



MacPen

发布时间：2020-08-23 18:50

OC 正则表达式的语法及使用

正则表达式就是可以分段进行匹配的语法规则，判断字符串中的每一个片段是否能够相应的内容规则匹配。

基本语法：

1.一般字符，除了元字符外的字符都是一般字符，匹配的就是字符的字面意思。

2.元字符，特殊字符，不是匹配的字符字面量。

- . 表示除换行符以外的任意字符；
- * 表示它前面的那个表达式重复0次或若干次；
eg:“ab*”表示“a”或“ab”或“abb”...
- + 表示它前面的那个表达式重复至少1次；
eg:“ab+”表示“ab”或“abb”或“abbb”...
- ? 表示它前面的那个表达式存在或不存在；
eg:“ab?”表示“a”或“ab”
- ^ 表示以它后面的表达式开头的字符串；
eg:“^abc”表示以“abc”开头的字符串
- \$ 表示以它前面的表达式结尾的字符串；
eg:“abc\$”表示以“abc”结尾的字符串
- [] 表示内容的范围；
eg:“[a-zA-Z]”表示任一字母； “[ab]”表示“a”或“b”
- { } 表示数量的范围；
eg:“{2}”表示2个； “[2,5]”表示2个到5个； “[{3,}”表示至少3个；
- | 表示 或；
eg:“a|b”表示“a”或“b”
- \num num为一个整数，表示前面的表达式复制num次；
eg:“(ab){3}”表示“ababab”
- \b 表示单词的开始或结束；
- \d 表示数字，相当于“[0-9]”；
- \s 表示任意空白符；
- \w 表示任意字符数字或下划线或汉字，相当于“[a-zA-Z0-9\u4E00-\u9FFF_]”

3.字符转义

- \ 表示字符转义，如果我们要使用元字符的字面意思的话，就需要转义字符进行字符转义。
eg:“\.”表示“.”； “*”表示“*”； “\|”表示“|”；

4.反义字符

- \W 表示不是字符数字下划线或者汉字的字符；
- \S 表示任意不是空白符的字符；
- \D 表示不是数字的字符；
- \B 表示不是单词开头或结束的位置；
- [^x] 表示除“x”外的任意字符；
- [^abc] 表示除“a”“b”“c”外的任意字符；

OC中正则表达式的写法又有些微的不同，使用转义字符作用在元字符 (. { | () [{ ^ ? * + |) 上的时候要在转义字符前再加“\”。eg:

正常规则下，“.”表示换行符以外的任意字符，“\.”表示“.”;在OC中，“.”依然表示换行符以外的任意字符，“\.”表示“.”。简单点理解就是按正常的规则先写表达式，然后把写好的表达式中的所有“.”都替换为“\.”即可。

更多正则表达式的语法可参考<http://deerchao.net/tutorials/regex/regex.htm>

几个实例：

邮箱格式验证：

```
- (BOOL)validateEmail:(NSString *)email {
    NSString *aimStr = [email stringByReplacingOccurrencesOfString:@" " withString:@""];
    NSString *regex = @"^[a-zA-Z0-9]\\w*@([a-zA-Z0-9]+\\.([a-zA-Z]{2,4})$)";
    NSPredicate *emailValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", regex];
    return [emailValidate evaluateWithObject:email];
}
```

手机号码验证

```
//联通号段：130 131 132 145 146 155 156 166 171 175 176 185 186
//移动号段：134 135 136 137 138 139 147 148 150 151 152 157 158 159 172 178 182 183 184 187 188 198
//电信号段：133 149 153 173 174 177 180 181 189 199
- (BOOL)validatePhone:(NSString *)phone {
    NSString *aimStr = [phone stringByReplacingOccurrencesOfString:@" " withString:@""];
    NSString *Curegex = @"^1(3[012])|4[56]|5[56]|66|7[156]|8[56])|0-9]{8}$";
    NSString *Mregex = @"^1(3[4-9]|4[78]|5[012789]|7[28]|8[23478]|98)|0-9]{8}$";
    NSString *Tregex = @"^1(33|49|53|7[347]|8[019]|99)|0-9]{8}$";
    NSPredicate *CemailValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", Curegex];
    NSPredicate *CemailValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", CMregex];
    NSPredicate *CTemailValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", CTregex];
    if ([CemailValidate evaluateWithObject:aimStr] || [CemailValidate evaluateWithObject:aimStr] || [CTemailValidate evaluateWithObject:aimStr]) {
        return YES;
    }
    return NO;
}
```

