

个人资料



ios\_long

访问： 31264次  
积分： 373分  
排名： 千里之外

原创： 11篇    转载： 13篇  
译文： 0篇    评论： 10条

文章搜索

文章分类

[语法 \(19\)](#)  
[第三方插件 \(6\)](#)  
[技巧总结 \(0\)](#)

文章存档

[2013年03月 \(2\)](#)  
[2012年12月 \(2\)](#)  
[2012年06月 \(2\)](#)  
[2012年02月 \(2\)](#)  
[2012年01月 \(1\)](#)

展开

阅读排行

[ios 正则表达式](#) (16858)  
[FFMPEG同步操作（转）](#) (1756)  
[FFMPEG编译](#) (1242)  
[mediafilesegmenter 分段](#) (1193)  
[iPhone半透明状态栏与号](#) (1192)  
[C语言指针传递和内存分](#) (1144)  
[ios线程监听者模式](#) (899)  
[ios中查询文件大小。](#) (781)  
[//启动画面为横屏！！](#) (675)  
[iPhone开发中的内存管理](#) (392)

更多相关资源： 正则表达式 ios

[7月推荐文章](#)    [【限时活动】建专辑得大奖](#)    [专访张路斌：从HTML5到Unity的游戏开发之路](#)    [当青春遇上互联网，能否点燃你的创业梦](#)    [推荐有礼--找出您心中的技术大牛](#)

ios 正则表达式

分类： 第三方插件    2011-08-26 09:46    17379人阅读    评论(6)    收藏    举报

[ios](#)    [正则表达式](#)    [null](#)    [email](#)    [frameworks](#)    [string](#)

1.去RegexKitLite下载类库，解压出来会有一个例子包及2个文件，其实用到的就这2个文件，添加到工程中。备用地址：  
<http://www.cocoachina.com/bbs/job.php?action-download-pid-135286-tid-18111-aid-11143.html>    - Lv

2.工程中添加libicucore.dylib frameworks。

3.现在所有的NSString对象就可以调用RegexKitLite中的方法了。

```
NSString *email = @"kkk@aaa.com";

[email isMatchedByRegex:@"\\b([a-zA-Z0-9%_+\\-]+)@([a-zA-Z0-9\\.\\-]+?\\.[a-zA-Z]{2,6})\\b"];

返回YES，证明是email格式，需要注意的是RegexKitLite用到的正则表达式和wiki上的略有区别。

searchString = @"http://www.example.com:8080/index.html" ;

regexString = @"\\bhttps?://[a-zA-Z0-9\\-]+(?:\\.(\\d+))?(?:\\.[a-zA-Z0-9\\-._?'+\\&%$=~!*()@:\\|\\]|)+)?";

NSInteger portInteger = [[searchString stringByMatching:regexString capture:1L] integerValue];

NSLog(@"portInteger: %ld", (long)portInteger);

// 2008-10-15 08:52:52.500 host_port[8021:807] portInteger: '8080'

取string中http的例子。
```

常用正则表达式：

匹配双字节字符(包括汉字在内)： [^x00-xff]

评注：可以用来计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）

匹配空白行的正则表达式： ns\*r

评注：可以用来删除空白行

匹配HTML标记的正则表达式： <(S\*?)[^>]\*.\*?<.\*? />

评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力

匹配首尾空白字符的正则表达式： ^s\*|s\*\$

评注：可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式

匹配Email地址的正则表达式： w+([+.]w+)\*@w+([+.]w+)\*.w+([+.]w+)\*

评注：表单验证时很实用

匹配网址URL的正则表达式： [a-zA-Z]+://([s]\*)

评注：网上流传的版本功能很有限，上面这个基本可以满足需求

匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)： ^[a-zA-Z][a-zA-Z0-9\_]{4,15}\$

评注：表单验证时很实用

匹配国内电话号码： d{3}-d{8}|d{4}-d{7}

评注：匹配形式如 0511-4405222 或 021-87888822

匹配腾讯QQ号： [1-9][0-9]{4,}

评注：腾讯QQ号从10000开始

匹配中国邮政编码： [1-9]d{5}(?!d)

评注：中国邮政编码为6位数字

匹配身份证： d{15}|d{18}

评注：中国的身份证为15位或18位

匹配ip地址： d+.d+.d+.d+

评注：提取ip地址时有用

匹配特定数字：

```
^[1-9]d*$      //匹配正整数
^-?[1-9]d*$    //匹配负整数
^-?[1-9]d*$    //匹配整数
^[1-9]d*I0$    //匹配非负整数（正整数 + 0）
^-?[1-9]d*I0$  //匹配非正整数（负整数 + 0）
^[1-9]d*.d*I0.d*[1-9]d*$  //匹配正浮点数
```

