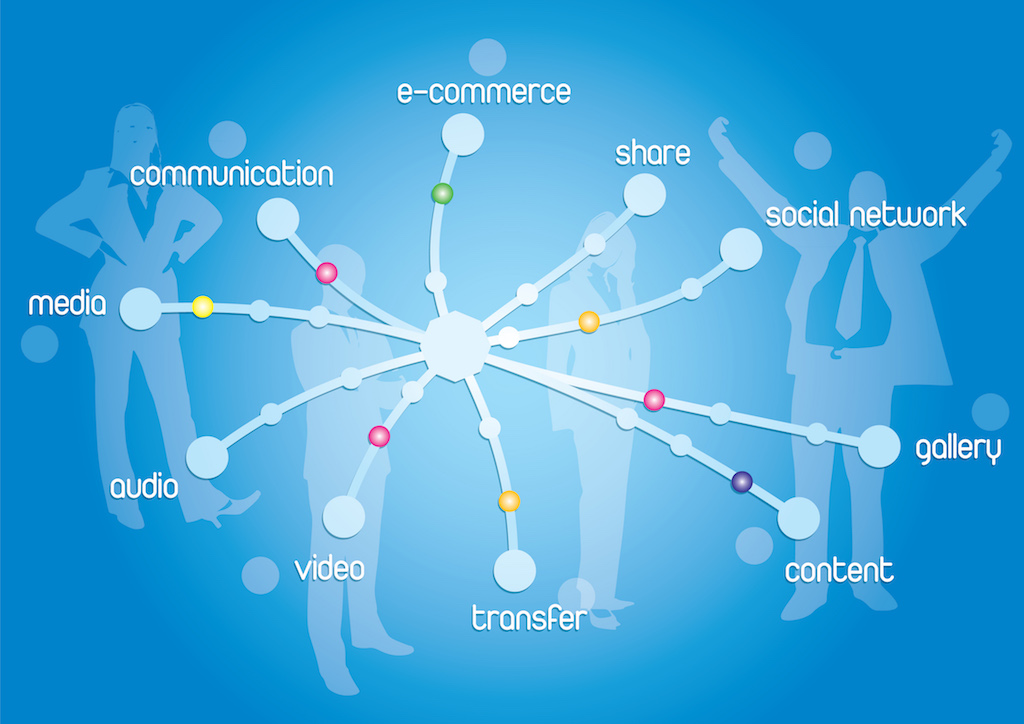
# 序

## 人生苦短，为何不写作呢？

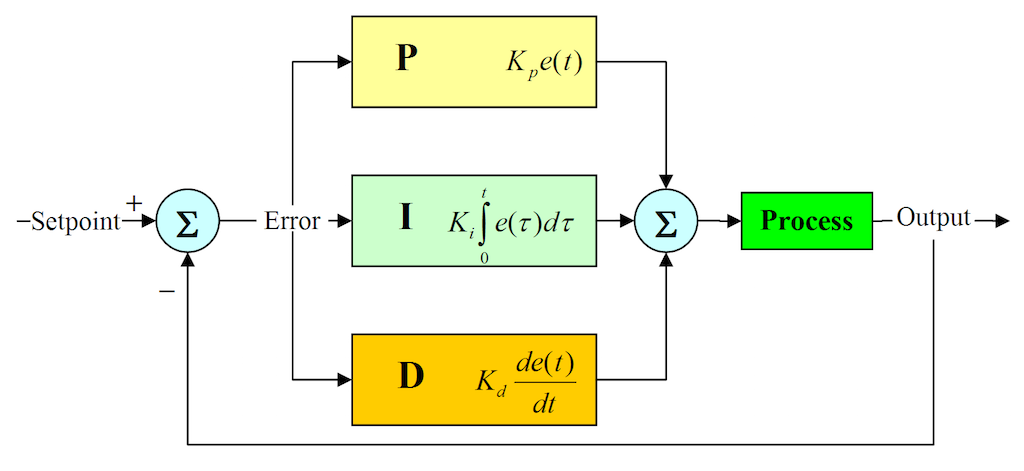
说起刚开始写技术文章时，我并没有这么多体会。并且我也觉得写技术文章是在浪费时间，还不如去多刷刷代码实在。后来我觉得我老了，很多东西都记不住了——越来越依赖于搜索引擎。我们忘记了一些知识点有很多原因：如很久不用、它并不重要，还有一点是它可能没有与其他知识点连接。

而如果你常写作的话，你会发现写作会帮助你连接一个又一个的知识点，构成你的知识网络。如下图所示：



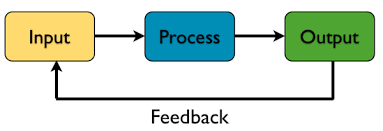
Social Connect

这点是我在写《[全栈增长工程师指南](https://github.com/phodal/growth-ebook)》时最大的体会，并且它也反过来不断地促进我学习——因为它让我发现了我欠缺了一系列的知识。忘记了的重要内容，又再一次需要去回顾。如下图所示：



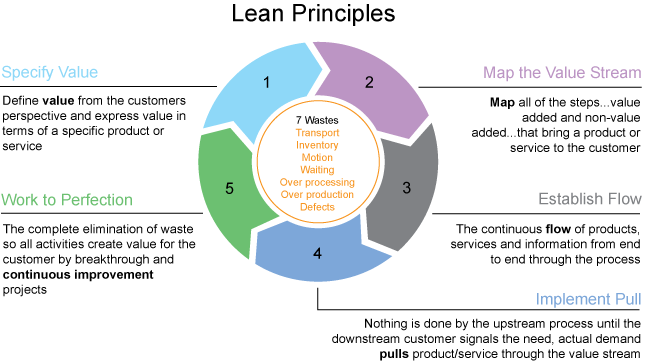
PID

我们就会在这个过程中，不断地去存储那些我们觉得值得重要的东西。



Output is Input

你也可以将之视作为，精益这个陈词滥调：



Lean Principles

我只是想说明：

* 如果一个东西你觉得没用，那么你为什么去学。
* 如果你觉得学了一个东西是有用的，那么你为什么不帮助自己更好的理解呢？

在学习的过程中不断总结，就会促进上面的循环。并且将这些知识分享出来，就会不断地提高自己。

人生苦短，为何不写作呢？

这一系列的文章由六部分组成，即：

* 序，说一些废话用的。
* Analytics，分析不同的内容对于不同受众的影响。
* Create，如何创建用户喜欢的内容。
* Share，如何去分享文章。
* Commucation，与用户沟通。
* 附录，如何反原创。

