

导读 | 余晟：我是怎么学习和使用正则的？

2020-06-08 余晟

正则表达式入门课

[进入课程 >](#)



讲述：冯永吉

时长 08:10 大小 7.49M



你好，我是余晟。受伟忠的邀请，今天我来和你聊聊我是怎么学习和使用正则的。

刚工作那会儿，因为密集用到正则表达式，所以我花了不少时间去钻研正则相关的问题，因此获得了机会，翻译了《精通正则表达式》（第三版），后来又写了一本书《正则指引》。到如今，许多年过去了，这些东西还历历在目，我也很乐意拿出来和你分享一下，希望在学习正则的道路上，能给你一些启发。

我经常在网上看到，许多人抱怨正则表达式“难学”，我知道，它确实不好学。但同时我也仔细看过大家的抱怨，发现和我之前的做法一样：用到什么功能，就去网上搜一个例子，又改，能跑通就满意。至于这例子到底如何构成的，自己是不是都懂了，其实心里没底，能大概看懂五六分，就已经很满足了。

这样浮光掠影的使用方法或许能解决眼前的问题，但一定不算“学会”。它有点像打井，每次挖到一点水就满足了，根本不管有没有持续性，也不关心挖没挖到含水层。结果就是，每次要喝水的时候，你都得重新打一眼井。

那么对于正则表达式，我们有没有可能“打出一口永不干涸的深井”呢？当然有，那就是一次性多投入点时间，由表及里，由术及道。一旦掌握了方法，之后就会简单很多了。

按照我的经验，如果每天花一刻钟或者半小时，坚持个把礼拜，通常都能登堂入室，达到“不会忘”的境界。不要以为这时间很多，我知道有些人很喜欢找“正则表达式五分钟入门”，其实每次都没有入门，日积月累，反而浪费了几十甚至上百个五分钟。

那多投入时间很好理解，但是什么叫掌握方法呢？用我的话说，就是摆脱了字符的限制，深入到概念思维的层面。不要盯着那些鬼画桃符一般的字符和表示法皱眉，而要摆脱桃符，把真正的“鬼”给认出来——虽然它们不那么容易看见。也正因为这样，我们才需要一次性多投入点时间。

那最终怎样才算“入门”了呢？按照我的经验，就是通过学习掌握方法，后来无论用正则表达式解决什么问题，都能自发遵循下面的流程去走，甚至能达到不需要这个流程，也能做到解决问题，那基本上就算入门了。

第一步，做分解。拿到一个问题后，我们要先思考：这个问题可以分为几个子问题？每个子问题是否独立？我们拿最常见的电子邮件地址匹配来说。从文本结构来看，它可以分为“username + @ + domain name”这三个独立的部分。怎么画呢？我们可以先画出逻辑结构图。通过这个过程来厘清思路。当然，这是软件工程最基本的思路，相信你做起来应该问题不大。

第二步，分析各个子问题。某个位置上可能有多个字符？那就用字符组。某个位置上可能有多个字符串？那就用多选结构。出现的次数不确定？那就用量词。对出现的位置有要求？那就用锚点锁定位置..... 某种程度上，这就像武术里的见招拆招，每个问题都有对应的解法，只要熟练掌握了，知道什么时候用字符组，什么时候用多选结构，什么时候用量词，什么时候用锚点，就很容易搭建起完整的概念模型。

第三步，套皮。你大概注意到了，到现在，我们还没有谈论正则表达式的典型标志，比如方括号、星号、花括号。要知道，这些典型标志无非只是一些符号而已，真正重要的是字符

组、多选结构、量词等等这些概念。一旦你的概念模型清楚了，写出正则表达式就非常简单了，无非是查阅语法手册，把之前得到的概念模型按照对应语言或工具的约定写下来而已。

许多人觉得正则表达式难懂，总是纠缠于“这里为什么要多一个星号？那里为什么是方括号而不是花括号？”，原因恰恰在于对概念模型不清楚。虽然各种语言或工具对正则表达式的支持大同小异，但细微差别仍然不可忽视。不过只要你心怀正念，洞若观火，这些差异其实并不是大问题。

第四步，调试。很多人都说，正则表达式的麻烦之处在于它像个黑箱子，很难调适，迄今为止仍然没有特别好用的工具，所以我们没法一步步跟进去看匹配的具体过程，只能笼统地知道“匹配了”或者“没匹配”。