经典资源下载一周推荐

- 1、ACM学习：历年国家集训队论文集
- 2、51套经典网站源码分享
- 3、10个Azure开发技术实例
- 4、GDI+程序设计资源整合
- 5、7款jQuery经典插件
- 6、学校管理系统数据库设计资源
- 7、高校ACM课件2014版
- 8、C/C++编程规范

评论排行

- ios 正则表达式 (6)
- C语言指针传递和内存分 (1)
- copy 和 retain (1)
- FFMPEG同步操作 (转) (1)
- 来源于 教程 分类 C++和 (1)
- iPhone半透明状态栏与号 (0)
- 遍历字符串应该取出字符 (0)
- 新建多级目录 (0)
- //启动画面为横屏！！ (0)
- 使用Notifications（各种 (0)

推荐文章

- \* android 动画分类
- \* 全面理解Android Navigation 逻辑
- \* iOS开发 - UINavigationController详解
- \* 一个未完成创业项目的思考——创业杂记
- \* Android系统开发（1）——GCC编译器的编译和安装过程
- \* Android学习路线（二十m） ActionBar Fragment运用

最新评论

- ios 正则表达式 keqi90: 很好！收藏了~
- FFMPEG同步操作 (转) mophon: 不错！
- ios 正则表达式 mrtianxiang: 学习了
- ios 正则表达式 zyzq527758142: 很好！
- ios 正则表达式 leewolf: 不错啊
- copy 和 retain \_\_Hadley: copy的理解完全错了
- ios 正则表达式 ALDRIDGE1: 给力，顶
- ios 正则表达式 wang0731308: 详细且使用，给力！
- C语言指针传递和内存分配 micro\_xzq: 学习了，以前真不懂呀。。。
- 来源于 教程 分类 C++和Objectiv lihonglin2016: 讲的不错

^~?([1-9]d\*.d\*10.d\*[1-9]d\*10?.0+10)\$ //匹配浮点数  
^1-9]d\*.d\*10.d\*[1-9]d\*10?.0+10\$ //匹配非负浮点数（正浮点数 + 0）  
^~([1-9]d\*.d\*10.d\*[1-9]d\*)10?.0+10\$ //匹配非正浮点数（负浮点数 + 0）  
评注：处理大量数据时有用，具体应用时注意修正  
匹配特定字符串：  
^[A-Za-z]+\$ //匹配由26个英文字母组成的字符串  
^[A-Z]+\$ //匹配由26个英文字母的大写组成的字符串  
^[a-z]+\$ //匹配由26个英文字母的小写组成的字符串  
^[A-Za-z0-9]+\$ //匹配由数字和26个英文字母组成的字符串  
^w+\$ //匹配由数字、26个英文字母或者下划线组成的字符串  
在使用RegularExpressionValidator验证控件时的验证功能及其验证表达式介绍如下：  
只能输入数字：“^[0-9]\*\$”  
只能输入n位的数字：“^d(n)\$”  
只能输入至少n位的数字：“^d(n,)\$”  
只能输入m-n位的数字：“^d(m,n)\$”  
只能输入零和非零开头的数字：“^0|[1-9][0-9]\*\$”  
只能输入有两位小数的正实数：“^0-9]+([0-9]{2})?\$”  
只能输入有1-3位小数的正实数：“^0-9]+([0-9]{1,3})?\$”  
只能输入非零的正整数：“^+?[1-9][0-9]\*\$”  
只能输入非零的负整数：“^-1-9[0-9]\*\$”  
只能输入长度为3的字符：“^.{3}\$”  
只能输入由26个英文字母组成的字符串：“^[A-Za-z]+\$”  
只能输入由26个大写英文字母组成的字符串：“^[A-Z]+\$”  
只能输入由26个小写英文字母组成的字符串：“^[a-z]+\$”  
只能输入由数字和26个英文字母组成的字符串：“^[A-Za-z0-9]+\$”  
只能输入由数字、26个英文字母或者下划线组成的字符串：“^w+\$”  
验证用户密码：“^[a-zA-Z]w(5,17)\$”正确格式为：以字母开头，长度在6-18之间，只能包含字符、数字和下划线。  
验证是否含有^%&';,=?\$"等字符：“^[^%&';,=?\$x22]+\$”  
只能输入汉字：“^[u4e00-u9fa5];{0,}\$”  
验证Email地址：“^w+([-.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*\$”  
验证InternetURL：“^http://([w-]+[w-]+(/[w-./?%&=])?)?\$”  
验证电话号码：“^((d(3,4))d(3,4)-)?d(7,8)\$”  
正确格式为：“XXXX-XXXXXXX”，“XXXX-XXXXXXX”，“XXX-XXXXXXX”，“XXX-XXXXXXX”，“XXXXXXX”，“XXXXXXX”，“XXXXXXX”。  
验证身份证号（15位或18位数字）：“^d(15)d{18}\$”  
验证一年的12个月：“^0?[1-9]1[0-2]\$”正确格式为：“01”-“09”和“1”“12”  
验证一个月的31天：“^((0?[1-9])|((112)[0-9])30131)\$”  
正确格式为：“01”“09”和“1”“31”。  
匹配中文字符的正则表达式：[u4e00-u9fa5]  
匹配双字节字符(包括汉字在内)：[^\x00-xff]  
匹配空行的正则表达式：\n\s\*\r  
匹配HTML标记的正则表达式：/(<.\*>.\*<.\*>)/>  
匹配首尾空格的正则表达式：(^s\*)(s\*)  
匹配Email地址的正则表达式：w+([-.]w+)\*@w+([-.]w+)\*.w+([-.]w+)\*  
匹配网址URL的正则表达式：[url=http://%2B [w-]+.%29+[w-]+%2B/[w]http://([w-]+)+[w-]+(/[w/]url/.?/%&=]\*)?