Blabla，废话终于说完了。

# 分析篇：什么样的文章受欢迎？

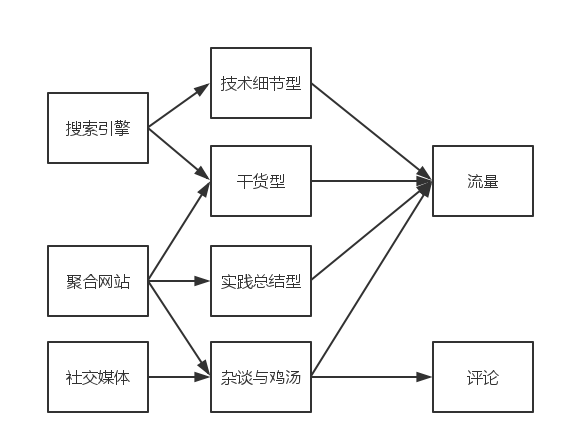
对于以技术为核心的技术博客来说，人们是冲着他们需要的内容去的，绝大多数情况下都不是在闲逛。如果你的网站里没有他想要的东西的话，他便会离开，人们是出于目的去搜索，基于动机，而不是无聊的在闲逛。无聊的话，他们更多的会去刷刷朋友圈的，看看鸡汤。

## 文章的类型

依据我的经验，我将技术博客分为下面几种类型:

|  |  |  |
| --- | --- | --- |
| 文章类型 | 频率 | 时间 |
| 技术细节型 | 几次/天 | 15~30分钟 |
| 干货型 | 几次/月 | 1~2小时 |
| 实践总结型 | 几次/ 周 | 2+小时 |
| 杂谈与鸡汤型 | 几次/季度 | 1 + 小时 ~ 几个月 |

并且他们的流量来源大概来源于下图所示：



traffic-source.png

分析如下：

* **前三者可以带来很多的流量，后者可以带来大量的评论啦。**
* **前三者可以让你在Google上有一个好位置，后者可以让你在用户心中有个好位置。**
* **社交媒体可以让第四种类型的文章，有大量的评论和转载。**
* **干货型和实践总结型的文章通过聚合网站来传播的效果最好。**
* **前三种类型的用户获取到他们想要的信息就走了，不会留下评论**。
* **第四种类型，没有大V的光环，偶尔文章被转企业V转转也能多个几十个粉丝。所以我没有第四种类型的文章太多的经验哈。**

不过，下面我们要依据不同的流量来源来分析文章的用户群体。

### 搜索引擎：完成工作

使用搜索引擎来找内容的用户一般都是为了完成工作，这一类的流量主要是以技术细节型的文章为主。先让我们简单地了解一下，什么是技术细节型的文章——就是看一些标题啦：

* [Django ElasticSearch Ionic 打造GIS 移动应用](https://www.phodal.com/blog/django-elasticsearch-ionic-build-gis-application/)
* [Django Haystack ElasticSearch 环境准备](https://www.phodal.com/blog/django-elasticsearch-haystack-prepare-enviorment/)
* [Ionic ElasticSearch 搜索服务](https://www.phodal.com/blog/ionic-searchview-django-elasticsearch-ionic-build-gis-application/)

这一些文章展现了特别多的技术细节。对于不懂这个领域的人来说，他们完成没有兴趣。因此就算你把这一类的文章推送到各个平台上，也不会有效果的。所以写好内容，然后取个好标题，然后将这件事交给搜索引擎就可以了。他们想要的只是获取他们想要的内容，而我们所要做的就是给他们想要的内容。因为：

用户不想在你的网站花费太多的时间，他们需要有足够的时间去做别的事情

我们只想解决我们当前的问题，完成后我们就可以去刷朋友圈了。

#### 用户行为分析

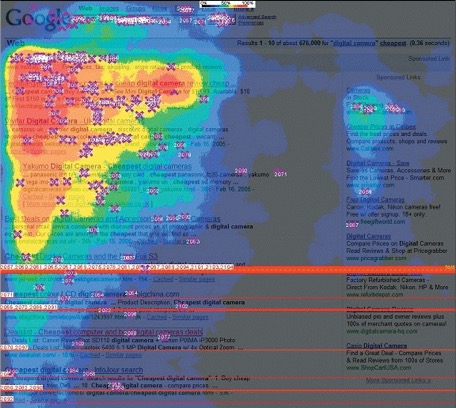
下图是我博客的流量来源分析：



user-behavior.jpg

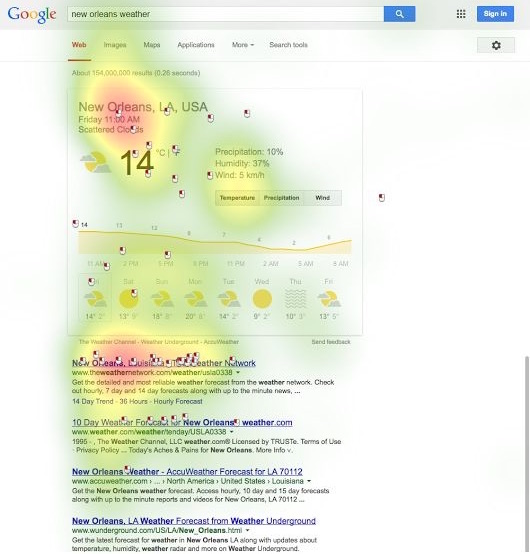
我们可以看到在第二栏“起始页”里，绝大部分的用户（13万/15万）都离开了——即用户找到了他们想要的内容，随后就离开了，这对于其他类型的文章也是一样的。不过，值得注意的是大概会有2万的用户会继续访问其他页面。

