6月进入前端技术的学习，正则算是跳不过的一个坎了，这部分没有太多需要理解的内容，知道就是王道。



正则表达式（Regular Expression）:在代码中常简写为regex，使用单个[字符](http://baike.baidu.com/view/263416.htm)串来描述、匹配一系列符合某个句法规则。在很多[文本编辑器](http://baike.baidu.com/view/487023.htm)里，正则表达式通常被用来检索、替换那些符合某个模式的文本，常见的正则字符如下表所示。

|  |  |
| --- | --- |
| 元字符 | 描述 |
| \ | 转义符。例如，“\\”匹配\。“\n”匹配换行符。 |
| ^或\A | 匹配输入字符串的开始位置。 |
| $或\Z | 匹配输入字符串的结束位置。 |
| \* | 匹配前面的子表达式任意次。 |
| + | 匹配前面的子表达式一次或多次(大于等于1次）。 |
| ? | 匹配前面的子表达式零次或一次。 |
| . | 匹配除了换行符以外的任意一个字符 |
| {m} | 匹配前一个内容，重复次数是m次 |
| {m, } | 匹配前一个内容，重复次数大于等于m次 |
| {m, n} | 匹配前一个内容，重复次数m次到n次 |
| [] | 匹配方括号中所包含的任意一个字符，例如[abc] |
| [a-z] | 字符范围，所有小写字母字符。 |
| [^] | 匹配方括号中原子以外的任意字符 |
| \b | 匹配单词的边界 |
| \B | 匹配单词边界意外的部分 |
| \d | 匹配一个数字字符。 |
| \D | 匹配一个非数字字符。 |
| 0 | 匹配其整体为一个原子，即模式单元 |

|  |  |
| --- | --- |
| 作用 | 示例 |
| 非负整数 | ^\d+$ |
| 正整数 | ^[0-9]\*[1-9][0-9]\*$ |
| 非正整数 | ^((-\d+)|(0+))$ |
| 负整数 | ^-[0-9]\*[1-9][0-9]\*$ |
| 英文字符串 | ^[A-Za-z]+$ |
| 英文字符数字串 | ^[A-Za-z0-9]+$ |
| 英文数字加下划线串 | ^\w+$ |
| E-mail地址 | ^[\w-]+(\.[\w-]+)\*@[\w-]+(\.[\w-]+)+$ |
| URL | ^[a-zA-Z]+://(\w+(-\w+)\*)(\.(\w+(-\w+)\*))\*(\?\s\*)?$ |

此外，还可以随时到百度查阅或查看站长工具：<http://tool.chinaz.com/regex/>

最近浏览了不少语言，包括python（1989）,ruby,perl,php等，大体上来看，都不太适合现在的web开发节奏，基本可以放弃了，python可以作为一个不错的辅助工具。而传说中的ruby on rails（2004）虽然很不错，但ruby却不是为互联网而生的语言。最终决定，web这部分学个nodejs就好，2009为互联年而生的boy，加油。其他的在巩固C#和Java的前提，了解一下与docker有关的Go语言即可。至于swift之类的，有空再说吧，最后借用闫恒敏大神的一首帅帅的打油诗，加强正则表达式的记忆。

|  |
| --- |
| 正则其实也势利，削尖头来把钱揣； （指开始符号^和结尾符号$）  特殊符号认不了，弄个倒杠来引路； （指\. \\*等特殊符号）  倒杠后面跟小w， 数字字母来表示； （\w跟数字字母;\d跟数字）  倒杠后面跟小d， 只有数字来表示；  倒杠后面跟小a， 报警符号嘀一声；  倒杠后面跟小b， 单词分界或退格；  倒杠后面跟小t， 制表符号很明了；  倒杠后面跟小r， 回车符号知道了；  倒杠后面跟小s， 空格符号很重要；  小写跟罢跟大写，多得实在不得了；  倒杠后面跟大W， 字母数字靠边站；  倒杠后面跟大S， 空白也就靠边站；  倒杠后面跟大D， 数字从此靠边站；  倒框后面跟大B， 不含开头和结尾；  单个字符要重复，三个符号来帮忙； （\* + ？）  0 星加1 到无穷，问号只管0 和1； （\*表0-n;+表1-n;?表0-1次重复）  花括号里学问多，重复操作能力强； （{n} {n,} {n,m}）  若要重复字符串，园括把它括起来； （（abc）{3} 表示字符串“abc”重复3次 ）  特殊集合自定义，中括号来帮你忙；  转义符号行不通，一个一个来排队；  实在多得排不下，横杠请来帮个忙； （[1-5]）  尖头放进中括号，反义定义威力大； （[^a]指除“a”外的任意字符 ）  1竖作用可不小，两边正则互替换； （键盘上与“\”是同一个键）  1竖能用很多次，复杂定义很方便；  园括号，用途多；  反向引用指定组，数字排符对应它； （“\b(\w+)\b\s+\1\b”中的数字“1”引用前面的“(\w+)”）  支持组名自定义，问号加上尖括号； （“(?\w+)”中把“\w+”定义为组，组名为“Word”）  园括号，用途多，位置指定全靠它；  问号等号字符串，定位字符串前面； （“\b\w+(?=ing\b)”定位“ing”前面的字符串）  若要定位串后面，中间插个小于号； （“(?<=\bsub)\w+\b”定位“sub”后面的字符串）  问号加个惊叹号，后面跟串字符串；  PHPer都知道， ！是取反的意思；  后面不跟这一串，统统符合来报到； （“\w\*d(?!og)\w\*”，“dog”不符合，“do”符合）  问号小于惊叹号，后面跟串字符串；  前面不放这一串，统统符合来报到；  点号星号很贪婪，加个问号不贪婪；  加号问号有保底，至少重复一次多；  两个问号老规矩，0次1次团团转；  花括号后跟个？，贪婪变成不贪婪；  还有很多装不下，等着以后来增加。 |