那到底怎么调试呢？我的经验是，复杂一点的正则表达式不能一次写对，这是很正常的。与其纠结“这个正则表达式看起来这么复杂，此处到底要用星号 * 还是加号 +，不如先搞清楚，星号（*）或加号（+）限定的到底是正则表达式中的哪一部分，对应要匹配文本中的哪一部分。这两个问题搞清楚了，整个问题就迎刃而解了。

另外，还有一点统摄全局的经验想和你说一下，**那就是学会了正则表达式之后，务必要保持克制**。写正则表达式很容易上瘾，毕竟它的功能那么强大，处理速度那么快，又像天书符咒那样充满了“神秘”色彩。于是，“写一条其他人看不懂的正则表达式，一次性解决所有问题”，就成了某些程序员的执念。但是，从软件工程的角度来看，这种办法绝对是噩梦，不但其他人无法理解，自己过一段时间也会挠头。

那到底该怎么“克制”呢？我的经验有以下三点。

第一，能用普通字符串处理的，坚决用普通字符串处理。字符串处理的速度不见得差，可读性却好上很多。如果要在大段文本中定位所有的 today 或者 tomorrow，用最简单的字符串查找，直接找两遍，明显比 to(day|morrow) 看起来更清楚。

第二，能写注释的正则表达式，一定要写注释。正则表达式的语法非常古老，不够直观，为了便于阅读和维护，如今大部分语言里都可以通过 x 打开注释模式。有了注释，复杂正则表达式的结构也能一目了然。

第三，能用多个简单正则表达式解决的，一定不要苛求用一个复杂的正则表达式。这里最明显的例子就是输入条件的验证。比如说，常见的密码要求“必须包含数字、小写字母、大写字母、特殊符号中的至少两种，且长度在 8 到 16 之间”。

你当然可以绞尽脑汁用一个正则表达式来验证，但如果放下执念，用多个正则表达式分别验证“包含数字”“包含小写字母”“包含大写字母”“包含特殊符号”这四个条件，要求验证成功结果数大于等于 2，再配合一个正则表达式验证长度，这样做也是可行的。虽然看起来繁琐，但可维护性绝对远远强于单个正则表达式。

小结

好了，到此为止，我的经验介绍完了，可以交棒了。

这些年，很多人问过我，我当时到底是怎么学会正则的？说实话，我那会儿根本没想什么，纯粹出于“干一行爱一行”的朴素想法。要用得多，就找书来，哪怕是囫圇吞枣，也要一鼓作气看完。**我一直觉得，真正值得学的东西，没有什么“平滑学习曲线”。在前面的阶段，你总得狠下心来，过了一个又一个坎儿，然后才能有一马平川。**

我觉得，正则表达式属于“没有维护成本”的技能。一旦学会了，每一次遇到这类问题都可以“零成本出击”。所以，长期来看，这绝对是一笔“无本万利”的生意。希望你能通过这个专栏早日达到一马平川！

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

[上一篇](#) 开篇词 | 学习正则，我们到底要学什么？

[下一篇](#) 01 | 元字符：如何巧妙记忆正则表达式的基本元件？

精选留言 (7)

写留言



crown

2020-06-09

我一直觉得要想掌握一门技术永不忘记，最好的办法是造一个出来，影响我最深的2本书<操作系统真相还原>

<自制编程语言-基于c语言>，真正把线程，进程，文件系统实现一遍，而且能运行，那这个知识点怕你一辈子也忘不了。

展开 ▾

1

7



Warn

2020-06-08

学习入门流程：

第一步，做分解。

第二步，分析各个子问题。

第三步，套皮。

第四步，调试。...

展开 ▾

作者回复: 优秀👍



5

5



愤怒阿青

2020-06-08

to(day|tomorrow)

这是开启了正则痴迷开关阿；)

作者回复: 哈哈，真爱才这么写，不过不建议这么用



1

1



chengzise

2020-06-09

余晟老师的这边总结太好了，我觉的这篇可以在课程最后再放一遍。

1

1



Chaos浩

2020-06-09

打卡，经常会用到正则，但每次都要对着文档写。。希望能掌握了

1

1



花生无翼

2020-06-09

短时间，高强度，一次性学会

展开 ▾



keke

2020-06-08

能不能用正则表达式写一本程序天书~让后人来寻宝解密