主要的用户来源都来自Google，这时就依赖于我们在Google中有一个好的排名。上神图：



google.png

依据Mediative在2014年的研究([The Evolution of Google’s Search Results Pages & Effects on User Behaviour](http://www.mediative.com/whitepaper-the-evolution-of-googles-search-results-pages-effects-on-user-behaviour/))表明，新的眼动研究如下图所示：

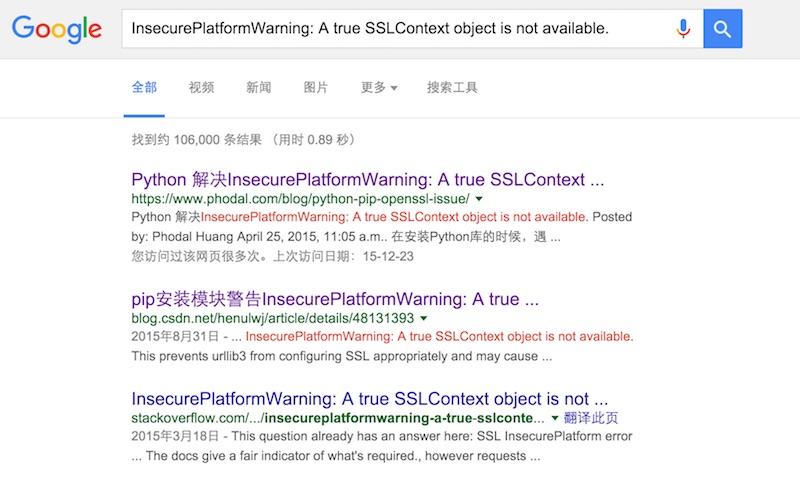


new-google-user-traffic.jpg

这只是在说明Google的语义化搜索，已经初显成效。值得注意的是：2005年时，前4位的链接点击比例是47.8%，而在2014年这次研究的结果是62.6%。

作为一个专业的程序员，我们会用问题的关键词去搜索，如InsecurePlatformWarning: A true SSLContext object is not available. 。 如果这时你的标题就是《[Python 解决 InsecurePlatformWarning: A true SSLContext object is not available](https://www.phodal.com/blog/python-pip-openssl-issue/)》，那么你就找到了你的用户了。

**这是我之前用了5分钟不到的时间写的一篇文章，它在半年的时间里带来了6,599个访问量**。它在Google的搜索结果中排第一：



google-search-python-issue.jpg

(PS: 虽然很不情愿地被排名第二的文章抄袭了——但是我还是第一，这该死的**伪原创**。)

对于大部分用户来说，一看到这个标题就深深觉得这会解决自己的问题，并且它还是中文的。同时，随着用户点击量的升高这篇文章的排名就会越来越高——这就是Google。

用户试了一下上面的解决问题的思路，发现可以工作，于是就走了。当用户发现按上面的步骤来不行，大部分用户就直接离开了。除非这是唯一的解决方案，那么用户可能会抱一点点希望的留个言。

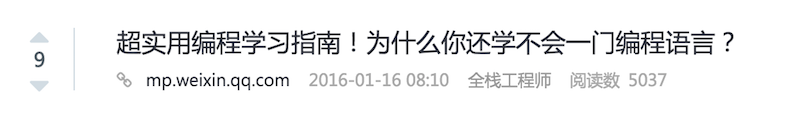
### 聚合网站

对于现今的大部分技术类聚合网站来说，他们的特点都是差不多的——即由用户来提交技术文章的链接，再由用户来对文章进行“顶”和“踩”来决定其在首页的位置。这一类比较流行的有：

* CSDN的极客头条 (http://geek.csdn.net)
* 开发者头条 (http://toutiao.io/)
* 稀土掘金 (http://gold.xitu.io/)

他们的特点都和上面是一样的，唯一有所区别的是开发者头条是由编辑来选出可以上头条的，随后也会进入用户顶和踩的通道。

这里是我们能统计到流量的网站的数据：



csdn-geek.png

上图是文章发布在CSDN极客头条上的，可以看到阅读数有5k+。



gold-xitu.jpg

而在稀土掘金上也有1k+的阅读量，而如果是一些更好的技术文章，那么它就会有更高的阅读量。在这些网站上比较受欢迎的文章类型就是：

* 干货型
* 实践总结型
* 杂谈与鸡汤型

除了好的内容之外，他也应该是普通用户能读懂的读物，如：

* [程序员如何提高影响力 2.0](https://www.phodal.com/blog/improve-impact-2/)
* [Developer进阶书单](https://github.com/phodal/booktree)
* [我的编程生涯里启发我的15本书](https://www.phodal.com/blog/most-important-15-books-of-phodal-programmer-life/)

对于这部分的内容来说，我们只需要关注一点：**人们只关注和自己相关的，并且也是可以达到的**。

除此，还要有一个好的标题：

* [最流行的编程语言JavaScript能做什么？](http://zhuanlan.zhihu.com/p/20731338) 这只是一篇介绍JavaScript在不同领域的应用。
* [听我说说我的博客: 月访问量过万的个人IT博客的技术史](https://www.phodal.com/blog/phodal-blog-techlogoy-development/) 这只用来介绍我博客的技术发展史。

这些文章和总结只要标题好、内容实在，都很容易在聚合网站上有一个好的排名。并且越容易获得比较高的关注，即“马太效应”。

实际上，如果你是在一个恰当的时间发布了你的文章，那么你也会获得一个高的关注。这是个秘密，就不要问我为什么了。

### 社交与新媒体

这一类主要于**传播性**为主，主要适合那些“杂谈与鸡汤型”和不需要代码的“干货型”。因为我不是一个大V啦，我只能介绍一些有限的经验。主要以下面的几个为主:

* 微博
* 微信公众号
* 知乎
* 论坛
* 媒体平台，如今日头条的头条号、企鹅媒体平台等等

需要注意的一点是，微信公众号是封闭的，即如果我们的粉丝量少，那么我们要推广我们的文章难度非常大。这一点同理于微博，但是我们可以在微博上@大V，如果内容好，那么大V自然也会帮你转。而知乎、论坛则是开放型的，而这主要是取决于在这个平台上用户的质量比较高。媒体平台则是另外一种形式，主要是由于笔者尝试了几次，发现在这些平台上也是可行的，除了读者的质量比较不高。

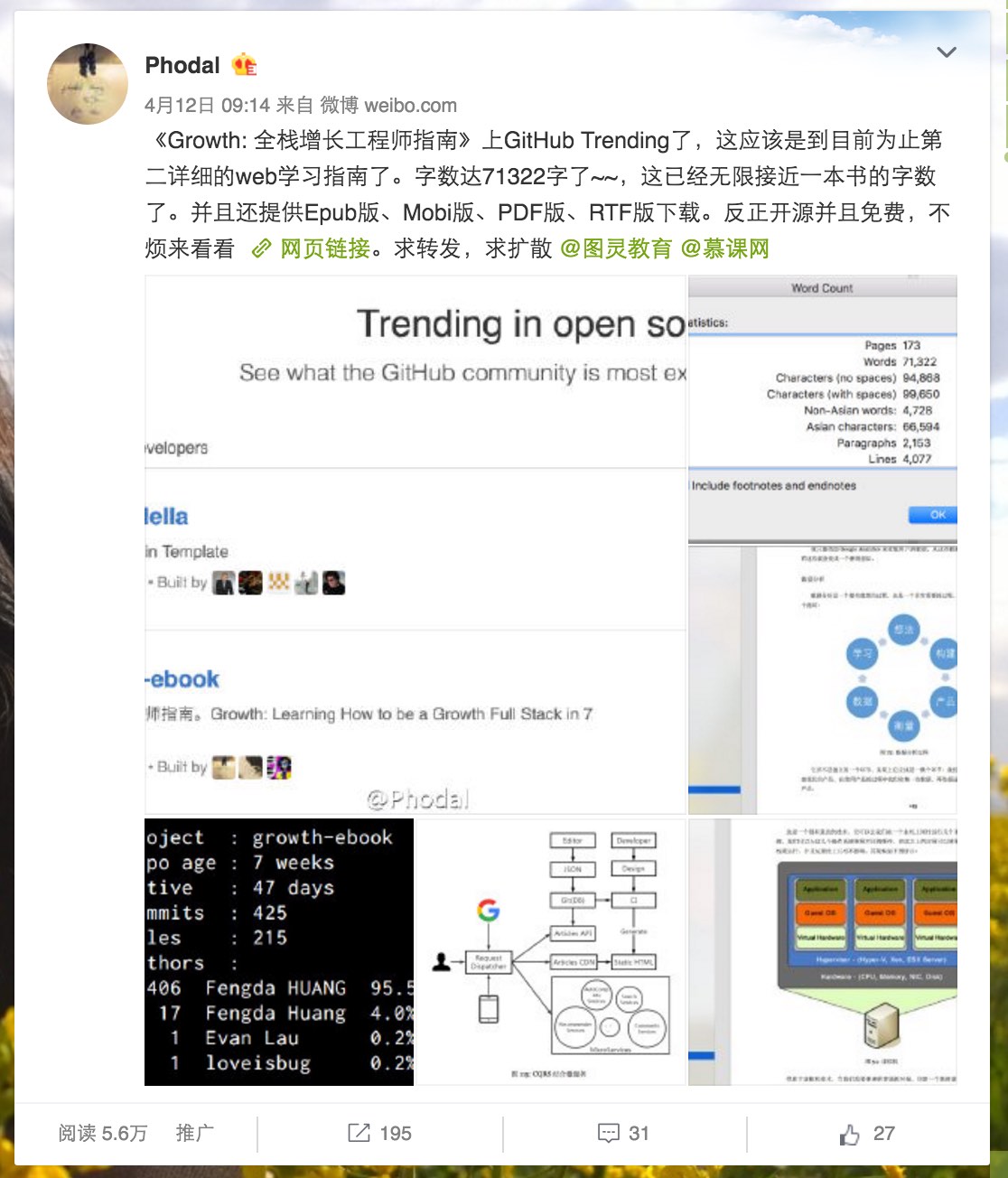
对比如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 类型 | 开放度 | 传播难度 | 粉丝质量 | 保护原创 | 上手难度 | 受众 | 持续性 |
| 微博 | 中等 | 一般 | 中等 | 特别差 | 容易 | 依赖于粉丝数 | 一天~几天 |
| 微信公众号 | 封闭 | 难 | 高 | 高 | 难 | 依赖于粉丝数 | 几天 |
| 知乎 | 高 | 容易 | 高 | 中等 | 容易 | 关注话题的人 | 几天~几个月 |
| 论坛 | 中等 | 容易 | 中等 | 一般 | 容易 | 论坛中的用户 | 几天 |
| 媒体平台 | 高 | 容易 | 一般 | 高 | 难 | 几乎所有相关的用户 | 一天~几天 |

下面就让我们展开上面的内容吧！

#### 微博

技术文章要在微博上传播开来的难度还是挺大的，除非你是大V。但是如果是一些好的内容也可以传播开来，主要看大V的扩散了——大V就是你的文章的引爆点。如下图是我之前写的 《[全栈增长工程师指南](https://github.com/phodal/growth-ebook)》，有幸得图灵教育和慕课网的扩散，因此有了195个转发。



growth-weibo.jpg

然并卵，由于没有分析工具，就没有办法帮分析了。值得注意的是，文章在微博上的传播时间比较短。换句话说，在明天没有人转发你的微博了。它唯一的主要作用是，帮你提高粉丝数。

