玩转 NSString

作者 LuisX (/u/6387bbe37e0a) ✓已关注
2016.06.02 11:28* 字数 267 阅读 2113 评论 7 喜欢 85
(/u/6387bbe37e0a)

玩转 NSString

> Objective-C里核心的处理字符串的类就是NSString和NSMutableString这两个类,这两个类完成了Objective-C中字符串大部分功能的处理

NSString

字符串属性

```
//////////字符串属性//////////
NSString *str1 = @"string";

NSLog(@"长度:                    %ld", str1.length);
NSLog(@"描述:                    %@", str1.description);
NSLog(@"哈希地址:                %lu", (unsigned long)str1.hash);
NSLog(@"字符串对应下标字符:       %c", [str1 characterAtIndex:2]);
//输出:   长度:                    6
//输出:   描述:                    string
//输出:   哈希地址:                10152471993823965
//输出:   字符串对应下标字符:       r
```

字符串截取

```
//截取指定位置之后的字符串
- (NSString *)substringFromIndex:(NSUInteger)from;
//从0下标开始截取到指定位置的字符串
- (NSString *)substringToIndex:(NSUInteger)to;
//截取指定范围的字符串
- (NSString *)substringWithRange:(NSRange)range;
//截取字符串指定范围的字符
- (void)getCharacters:(unichar *)buffer range:(NSRange)range;
```

字符串比较

> 将两个字符串进行计较,返回NSComparisonResult枚举结果

比较结果:

NSComparisonResult	说明
NSOrderedAscending	升序 (左小右大)
NSOrderedSame	相同 (内容相同)
NSOrderedDescending	降序 (左大右小)

比较选项:

NSStringCompareOptions	说明
NSCaseInsensitiveSearch	不区分大小写比较
NSLiteralSearch	区分大小写比较
NSBackwardsSearch	从字符串末尾开始搜索
NSAnchoredSearch	搜索限制范围的字符串
NSNumericSearch	按照字符串里的数字为依据，算出顺序
NSDiacriticInsensitiveSearch	忽略 "-" 符号的比较
NSWidthInsensitiveSearch	忽略字符串的长度，比较出结果
NSForcedOrderingSearch	忽略不区分大小写比较的选项
NSRegularExpressionSearch	只能应用于 rangeOfString:..., stringByReplacingOccurrencesOfString:...和 replaceOccurrencesOfString:...方法

```
   //////////字符串比较//////////
//1.比较
- (NSComparisonResult)compare:(NSString *)string;
//2.比较(指定字符串,条件)
- (NSComparisonResult)compare:(NSString *)string options:(NSStringCompareOptions)
mask;
//3.比较(指定字符串,条件,范围)
- (NSComparisonResult)compare:(NSString *)string options:(NSStringCompareOptions)
mask range:(NSRange)compareRange;
//4.比较(指定字符串,条件,范围,本地化)
- (NSComparisonResult)compare:(NSString *)string options:(NSStringCompareOptions)
mask range:(NSRange)compareRange locale:(nullable id)locale;
//5.比较(不区分大小写)
- (NSComparisonResult)caseInsensitiveCompare:(NSString *)string;

//6.本地化比较
- (NSComparisonResult)localizedCompare:(NSString *)string;
//7.本地化比较(不区分大小写)
- (NSComparisonResult)localizedCaseInsensitiveCompare:(NSString *)string;
//8.本地化比较(标准)
- (NSComparisonResult)localizedStandardCompare:(NSString *)string;

//9.判断两个字符串是否内容一致
- (BOOL)isEqualToString:(NSString *)aString;
```