车牌号验证

```
//除了原有的五位车牌号之外，增加两种新能源汽车车牌号，小型家庭用车：在城市码之后加D或F，再加五位编码；大型汽车：城市码之后是五位编码，再加D或F；D和F分别表示纯电动车和混合动力汽车。
- (BOOL)validateCarNumber:(NSString *)carNo {
    NSString *aimStr = [carNo stringByReplacingOccurrencesOfString:@" " withString:@""];
    NSString *regex = @"^[\\u4E00-\\u9FFF][a-zA-Z]([DdFf][a-zA-Z0-9]{5}|[A-a-z0-9]{5})|([a-zA-Z0-9]{5})([DdFf])$";
    NSPredicate *carNoValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", regex];
    return [carNoValidate evaluateWithObject:aimStr];
}
```

身份证号验证

```
//18位的身份证号前十七位是本体码，最后一个是校验码（0-10），10用罗马数字x或X代替
//使用身份证的校验方法和校验码可以判定身份证的真假，校验方法如下：
//1.7-9-10-5-8-4-2-1-6-3-7-9-10-5-8-4-2这十七个数分别和身份证号的前十七位数字相乘，然后将结果相加
//2.将上一步的计算结果除以11，求余数（0到10之间）
//3.上一步中的余数从0到10分别对应1-0-X-9-8-7-6-5-4-3-2，也就是身份证号的最后一位，如果不符合就是假的
- (BOOL)validatePersonalID:(NSString *)personID {
    NSString *aimStr = [personID stringByReplacingOccurrencesOfString:@" " withString:@""];
    NSString *regex = @"^[1-6][0-7]0[0-9]{14}[0-9Xx]$";
    NSPredicate *personValidate = [NSPredicate predicateWithFormat:@"SELF MATCHES %@", regex];
    if (![personValidate evaluateWithObject:aimStr]) {
        return NO;
    }
    } else {
        NSArray *modulusArr = @[7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2];
        NSArray *referArr = @[1,0,10,9,8,7,6,5,4,3,2];
        NSInteger sum = 0;
        for (int i = 0; i
```

银行卡号验证

```
//国内的银行卡卡号基本上是16位或者19位，几乎所有的卡号都满足Luhn算法规则，所以银行卡的验证就不使用正则表达式了
//银行卡号编码规则及其应用https://zhuanlan.zhihu.com/p/21399490
//根据BIN码查银行卡种http://blog.csdn.net/archer119/article/details/52081046
- (BOOL)validateBankNo:(NSString *)bankNo {
    NSString *aimStr = [bankNo stringByReplacingOccurrencesOfString:@" " withString:@""];
    if (aimStr.length == 0) {
        return NO;
    }
    } else {
        NSMutableArray *noArr = [NSMutableArray array];
        int sum = 0;
        for (NSInteger i = aimStr.length-1; i>=0; i--) {
            int tempNo = [[aimStr substringWithRange:NSMakeRange(i, 1)] intValue];
            if ((aimStr.length-i-1)%2 == 0) {
                [noArr addObject:@(tempNo)];
            } else {
                [noArr addObject:tempNo*2>9 ? @(tempNo*2-9) : @(tempNo*2)];
            }
        }
        for (NSNumber *number in noArr) {
            sum += (int)number;
        }
        return sum%10==0;
    }
}
```

你可能感兴趣的

- [LeetCode\[位运算\] - #137 Single Number II](#) Cwind [Java](#) [Algorithm](#) [LeetCode](#) [题解](#) [位运算](#)
- [《JavaScript语言精粹》笔记](#) aijuans [JavaScript](#)
- [你应该更新的Java知识之常用程序库](#) Kai_Ge [java](#)
- [HttpClient](#) 120153216 [httpClient](#)
- [Django model字段类型清单](#) 2002wmj [django](#)
- [在SQLSERVER中查找消耗CPU最多的SQL](#) 357029540 [SQL Server](#)
- [Myeclipse项目无法部署，Undefined exploded archive location](#) 7454103 [eclipse](#) [MyEclipse](#)
- [GMT时间格式转换](#) adminjun [GMT](#) [时间转换](#)

按字母分类： A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 其他