#### 微信公众号

微信公众号就是一个门槛更高的活动了，坑比较深。如果好友不多、粉丝不高、自身知识度不高，传播起来难度就更大了。由于我的粉丝数量还不是非常多，我只能从中挑选中阅读量最高的一篇文章。这篇《[为什么说全栈工程师是未来？](http://mp.weixin.qq.com/s?__biz=MjM5Mjg4NDMwMA==&mid=405019922&idx=1&sn=354643ee38b0ca423cd0f3025fb98338#rd)》一共有2897个阅读，244个转发。



growth-wp.jpg

所以，你也猜到了主要是靠朋友圈转发。而这有一个明显的限制是，假设1000个粉丝里只有20个转发，那么这篇文章的每天阅读人数就会是就会呈现递减趋势。通常这也是大部分文章的阅读量模型，除非中间有大V转发。

#### 知乎

知乎也是我最近尝试去玩的平台，总得来说上手是最容易的——只要你的内容好，那么完全没有难度。如下图是《[全栈增长工程师指南](https://github.com/phodal/growth-ebook)》的流量来源，会发现知乎占了很大的一部分。这种软植入的广告，在大家看来还是可以接受的。



zhihu.jpg

然，如果我们尝试在上面推广微信公众号的话，可能会遇到一些挫折。不过可以放上微信上的文章的链接。除此还有一个小技巧便是，把你写过的文章贴在相关的答案里加上原文链接即可。

这里还需要注意的一点是，如果一个问题的关注人数比较少，那么可能以后关注的人数也不会多。所以如果你还只是一个小V的话，不烦和我一样将目标放在那些关注人数稍微多一点的问题上。在自己的关注人数上升后，再转到这些问题上，要不会吃力不讨好。

#### 媒体平台

今日头条的头条号是让我最吃惊的一个平台——随便一篇文章都可以有上千、上万的阅读量。如下图就是我最近发的一篇文章：



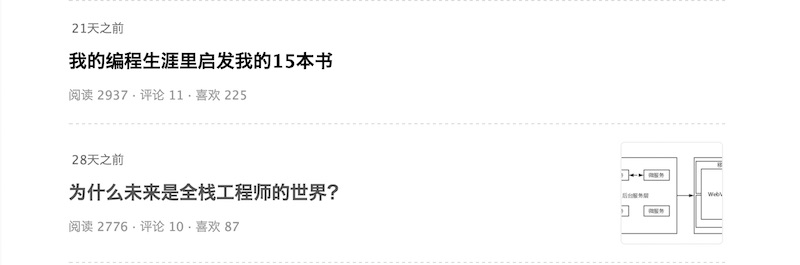
mp-toutiao.jpg

开始在这个平台上使用的原因是，自己发的文章经常到转载到上面，并且被修改得面目全非。同样的，我在开始使用微信公众号时也是如此。随后，我发现在上面可以有极大的阅读量，这下我就放心了。让更多的人可以接触我的文章这就足够了~~。

今日头条采用的是推荐机制，即你只要看过某篇技术文章，那么你还会类似的技术文章。因此我们的文章就很容易被推荐了，那么阅读量就上去了。唯一的缺点是，由于面向的读者是大众，很容易遇到很多喷子。

#### 简书

简书也是一个很不错的写作平台，只是如果我们的文章上不了首页推荐的话，那么我们也不难获得很高的阅读。当你上了首页的推荐后，就会有下面的阅读量和喜欢。受限于技术文章的阅读群体比较小，也不会特别广泛。



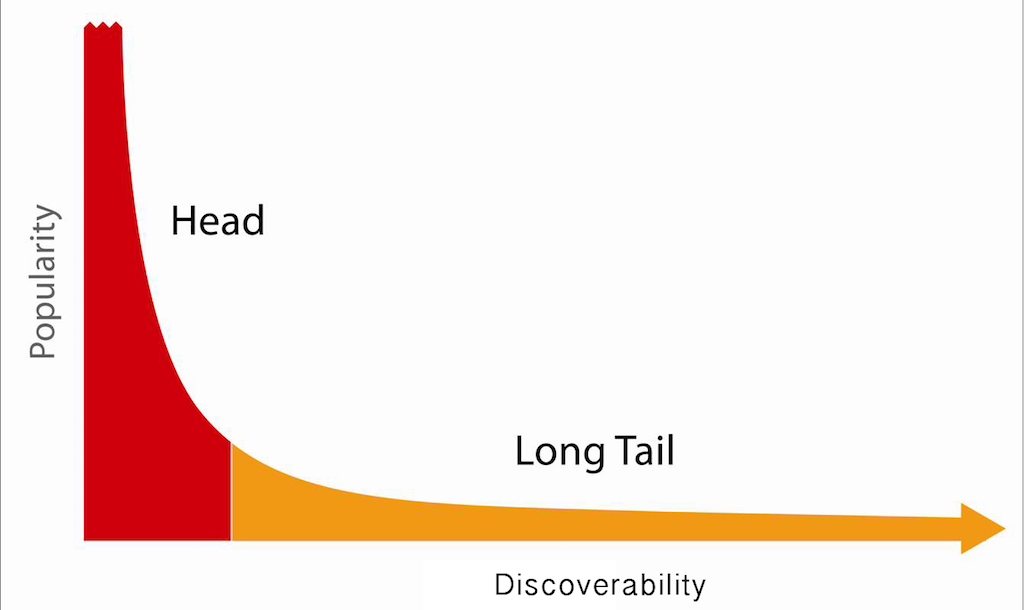
jianshu.jpg

不过，如果你写的是什么值得推荐的APP那就还好，一般人都看得懂。并且简书上是打不了广告的，我们只依靠于在文中放点链接，哈哈~~。

这也恰恰是它为什么是一个非常好的写作平台的原因。

### 小结： 大连接

最后，让我们一张图结束这一篇的内容吧！你能想到什么？？



longtail.png

**如果我们有一篇文章很火爆，那么它应该也要带动其他文章的增长。**

# 创作篇：写作的要点

## 文章类型

### 技术细节型

在我的博客里很大一部分文章都是这一类，主要是原自早先对SEO的研究。对于先这种没有光环的人来说，这是我博客的主要流量来源。这也是让我坚持下去的动力，每天有大概三百个用户来自搜索引擎。