字符串搜索

```
   //////////搜索基本/////////
//1.是否包含前缀(以str开头)
- (BOOL)hasPrefix:(NSString *)str;
//2.是否包含后缀(以str结尾)
- (BOOL)hasSuffix:(NSString *)str;
//3.获取公共前缀
- (NSString *)commonPrefixWithString:(NSString *)str options:(NSStringCompareOpti
ons)mask;
//4.是否包含字符串
- (BOOL)containsString:(NSString *)str;
//5.本地化是否包含字符串(不区分大小写)
- (BOOL)localizedCaseInsensitiveContainsString:(NSString *)str;
//6.本地化是否包含字符串(标准)
- (BOOL)localizedStandardContainsString:(NSString *)str;
//7.本地化搜索字符串范围(标准)
- (NSRange)localizedStandardRangeOfString:(NSString *)str;

   //////////搜索字符串/////////
//8.搜索(指定字符串)
- (NSRange)rangeOfString:(NSString *)searchString;
//9.搜索(指定字符串,条件)
- (NSRange)rangeOfString:(NSString *)searchString options:(NSStringCompareOptions
)mask;
//10.搜索(指定字符串,条件,范围)
- (NSRange)rangeOfString:(NSString *)searchString options:(NSStringCompareOptions
)mask range:(NSRange)searchRange;
//11.搜索(指定字符串,条件,范围,本地化)
- (NSRange)rangeOfString:(NSString *)searchString options:(NSStringCompareOptions
)mask range:(NSRange)searchRange locale:(nullable NSLocale *)locale;

   //////////搜索字符集合/////////
//12.搜索(指定字符集合)
- (NSRange)rangeOfCharacterFromSet:(NSCharacterSet *)searchSet;
//13.搜索(指定字符集合,条件)
- (NSRange)rangeOfCharacterFromSet:(NSCharacterSet *)searchSet options:(NSStringC
ompareOptions)mask;
//14.搜索(指定字符集合,条件,范围)
- (NSRange)rangeOfCharacterFromSet:(NSCharacterSet *)searchSet options:(NSStringC
ompareOptions)mask range:(NSRange)searchRange;

//15.用字符串的字符编码指定索引查找位置
- (NSRange)rangeOfComposedCharacterSequenceAtIndex:(NSUInteger)index;
//16.用字符串的字符编码指定区域段查找位置
- (NSRange)rangeOfComposedCharacterSequencesForRange:(NSRange)range;
```

字符串拼接

```
    //////////字符串拼接/////////
    NSString *string = @"1";
    NSString *appStr = @"2";
    NSString *resultStr1 = [string stringByAppendingString:appStr];
    NSString *resultStr2 = [string stringByAppendingFormat:@" + %@", appStr];

    NSLog(@"普通拼接:     %@", resultStr1);
    NSLog(@"格式化拼接:   %@", resultStr2);
    //输出:   普通拼接:     12
    //输出:   格式化拼接:   1 + 2
```

字符串基本类型转换

```
    //////////字符串基本类型转换/////////
    NSString *numStr = @"87234.2345";

    NSLog(@"double型:        %f", numStr.doubleValue);
    NSLog(@"float型:         %f", numStr.floatValue);
    NSLog(@"int型:           %d", numStr.intValue);
    NSLog(@"NSInteger型:     %ld", numStr.integerValue);
    NSLog(@"long long型:     %lld", numStr.longLongValue);
    NSLog(@"BOOL型:          %d", numStr.boolValue);
    //输出:   double型:        87234.234500
    //输出:   float型:         87234.234375
    //输出:   int型:           87234
    //输出:   NSInteger型:     87234
    //输出:   long long型:     87234
    //输出:   BOOL型:          1
```

字符串大小写转换

```
    //////////字符串大小写转换/////////
    NSString *string = @"string";

    NSLog(@"大写:         %@", string.uppercaseString);
    NSLog(@"小写:         %@", string.lowercaseString);
    NSLog(@"首字母大写:    %@", string.capitalizedString);
    //输出:   大写:         STRING
    //输出:   小写:         string
    //输出:   首字母大写:    String

//本地化(大写)
- (NSString *)uppercaseStringWithLocale:(nullable NSLocale *)locale;
//本地化(小写)
- (NSString *)lowercaseStringWithLocale:(nullable NSLocale *)locale;
//本地化(首字母大写)
- (NSString *)capitalizedStringWithLocale:(nullable NSLocale *)locale;
```

