创建字符串
06.     NSMutableString *str = [NSMutableString stringWithString:@"ju
07.
08.     //在str第10位插入字符串
09.     [str insertString:@"best man" atIndex:10];
10.
11.     NSLog(@"str = %@", str);
12.
13. }
```

10. 字符串拷贝

C代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //创建字符串
06.     NSMutableString *str1 = [NSMutableString stringWithString:@"
字符串1"];
07.     NSMutableString *str2;
08.
09.     //字符串赋值
10.     str2 = str1;
11.
12.     [str2 appendString:@" 和字符串2"];
13.
14.     NSLog(@"str1 = %@", str1);
15.
16.     NSLog(@"str2 = %@", str2);
17. }
```


为什么给str2添加数据后 str1的数据也改变了? 这就是指针的魅力所在, 因为我们操作的是指针, str2 = str1 意思是两个指针同时指向一块内存, 那么str2指向的内存内容改变以后str1当然跟着改变了。

11. 字符串与指定类型转换

如果转换的参数非法的话不会抛出异常, 好比用中文去转整型。不会报错 但是转换结果为 0 , 默认值。

C代码 ☆

```
01. - (void) viewDidLoad
02. {
03.     [super viewDidLoad];
04.
05.     //字符串转整型
06.     NSString *str0 = @"1121";
07.     //NSString *str0 = @"中国";
08.
09.     //把字符串强转成整型
10.     int i = [str0 intValue];
11.
12.     NSLog(@"转换后: %i", i);
13.
14.     //字符串转integer
15.     NSString *str1 = @"1985";
16.     // NSString *str1 = @"中国";
17.
18.     //把字符串强转成integer
19.     NSInteger ii = [str1 integerValue];
20.
21.     NSLog(@"转换后: %i", ii);
22.
23.
24.
25.     //字符串转double
26.     NSString *str2 = @"3.145926";
27.     //NSString *str2 = @"中国";
28.
29.     //把字符串强转成double
30.     double d = [str2 doubleValue];
31.
32.     NSLog(@"转换后: %f", d);
33.
34.     //字符串转float
35.     NSString *str3 = @"3.145926";
36.     //NSString *str3 = @"中国";
37.     //把字符串强转成float
38.     double f = [str3 floatValue];
39.
40.     NSLog(@"转换后: %f", f);
41.
42. }
```

欢迎喜欢交流和热心的iphone开发朋友加入qq群参与讨论: 186739796, 验证码: csdn。

上一篇

Objective-C中的一些特殊的数据类型

下一篇

发邮件 In App Email (实例)

顶

0

踩

0

主题推荐

字符串

代码

实例

nsstring

objective-c

class

猜你在找

C/C++单元测试培训

QuickTest Professional深入剖析--【下部】

疯狂IOS讲义之Objective-C面向对象设计

C++语言基础

Cocos2d-x 实战演练基础篇

查看评论

2楼 [jxaufang168](#) 2013-08-27 17:13发表

C

宇神，好文章，多多指导

1楼 [xiongyoudou1](#) 2012-11-03 15:38发表

C

好东西

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 银行汇款帐号 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服


杂志客服

微博客服

webmaster@csdn.net

400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持

京 ICP 证 070598 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved



http://blog.csdn.net/likendsl/article/details/7417878

第 10 页 (共 10 页)