这种类型主要来自于日常工作，但是好像写这一类的人不多。每天我们都会遇到不同的技术问题——如某个第三库更新，某个浏览器bug，如何使用某个技术。因此，我们就会用这样的关键字和词去搜索，反正大部分教程序员使用Google的文章就是这么写的。

既然你想要这样的搜索，那我就直接把搜索结果放前面好了，如：

1. [Mac OS Laravel 安装](https://www.phodal.com/blog/mac-os-install-laravel-4/)
2. [Homebrew versions安装特定版本软件](https://www.phodal.com/blog/use-brew-versions-manage-specif-software-version/)
3. [brew versions gradle安装指定版本gradle](https://www.phodal.com/blog/homebrew-version-manage-gradle-version/)

如果你也擅长用Google的话，我想你就懂了奥秘。标题的前半部分是关键字，后半部分指名意图。

接着，就是贴解决方案和代码咯~~。在这时候需要注意几点：

* 试着去表达一些技术上的细节，也会省去一些技术上的细节。
* 以自己学习的角度去想怎样写可以一步步下来。
* 加入一点点自己的情感，博客和文档是不一样的。（详细见下文）
* 编写更简洁的内容，因为技术博客与此文不同的是，技术博客是面向寻找信息的人。

### 干货型

一般来说，这种类型的文章更类似于GitHub上的那些Awesome类型的Repo。说说几个标题，大概可能就会有点感觉了：

* [每个程序员必知之SEO](https://www.phodal.com/blog/every-programmer-should-know-how-seo/)
* 作为一个前端工程师你了解你的小伙伴么 - chrome
* App 上架相关事宜、解决技巧
* 20 个非常有用的 Java 程序片段
* Java 开发者不容错过的 12 种高效工具

通常来说，这一类型的文章都是作者一段时间对于某一个东西的总结，非常有收藏价值。故而，这种类型的文章会在GitHub或者聚合网站上比较受欢迎。所以，它也更容易传播。

但是写起来的难度比较大，这依赖于你的使用经验。所以，也不是一天、两天就能搞定的。

### 实践总结型

看看标题，可能大家都知道这一类文章要怎么写了：

* JavaScript之闭包与高阶函数（一）
* [编辑-发布-开发分离：git作为NoSQL数据库](https://www.phodal.com/blog/editing-publishing-coding-seperate/)
* [构建基于Javascript的移动CMS——添加滑动](https://www.phodal.com/blog/use-jquery-backbone-mustache-build-mobile-app-cms-add-swipe/)
* Android开发从零开始

这通常是一系列的文章，而这一系列的文章一般是连续写出来的。也因此，我们可以发现很大的书都是由这一类的文章衍生出来的。

这一类的文章更像是干货型和技术细节型的结合，面向特定领域的技术，也属于干货。对于这种类型的文章来说，更依赖于代码——读者需要依据代码一步步往下深入。

所以在这一类型的文章中代码往往比较重要。

### 杂谈与鸡汤型

来几一些标题：

1. 为什么整个互联网行业都缺前端工程师?
2. 程序员职业生涯中应该思考的10个问题
3. 怎么减少编程中的 bug？
4. 为什么大公司看起来都那么糟糕？

BlaBla，简单地来说这一类文章基本上是没有技术的，都是一些以理论为主的概括。同时，写这一类文章的时候，也意料着可能在某一领域有一定的水平——写出来才会有人看。而由于偏向理论，这一类文章看的人往往会更多，因为它并不倾向于使用领域特定的技术。

所以这种类型是受众很广的文章，并且也像鸡汤一样容易传播。同时，这种类型的文章，更依赖于社区媒体及聚合网站的传播。因为大部分情况下，用户不会这么去搜索文章的。

## 写作要点

1. 标题、小标题。一个好的标题会带来更多的流量的。
2. 内容。以代码为例的文章，最好有代码。代码可以放在GitHub上，一举两得。
3. 图 —— 一图胜千言，如本文第一节中的框架图。，一个框架图可以省去你的很多话语。
4. 排版，又可称为UI设计。基本的语法高亮要有，容易阅读也要有。最简单就是白底黑字。

下面有我之前在《[招聘笔杆子——成为笔杆子](https://www.phodal.com/blog/think-of-rework-be-a-writer/)》一文中说到的一些要点：

### 标题

对于写博文的人来说，重点的是如何清楚的去表达他们的想法，标题算是其中之一，这个也就是为什么标题党成为了标题党，而《设计模式》成为了经典。刚开始学编程的时候，更吸引你注意力的可能是《72小时学会Javascript》，而不是《Javascript 权威指南》，兴许让你买前者的原因是因为你能看懂前者，而后者不仅看不懂，而且价格更贵。只是一年以后，《72 小时学会Javascript》被你扔到了垃圾箱，而《Javascript 权威指南》却放在了原来放那本书的位置上。你定义的类难道仅仅应该是class class1么？

### 小标题

小标题有点类似于sitemap.xml，只是他就是站点地图，一点就到了相应的地方。他应该直接了解的说这是开始菜单，标题栏，菜单栏，而不应该是简简单单的第一章，如果你真是那样写的话，你写的函数想必是

def fun1  
end

如果你写的是get\_data那么，我想你的函数名应该和你的文章一样，告诉人们，你要的是get\_data。所以不要吝啬你鼠标的一下，它可以承受上百万字的点击。如果因为那样坏了，你可以告诉我，我可以帮你免费换一个欧姆龙的微动，前提是你的鼠标可以换。如果是HTML那么应该是h2 h3，markdown也就是用得比较多的github上的README.md的##或者是###

### 内容——函数体

这里可不是让你用一个让人难以理解的Magic Number，你写得越复杂，别人看的时间就越久，通俗易懂，就是一个很好的开始。你可以把一个个复杂的方法分解出来，或者提炼函数，或者重命名。当你相信你看不懂你的文章，正如你看不懂你写的hello,world我想你是时候去重构你的函数了。