字符串分行,分段

```
   //////////分行/////////
//1.指定范围,分行取字符串
- (void)getLineStart:(nullable NSUInteger *)startPtr end:(nullable NSUInteger *)l
ineEndPtr contentsEnd:(nullable NSUInteger *)contentsEndPtr forRange:(NSRange)ran
ge;
//2.获取指定范围该行的范围
- (NSRange)lineRangeForRange:(NSRange)range;

   //////////分段/////////
//3.指定范围,分段取字符串
- (void)getParagraphStart:(nullable NSUInteger *)startPtr end:(nullable NSUIntege
r *)parEndPtr contentsEnd:(nullable NSUInteger *)contentsEndPtr forRange:(NSRange
)range;
//4.获取指定范围该段落的范围
- (NSRange)paragraphRangeForRange:(NSRange)range;

  //////////字符串分行,分段/////////
    NSString *string = @"123 456\nABC,DEF\nabc.def";
    //获取该行的范围(指定范围)
    NSRange range1 = [string lineRangeForRange:NSMakeRange(0, 10)];
    NSLog(@"%ld 行长度:     %ld", range1.location, range1.length);
    //输出:       0 行长度:     16

    //获取该段落范围(指定范围)
    NSRange range2 = [string paragraphRangeForRange:NSMakeRange(0, 3)];
    NSLog(@"%ld 段落长度:   %ld", range2.location, range2.length);
    //输出:       0 段落长度:   8
```

字符串列举(按条件)

列举选项:

NSStringEnumerationOptions	说明
NSStringEnumerationByLines	按行
NSStringEnumerationByParagraphs	按段落
NSStringEnumerationByComposedCharacterSequences	按字符顺序
NSStringEnumerationByWords	按单词,字
NSStringEnumerationBySentences	按句子
NSStringEnumerationReverse	反向遍历
NSStringEnumerationSubstringNotRequired	不需要子字符串
NSStringEnumerationLocalized	本地化

```
   //////////字符串列举//////////
   NSString *string = @"123456\nABCDEF\nabcdef
    //1.列举(按行)
   [string enumerateLinesUsingBlock:^(NSString * _Nonnull line, BOOL * _Nonnull s
top) {
       NSLog(@"每行:   %@", line);
   }];
    //输出:    每行:   123456
    //输出:    每行:   ABCDEF
    //输出:    每行:   abcdef


    //2.列举(范围,条件)
    [string enumerateSubstringsInRange:NSMakeRange(5, 10) options:NSStringEnumera
tionByLines usingBlock:^(NSString * _Nullable substring, NSRange substringRange,
NSRange enclosingRange, BOOL * _Nonnull stop) {

        NSLog(@"%@", substring);
        NSLog(@"%ld   %ld", substringRange.location, substringRange.length);
        NSLog(@"%ld   %ld", enclosingRange.location, enclosingRange.length);

    }];
    //输出:   6
    //输出:   5 1
    //输出:   5 2

    //输出:   ABCDEF
    //输出:   7   6
    //输出:   7   7

    //输出:   a
    //输出:   14   1
    //输出:   14   1
```

字符串编码和转换

```
    //////////字符串编码和转换//////////
    NSString *string = @"string";

    NSLog(@"最快编码值(枚举):      %lu", (unsigned long)string.fastestEncoding);
    NSLog(@"最小编码值(枚举):      %lu", (unsigned long)string.smallestEncoding);
    NSLog(@"UTF8编码值:            %s", string.UTF8String);
    //输出:     最快编码值(枚举):      1
    //输出:     最小编码值(枚举):      1
    //输出:     UTF8编码值:            string
```

```
//1.选择编码,是否允许有损编码
- (nullable NSData *)dataUsingEncoding:(NSStringEncoding)encoding allowLossyConve
rsion:(BOOL)lossy;
//2.选择编码
- (nullable NSData *)dataUsingEncoding:(NSStringEncoding)encoding;
//3.判断是否可以无损编码
- (BOOL)canBeConvertedToEncoding:(NSStringEncoding)encoding;

//4.C字符编码转换
- (nullable __strong const char *)cStringUsingEncoding:(NSStringEncoding)encoding
;
//5.判读C字符转化是否可以成功
- (BOOL)getCString:(char *)buffer maxLength:(NSUInteger)maxBufferCount encoding:(
NSStringEncoding)encoding;
//6.指定缓存区转换
- (BOOL)getBytes:(nullable void *)buffer maxLength:(NSUInteger)maxBufferCount use
dLength:(nullable NSUInteger *)usedBufferCount encoding:(NSStringEncoding)encodin
g options:(NSStringEncodingConversionOptions)options range:(NSRange)range remaini
ngRange:(nullable NSRangePointer)leftover;

//7.对字符串进行编码时所需的最大字节数
- (NSUInteger)maximumLengthOfBytesUsingEncoding:(NSStringEncoding)enc;
//8.对字符串进行编码时所需的字节数
- (NSUInteger)lengthOfBytesUsingEncoding:(NSStringEncoding)enc;

//9.可用字符串编码
+ (const NSStringEncoding *)availableStringEncodings;
//10.本地化编码名称
+ (NSString *)localizedNameOfStringEncoding:(NSStringEncoding)encoding;
//11.默认C字符串编码
+ (NSStringEncoding)defaultCStringEncoding;
```