基本使用的例子（更多信息参看 官方文档 ）

```
1.

NSString *searchString = @"This is neat." ;
NSString *regexString = @"(//w+)//s+((//w+)//s+((//w+))" ;
NSRange matchedRange = NSMakeRange(NSNotFound, 0UL);
NSError *error = NULL;

matchedRange = [searchString rangeOfRegex:regexString options:RKLNoOptions inRange:searchRange capture :2L error:&error];
NSLog(@"matchedRange: %@", NSStringFromRange(matchedRange));
// 2008-03-18 03:51:16.530 test[51583:813] matchedRange: {5, 2}. //匹配到'is'
NSString *matchedString = [searchString substringWithRange:matchedRange];
NSLog(@"matchedString: '%@'", matchedString);
// 2008-03-18 03:51:16.532 test[51583:813] matchedString: 'is' //生成子字符串

NSString *searchString = @"This is neat.";
NSString *regexString = @"(//w+)//s+((//w+)//s+((//w+))";
NSRange matchedRange = NSMakeRange(NSNotFound, 0UL);
NSError *error = NULL;

matchedRange = [searchString rangeOfRegex:regexString options:RKLNoOptions inRange:searchRange capture :2L error:&error];
NSLog(@"matchedRange: %@", NSStringFromRange(matchedRange));
// 2008-03-18 03:51:16.530 test[51583:813] matchedRange: {5, 2}. //匹配到'is'
NSString *matchedString = [searchString substringWithRange:matchedRange];
NSLog(@"matchedString: '%@'", matchedString);
// 2008-03-18 03:51:16.532 test[51583:813] matchedString: 'is' //生成子字符串
```

- 1、ACM学习：历年国家集训队论文集
- 2、51套经典网站源码分享
- 3、10个Azure开发技术实例
- 4、GDI+程序设计资源整合
- 5、7款jQuery经典插件
- 6、学校管理系统数据库设计资源
- 7、高校ACM课件2014版
- 8、C/C++编程规范

## 2.找到第一个匹配并返回一个NSString

```
NSString *searchString = @ "This is neat." ;
NSString *regexString = @"(/w+//s+(/w+//s+(/w+))";
NSString *matchedString = [searchString stringByMatching:regexString capture:2L];
NSLog(@"matchedString: %@", matchedString);
// 2008-03-18 03:53:42.949 test[51583:813] matchedString: 'is'

NSString *searchString = @"This is neat.";
NSString *regexString = @"(/w+//s+(/w+//s+(/w+))";
NSString *matchedString = [searchString stringByMatching:regexString capture:2L];
NSLog(@"matchedString: %@", matchedString);
// 2008-03-18 03:53:42.949 test[51583:813] matchedString: 'is'
```