复杂的部分，就用段落来解决，一个函数如同一个段落只应该表达一个思想，太长了就如同这篇文章一样没有多少人会认真去看。你需要给你写的一个精美的代码加一个注释，所以你也需要给你复杂的地方加上个（PS）。

在编写内容的时候需要注意下面的几点：

* 代码。必要的时候贴出代码，会是一种快速有效的方法，工作的时候是为了完成某个任务。
* 列表。列表可以简明扼要的说明你想要表达的内容。
* 图表。一张流程图可以说明程序的步骤，一个表格可以作好一个对比。
* 标题。标题的好处，可以让用户快速定位，让用户找到想要的内容。

链接。一个好的URL对于使用英文的人来说，可以利于SEO，对于中文，使用的人可以判断出文章的专业与否。

### 引言——README

我想都会去看的，无论是在破解软件的时候，还是github上面的项目。简单的说说，这篇文章是干什么的，这个程序是干什么的。大家都会，不是么？

没有什么好写的？ 说说你是怎么开始编程吧，然后写在你的博客上，你会发现你会爱不释手的。

## 文章排版

### 可读性

### 代码高亮

## 收集灵感

下面是我最近在做的，感觉很不错，推荐一下咯：一般是先有个论点，然后再论据，最后才写。

第一步，在平时把想到的一些能写的文章用TODO的形式列出来，如我用的是“奇妙清单”。下面就是我最近想写的一些文章的一些标题：



wunderlist

但是有一些还没灵感，只是想到了个大概——所以有的是只是一个标题。这些文章大部分都属于杂谈型的，想到什么就说点什么，但是会有个核心。

第二步，在平时偶然会有点灵感的时候，就用Google Keep创建一个新的记事，稍微写几句，提醒一下自己咯。在这一步，相当于就是找一些论据来支撑自己的观点，等到这些观点差不多的时候就可以写了。

Google Keep

Google Keep

第三步，便是写。写的时候其实没有啥感觉，就如同我之前在《[编程同写作，写代码只是在码字](https://www.phodal.com/blog/write-similar-program-just-typing/)》一文中所说的。不得不再说说这个标题，一看就知道要写什么了~~。

真正的想法都在脑子里，而不在纸上，或者IDE里。

所以，其实写代码的时间很短，真实长的是想出怎么写，文章也是如此。

# 分享篇：如何营销

分享和营销之间容易让人傻傻分不清，我也没有想到一个好的标题。于是，我就用了这个标题，到底这也算是一门很有意思的艺术。

## 技术社区

在技术社区上分享自己的文章，大概可以分为两类：头条类网站、网站博客。而如果细分的话，我们可以分得更广泛一点：

* 头条：如开发者头条、极客头条、稀土掘金
* 技术论坛：如V2EX
* 博客：如CSDN
* 问答社区：如SegmentFault
* 社区：如开源中国
* 自媒体：如微信公众号、今日头条的头条号等等

对于我来说，不同的文章会有不同的发布流程，如技术型的文章，我只发在自己的博客上。因为这时候的用户受众主要是来自于Google等搜索引擎。

而对于其他类型，如干货型、实践总结型来说，我的发布流程是这样的：先在自己的博客上发一下，接着先在自己的头条号和微信公众号发布，然后将再文章发布各个博客平台和技术论坛上，最后将文章发布到各个头条上。这样可以保证文章的读者最大化，并且可以极大的提高曝光率。

有时候在别人看来，这仿佛就是一个特别大的新闻——因为你尽可能地在每个社区上看到这篇文章。如我在发布Groth 2.0的时候写了一篇文章《[学好编程，你还需要这个开源APP | Growth 2.0现已发布](https://github.com/phodal/articles/issues/23)》，我的做法如下：

1. 首先，我在我的今日头条号上发了这篇文章，这篇文章将推广给那些对这个话题有兴趣的用户——这些用户是使用今日头条的用户中的一部分，受众最为广泛。
2. 然后在我的微信公众号上发了这篇文章，公众号的受众群体决定了这篇文章的传播程度。但是这篇文章可以在朋友群中传播，这就意味着我们可以在朋友圈获取更多的阅读量。
3. 接着，再发布到社区平台上，如自己的微博上，可以获得一些转发自然是最好的了。
4. 然后，我在我的GitHub上的一个类似于博客的Repo里上传了我的图片，并用Markdown格式再次发表了文章。原因是因为很多网站都支持Markdown，但是不支持图片上传，如v2ex。
5. 最后，再将文章发布到自己的各个专栏上，如知乎、CSDN、SegmentFault等等，以达到最后的扩散效果。

这又可以称之为“媒体矩阵”。

总之就是尽可能大的去接触最终读者，这样可以获得最好的效果。不过，这种行为并不适合经常做，容易招致反感。

## 关联自己

使用同一个ID

## 搜索引擎优化

进行SEO优化

# 沟通篇

## 构建渠道

* 博客评论
* 微博转发
* 后台回复

## 方法

// TODO

# 反剽窃的艺术

## 剽窃

### 聚合网站的危害

当我开始意识到我需要与聚合网站作战的起因是，我的博客在Google中的索引在不断地下降。如我在《[博客反爬虫 策略一——根据User Agent](/static/media/uploads/https://www.phodal.com/blog/nginx-disable-crawlers-by-user-agent/)》 所看到的索引值一样在下降。

Anti Bad Bots

Anti Bad Bots

索引值下降通常不外乎几个简单的原因：

1. Google修改爬虫算法
2. Google认为你作弊了（一些不合法的SEO规则）
3. Google认为你的内容对用户无用 （如重复，太短）
4. 等等

而我们面对的降合网站应该属于第三种结果，他们复制我们的文章，然后Google认为我们作弊了！！！这是什么概念？

**别人抄了你的作业，然后老师说你抄袭！** **别人抄了你的作业，然后老师说你抄袭！** **别人抄了你的作业，然后老师说你抄袭！**

长此以远，你在Google中就看不到你的文章了。当时我也遇到了类似的情形，于是我开始反击。从禁止特爬虫到限制RSS的长度，也才有了下面的结果：

Google index Results

Google index Results

但是第二个网站仍然是聚合网站，What's the fuck。对于我来说，这已经是很好的结局，在Google中的第一个结果是自己的文章。对于必应来说也是一个不错是的结果，然而下面仍然是一堆的聚合网站：