字符串分割

字符集合:

NSCharacterSet	说明
controlCharacterSet	控制符
whitespaceCharacterSet	空格符
whitespaceAndNewlineCharacterSet	空格换行符
decimalDigitCharacterSet	小数
letterCharacterSet	文字
lowercaseLetterCharacterSet	小写字母
uppercaseLetterCharacterSet	大写字母
nonBaseCharacterSet	非基础
alphanumericCharacterSet	数字字母
decomposableCharacterSet	可分解
illegalCharacterSet	非法
punctuationCharacterSet	标点
capitalizedLetterCharacterSet	大写
symbolCharacterSet	符号
newlineCharacterSet	换行符

```
    //////////字符串分割//////////
    NSString *string = @"A_B_c_D_E_F";

    //分割(指定字符串)
    NSArray *resultArr1 = [string componentsSeparatedByString:@"_"];
    NSLog(@"%@", resultArr1);
    //输出:   (A, B, c, D, E, F)

    //分割(指定字符集合)
    NSArray *resultArr2 = [string componentsSeparatedByCharactersInSet:[NSCharact
erSet lowercaseLetterCharacterSet]];
    NSLog(@"%@", resultArr2);
    //输出:   ("A_B_" , "_D_E_F")
```

字符串操作(修剪,填充,折叠,替换)

```
    NSString *string = @"3EWRs a;af";

    //////////修剪/////////
    NSString *result1 = [string stringByTrimmingCharactersInSet:[NSCharacterSet l
owercaseLetterCharacterSet]];
    NSLog(@"%@", result1);
    //输出:   3EWRs a;


    //////////填充/////////
    NSString *result2 = [string stringByPaddingToLength:20 withString:@"填充" star
tingAtIndex:1];
    NSLog(@"%@", result2);
    //输出:   3EWRs a;af充填充填充填充填充填


    //////////折叠/////////
    NSString *result3 = [string stringByFoldingWithOptions:NSNumericSearch locale
:[NSLocale systemLocale]];
    NSLog(@"%@", result3);
    //输出:   3EWRs a;af


    //////////替换/////////
    //替换(指定字符串)
    NSString *result4 = [string stringByReplacingOccurrencesOfString:@" " withStr
ing:@"替换"];
    NSLog(@"%@", result4);
    //输出:   3EWRs替换a;af


    //替换(指定字符串,条件,范围)
    NSString *result5 = [string stringByReplacingOccurrencesOfString:@"a" withStr
ing:@"替换" options:NSRegularExpressionSearch range:NSMakeRange(0, string.length -
 2)];
    NSLog(@"%@", result5);
    //输出:   3EWRs 替换;af


    //替换(指定范围)
    NSString *result6 = [string stringByReplacingCharactersInRange:NSMakeRange(0,
 string.length) withString:@"替换"];
    NSLog(@"%@", result6);
    //输出:   替换
```

字符串翻译

翻译选项:

翻译选择	说明
NSStringTransformLatinToKatakana	拉丁->片假名
NSStringTransformLatinToHiragana	拉丁->平假名
NSStringTransformLatinToHangul	拉丁->韩语
NSStringTransformLatinToArabic	拉丁->阿拉伯语
NSStringTransformLatinToHebrew	拉丁->希伯来语
NSStringTransformLatinToThai	拉丁->泰国
NSStringTransformLatinToCyrillic	拉丁->西里尔字母
NSStringTransformLatinToGreek	拉丁->希腊
NSStringTransformToLatin	拉丁
NSStringTransformMandarinToLatin	普通话->拉丁
NSStringTransformHiraganaToKatakana	平假名->片假名
NSStringTransformFullwidthToHalfwidth	全角->半角
NSStringTransformToXMLHex	XML16进制
NSStringTransformToUnicodeName	Unicode名
NSStringTransformStripCombiningMarks	结合地带商标
NSStringTransformStripDiacritics	带音符

```
    //////////字符串翻译//////////
    NSString *dalian = @"大连";
    NSString *result = [dalian stringByApplyingTransform:NSStringTransformMandari
nToLatin reverse:NO];

    NSLog(@"%@", result);
    //输出:   dà lián
```