## 3.查找和替换, 加括号和概念和Python中的一样, \$1指代第一个括号中的内容

```
NSString *searchString = @ "This is neat." ;
NSString *regexString = @"(/b(/w+//b)";
NSString *replaceWithString = @"{$1}";
NSString *replacedString = NULL;
replacedString = [searchString stringByReplacingOccurrencesOfRegex:regexString withString:replaceWithString];
//NSMutableString可以直接替换, 并返回替换的次数
NSLog(@"replaced string: %@", replacedString);
// 2008-07-01 19:03:03.195 test[68775:813] replaced string: '{This} {is} {neat}.'
NSMutableString *mutableString = [NSMutableString stringWithString:@"This is neat." ];
NSString *regexString = @"(/b(/w+//b)";
NSString *replaceWithString = @"{$1}";
NSUInteger replacedCount = 0UL;
replacedCount = [mutableString replaceOccurrencesOfRegex:regexString withString:replaceWithString];
NSLog(@"count: %lu string: %@", (u_long)replacedCount, mutableString);
// 2008-07-01 21:25:43.433 test[69689:813] count: 3 string: '{This} {is} {neat}.'

NSString *searchString = @"This is neat.";
NSString *regexString = @"(/b(/w+//b)";
NSString *replaceWithString = @"{$1}";
NSString *replacedString = NULL;
replacedString = [searchString stringByReplacingOccurrencesOfRegex:regexString withString:replaceWithString];
//NSMutableString可以直接替换, 并返回替换的次数
NSLog(@"replaced string: %@", replacedString);
// 2008-07-01 19:03:03.195 test[68775:813] replaced string: '{This} {is} {neat}.'
NSMutableString *mutableString = [NSMutableString stringWithString:@"This is neat." ];
NSString *regexString = @"(/b(/w+//b)";
NSString *replaceWithString = @"{$1}";
NSUInteger replacedCount = 0UL;
replacedCount = [mutableString replaceOccurrencesOfRegex:regexString withString:replaceWithString];
NSLog(@"count: %lu string: %@", (u_long)replacedCount, mutableString);
// 2008-07-01 21:25:43.433 test[69689:813] count: 3 string: '{This} {is} {neat}.'
```

## 4.用于拆分, 返回一个拆分后的字符串数组

```
NSString *searchString = @ "This is neat." ;
NSString *regexString = @"//s+";
NSArray *splitArray = NULL;
splitArray = [searchString componentsSeparatedByRegex:regexString];
// splitArray == { @"This", @"is", @"neat." }
NSLog(@"splitArray: %@", splitArray);

NSString *searchString = @"This is neat.";
NSString *regexString = @"//s+";
NSArray *splitArray = NULL;
splitArray = [searchString componentsSeparatedByRegex:regexString];
// splitArray == { @"This", @"is", @"neat." }
NSLog(@"splitArray: %@", splitArray);
```

[经典资源下载一周推荐](#)

- 1、ACM学习: 历年国家集训队论文集
- 2、51套经典网站源码分享
- 3、10个Azure开发技术实例
- 4、GDI+程序设计资源整合
- 5、7款jQuery经典插件
- 6、学校管理系统数据库设计资源
- 7、高校ACM课件2014版
- 8、C/C++编程规范

[更多相关资源:](#) 正则表达式 ios

5. 返回所有匹配的字符串数组，这个例子中虽然有多个括号，但是 **componentsMatchedByRegex** 不管

```
NSString *searchString = @"$10.23, $1024.42, $3099" ;
NSString *regexString = @"//$(//d+)(?://.(/d+)|//.?)" ;
NSArray *matchArray = NULL;
matchArray = [searchString componentsMatchedByRegex:regexString];
// matchArray == { @"$10.23", @"$1024.42", @"$3099" };
NSLog(@"matchArray: %@", matchArray);
6. 返回所有匹配的字符串数组处理所有的括号
NSString *searchString = @"$10.23, $1024.42, $3099" ;
NSString *regexString = @"//$(//d+)(?://.(/d+)|//.?)" ;
NSArray *capturesArray = NULL;
capturesArray = [searchString arrayOfCaptureComponentsMatchedByRegex:regexString];
/* capturesArray ==
[NSArray arrayWithObjects:
 [NSArray arrayWithObjects: @"$10.23", @"10.23", @"10", @"23", NULL],
 [NSArray arrayWithObjects:@"$1024.42", @"1024.42", @"1024", @"42", NULL],
 [NSArray arrayWithObjects: @"$3099", @"3099", @"3099", @"", NULL],
 NULL];
*/
NSLog(@"capturesArray: %@", capturesArray);
```