Bing Index Results

Bing Index Results

而这是聚合网站的索引，在百度、360搜索、搜狗等等的搜索引擎就没有那么好运。

### 与聚合网站斗争

你辛辛苦苦写地东西就这样无情地被盗取了，想想便觉得还是GitHub好，至少可以看到有多少人Fork你的代码。虽然是不同类似的网站，但是作为一个技术人员我们总是应该做出点什么。

首先，我们要理解为什么他们的排名会比我们高——主要是因为他们的Rank高，即他们的网站排名比我们高。毕竟，他们抄袭了那么多的网站，很容易就排到我们前面去了。

他们主要有下面两种行为：

* 转载不添加链接
* 原文链接包含rel=nofollow标签 （告诉Google不要前往此链接）

而他们转载文章的基本手法也就是：

* RSS转载
* 爬虫

而对于RSS转载来说，一种很有效的策略就是限定字符的全文输出，并且在RSS的最后加一个原文链接。如下图所示：

With Origin Links

With Origin Links

尽管原文链接会被加上rel=nofollow，但是有趣的一点出现了。我们的文章是全文输出，而聚合网站的文章则会变成摘要~~，你懂的。搜索引擎就会知道哪里会有问题。与此同时一种有效的策略就是使用图片——放在自己网站上的图片，而这个图片多数时候也会被复制过去。那么胜利的天平将会向你倾斜。

与此同时，还可以在一些Rank更高的网站上粘贴自己的文章，这是我最近在尝试的。但是这招只对于Google、Bing来说是有效的，对于百度来说，你懂的钱多就行了——从不指望百度可以从SEO中获取一些好处。并且在这些网站上粘贴上原文链接，那么依据Page Rank就会有下面的结果：

Page Rank for Copy Website

Page Rank for Copy Website

除此还有防不胜防的爬虫，不断地修改自己的Rule?这就会变成一场无止尽的斗争。

## RSS限制

Robot复制/RSS复制/人为复制

没有人会看所有的内容~~ -> 限制RSS长度 -> 简介很重要

看了看所谓的垃圾站的一些文章，发现都是原文的，也没有一些特有的元素。有种用RSS订阅过来的感觉，那么显然在我们的文章加一个原文链接的作用可能不是很大。

* 如果对方的rank比你高，你的文章就很容易被Google定为抄袭
* 对方可以在link上加nofollow，Google从理论上是支持这个的，实现有可能会继续往下走。

于是策略上就有两部分:

* 减少原文字数
* 加原文链接

不良影响:

* 影响RSS阅读用户。

但是从理论上来说，如果用户有兴趣的话，读原文可能是不错的。只能后面再看看，是否真的有影响了。

## 反爬虫

Nginx 限制IP

于是再度拿来了[goaccess 分析 nginx log](http://www.phodal.com/blog/use-goaccess-analyse-nginx-log/)。

|  |  |  |
| --- | --- | --- |
| Visitors | % | Name |
| 58222 | 50.71% | Unknown |
| 21231 | 18.49% | Safari |
| 18422 | 16.04% | Chrome |
| 7049 | 6.14% | Crawlers |

至少有6%是Crawlers，没有办法确认50%的未知是什么情况，这些爬虫有:

* Googlebot
* bingbot
* Googlebot-Mobile
* Baiduspider
* YisouSpider

当然还有一些坏的爬虫:

* AhrefsBot
* Python-urllib/2.6
* MJ12bot/v1.4.5
* Slurp

先让我们干掉这些Bug。。。

### 用User Agent禁用爬虫

于是在网上找到了一些nginx的配置

# Forbid crawlers such as Scrapy  
if ($http\_user\_agent ~\* (Scrapy|Curl|HttpClient)) {  
 return 403;  
}  
   
# Disable specify UA and empty UA access  
if ($http\_user\_agent ~ "FeedDemon|JikeSpider|Indy Library|Alexa Toolbar|AskTbFXTV|AhrefsBot|CrawlDaddy|CoolpadWebkit|Java|Feedly|UniversalFeedParser|ApacheBench|Microsoft URL Control|Swiftbot|ZmEu|oBot|jaunty|Python-urllib|lightDeckReports Bot|YYSpider|DigExt|YisouSpider|HttpClient|MJ12bot|heritrix|EasouSpider|Ezooms|^$" ) {  
 return 403;  
}  
   
# Forbid crawlers except GET|HEAD|POST method  
if ($request\_method !~ ^(GET|HEAD|POST)$) {  
 return 403;  
}

顺手也把curl也禁用掉了，

curl -I -s www.phodal.com  
HTTP/1.1 403 Forbidden  
Server: mokcy/0.17.9  
Date: Thu, 09 Apr 2015 14:22:42 GMT  
Content-Type: text/html; charset=utf-8  
Content-Length: 169  
Connection: keep-alive

当User Agent是上面的爬虫时，返回403。如果是Google的Bot的话:

curl -I -s -A 'Googlebot' www.phodal.com  
HTTP/1.1 200 OK  
Server: mokcy/0.17.9  
Content-Type: text/html; charset=utf-8  
Connection: keep-alive  
Vary: Accept-Encoding  
Vary: Accept-Language, Cookie  
Content-Language: en  
X-UA-Compatible: IE=Edge,chrome=1  
Date: Thu, 09 Apr 2015 14:39:11 GMT  
X-Page-Speed: Powered By Phodal  
Cache-Control: max-age=0, no-cache

符合预期

## SEO

copy/paste -> PR比较高的网站 => 添加原文链接

原文链接 -> 你的网站

Nginx 技术手段

## 微信公众号-原创

粉丝经济

与

建立技术社群

## 难以逾越的沟

图片文字 -> 微博 / 公众号

微博 -> 默许 如知乎

技术难度