字符串写入

```
    //////////字符串写入//////////
    //写入到指定路径,编码的文件中
    [string writeToFile:@"/Users/Desktop/LuisX.txt" atomically:YES encoding:NSUTF
8StringEncoding error:nil];

    //写入到指定URL,编码的文件中
    [string writeToURL:[NSURL URLWithString:@"file://LuisX.text"] atomically:YES
encoding:NSUTF8StringEncoding error:nil];
```

NSMutableString

> 继承自NSString
> NSString中的方法在NSMutableString都可以使用
> NSMutableString是动态的字符串，可以动态的添加，修改，删除等

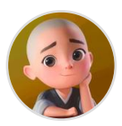
可变字符串分配容量

> 提前指定的容量，知识用于内存优化，实际大小可以大于设置的容量capacity

```
    //////////字符串分配容量//////////

    //1.初始化容量为Capacity大小的字符串  (需要手动释放内存)
    NSMutableString *mutableStr1 = [[NSMutableString alloc] initWithCapacity:20];

    //2.初始化容量为Capacity大小的字符串  (不需要手动释放内存)
    NSMutableString *mutableStr2 = [NSMutableString stringWithCapacity:20];
```

可变字符串扩展方法

```
NSMutableString *str = [NSMutableString stringWithString:@"string"];

//////////插入/////////
[str insertString:@"123" atIndex:2];
NSLog(@"%@", str);
//输出:   st123ring


//////////删除/////////
[str deleteCharactersInRange:NSMakeRange(2, 2)];
NSLog(@"%@", str);
//输出:   st3ring


//////////拼接/////////
[str appendString:@"456"];
NSLog(@"%@", str);
//输出:   st3ring456


//////////格式化拼接/////////
[str appendFormat:@"7 89"];
NSLog(@"%@", str);
//输出:   st3ring4567 89


//////////设置新字符串/////////
[str setString:@"string"];
NSLog(@"%@", str);
//输出:   string

//////////字符串替换/////////
//1.指定范围替换
[str1 replaceCharactersInRange:NSMakeRange(2, 2) withString:@"123"];
NSLog(@"%@", str1);
//输出:   st123ng


//2.指定字符串,条件,范围替换
[str1 replaceOccurrencesOfString:@"123" withString:@"--" options:NSRegularExp
ressionSearch range:NSMakeRange(0, str1.length)];
NSLog(@"%@", str1);
//输出:   st--ng
```

LuisX (/u/6387bbe37e0a)

写了 19783 字，被 279 人关注，获得了 574 个喜欢 (/u/6387bbe37e0a)

已关注

iOS开发工程师 不为圣贤，便为禽兽，莫问收获，但问耕耘

会跳舞的狮子 (/u/d0ac1ca15560)
6楼 · 2016.06.03 16:36
(/u/d0ac1ca15560)
写的不错,比我的还差点

1人赞 回复

> LuisX (/u/6387bbe37e0a)： @会跳舞的狮子 (/users/d0ac1ca15560) 整理而已,多谢大神指点
> 2016.06.03 17:06 回复
>
> 添加新评论

Unc1eWang (/u/7bcf2b8aec25)
2楼 · 2016.06.02 11:38
(/u/7bcf2b8aec25)
不错👍

赞 回复

舒大福 (/u/4612f27551d9)
3楼 · 2016.06.03 13:54
(/u/4612f27551d9)
整理的很好 谢谢

赞 回复

LongerONE (/u/e8da66f6193b)
4楼 · 2016.06.03 15:11
(/u/e8da66f6193b)
整理很全面。👍

赞 回复

昨夜雨轻栏 (/u/8d7b4aa671b3)
5楼 · 2016.06.03 15:13
(/u/8d7b4aa671b3)
mark.....

赞 回复

Joyce26 (/u/8cb3a0630c6e)
7楼 · 2016.10.26 16:28
(/u/8cb3a0630c6e)
我看完了

赞 回复