输出结果:

```
shell% ./capturesArray
```

```
2009-05-06 03:25:46.852 capturesArray[69981:10b] capturesArray: (
    (
        "$10.23" ,
        "10.23" ,
        10,
        23
    ),
    (
        "$1024.42" ,
        "1024.42" ,
        1024,
        42
    ),
    (
        "$3099" ,
        3099,
        3099,
        ""
    )
)
```

```
NSString *searchString = @"$10.23, $1024.42, $3099";
NSString *regexString = @"//$(//d+)(?://.(/d+)|//.?)";
NSArray *matchArray = NULL;
matchArray = [searchString componentsMatchedByRegex:regexString];
// matchArray == { @"$10.23", @"$1024.42", @"$3099" };
NSLog(@"matchArray: %@", matchArray);
6. 返回所有匹配的字符串数组处理所有的括号
NSString *searchString = @"$10.23, $1024.42, $3099";
NSString *regexString = @"//$(//d+)(?://.(/d+)|//.?)";
NSArray *capturesArray = NULL;
capturesArray = [searchString arrayOfCaptureComponentsMatchedByRegex:regexString];
/* capturesArray ==
[NSArray arrayWithObjects:
 [NSArray arrayWithObjects: @"$10.23", @"10.23", @"10", @"23", NULL],
 [NSArray arrayWithObjects:@"$1024.42", @"1024.42", @"1024", @"42", NULL],
 [NSArray arrayWithObjects: @"$3099", @"3099", @"3099", @"", NULL],
 NULL];
*/
NSLog(@"capturesArray: %@", capturesArray);
```

输出结果:

```
shell% ./capturesArray
```

```
2009-05-06 03:25:46.852 capturesArray[69981:10b] capturesArray: (
    (
        "$10.23" ,
        "10.23" ,
        10,
        23
    ),
    (
        "$1024.42" ,
        "1024.42" ,
        1024,
        42
    ),
    (
        "$3099" ,
        3099,
        3099,
        ""
    )
)
```

[经典资源 下载一周推荐](#)

- 1、ACM学习：历年国家集训队论文集
- 2、51套经典网站源码分享
- 3、10个Azure开发技术实例
- 4、GDI+程序设计资源整合
- 5、7款jQuery经典插件
- 6、学校管理系统数据库设计资源
- 7、高校ACM课件2014版
- 8、C/C++编程规范

更多相关资源: [正则表达式](#) [ios](#)

```
(
    "$3099",
    3099,
    3099,
    ""
)
)
```

下一篇 ios线程监听者模式

主题推荐 正则表达式 ios frameworks 中国 download

猜你在找

- iPhone之实现自定义进度条Progress
- IOS CoreText.framework 一行 CTLineRef
- IOS类似9.png
- iOS安全攻防（八）：键盘缓存与安全键盘
- CopyWithZone:关于深拷贝/浅拷贝
- IOSBase64和3DES加密
- iPhone开发【十七】多线程开发之
- 判断特殊字符的正则表达式
- 关于uINavigationController跳转时，若跳转到的页面
- IOS开发笔记（7）基本控件

Android开发培训-Mars主讲

mobiletrain.org  
千锋3G学院,Mars老师带你学android 欢迎免费咨询

查看评论

6楼 keqi90 2014-07-03 21:41发表

很好! 收藏了~

5楼 mrtianxiang 2014-05-12 16:32发表

学习了

4楼 zyg527758142 2014-03-10 17:39发表

很好!

3楼 leewolf 2013-08-13 09:21发表

不错啊

2楼 ALDRIDGE1 2013-04-10 08:59发表

给力，顶

1楼 wang0731308 2013-03-15 14:31发表

详细且使用，给力！

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

经典资源 下载一周推荐

核心技术类目

- 全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 BI HTML5 Spring Apache .NET API HTML SDK IIS Splashtop UML components Windows Mobile Rails QEMU KDE coremail OPhone CouchBase 云计算 iOS6 Rackspace Web Ap Compuware 大数据 aptech Perl Tornado Ruby Hibernate Ti Angular Cloud Foundry Redis Scala Django Bootstrap
- 1、ACM学习：历年国家集训队论文集
  - 2、51套经典网站源码分享
  - 3、10个Azure开发技术实例
  - 4、GDI+程序设计资源整合
  - 5、7款jQuery经典插件
  - 6、学校管理系统数据库设计资源
  - 7、高校ACM课件2014版
  - 8、C/C++编程规范

更多相关资源: 正则表达式 ios

经典资源 下载一周推荐 

- 1、ACM学习：历年国家集训队论文集
- 2、51套经典网站源码分享
- 3、10个Azure开发技术实例
- 4、GDI+程序设计资源整合
- 5、7款jQuery经典插件
- 6、学校管理系统数据库设计资源
- 7、高校ACM课件2014版
- 8、C/C++编程规范

更多相关资源:

正则表达式

ios