

第一期月刊出炉了，我甚至还没来得及细看它的诞生，它就要带给我一个大大的惊喜了。说起来，俱乐部很有一些文笔不错的家伙，平时只是不显山不露水罢了。但我觉得，我们所关注的，却并非是优美的文字。朝暮相处，互相习惯的朋友们，忽然变成跳跃的文字，以另一面展现给我们，会带来格外的震撼吧。他们所经历的，也是你所经历的，他们就在你身边，于是文字内外的世界，就在这一刻重重叠叠，忽然间立体起来了。你看这一篇篇鲜活的文字，或儒雅，或铿锵，是否和平时见到的他们大有不同？八卦的乐趣，就在于此。

谈到技术，俱乐部的 staff 们从来都不缺乏认真的。与人闲聊，谈到自己所热爱的领域，都会意气风发，声调也高亢不少。在我看来，在那一刻，专注于自己的世界，眼中仿佛闪烁着光芒，多平常的面容也会变得潇洒起来。而这次月刊的很多篇文章，恰恰把这精彩的一刻定格，让我们得以反复的品味，品味知识，也品味那一份认真。

纳新结束了，新的 staff，新的主席，突然觉得我们很有些光芒刺眼的味道。我不知道 MSTC 还要带给我多少惊喜，但是我知道，融入了俱乐部，生活就不再孤单。也许短短数年间，大家就会经历离别，但是心中有了这群纯真的朋友，会变成长久的寄托吧。正如我游戏中的名字：心之所向，命之所在.....

卷首语

by cerror

ImageToday design source

ImageToday Design Source ImageToday Design Source

ImageToday Design Source ImageToday Design Source

ImageToday Design Source ImageToday Design Source ImageToday

技术篇

1 移动派报告 No.1

边走边玩

身为达人，怎可不了解手机江湖的
二三事呢？说起江湖，那么.....

8 浅谈 Scala

通天塔

作为 JVM 平台上的新兴语言，
Scala 似乎备受关注，那么它的独
特之处究竟在哪里呢？本文将给
您带来 Scala 的第一印象。



16 围观 Photoshop CS4 中的新功能

itsuhane 在几天前一个不冷静，在自己的电脑上安装了 Photoshop CS4.....

20 小课堂

明明白白用 Word — quark 讲述老少皆宜的话题。

21 展望未来

Google 开始撤退，苹果终止 ZFS 项目，Seven Languages in Seven Weeks，
硅时代的终结.....

休闲篇



23 涂鸦板

大家快来围观小朋友们的涂鸦！

25 lam 和 quark 手牵手

论坛 8g 大点兵。

26 特别篇：玫瑰盛开

公元 2009.1024，Rhythm 在紫金港向 mm
求婚，MSTC 见证了这一美好时刻。

28 实习生活

传说中的魔都上海，居住着一只生命力旺盛的小强；遥远的首都北京，寄居着英俊潇洒的周 core，想知道他们在实习时发生了什么惊天动地的事？

30 读的不是研，是寂寞

读研是什么？好吃么？

houshui 如是说：“第一次学院大会，到会三百多个人，后来大多再没有见过，见过的也大多叫不出名字，我的同学们。”

32 出国体验

飘到枫叶国的熊猫~

34 TCer 大点兵

MSTC 大事记

中秋出游、新人名录、保研名录、换届名录。



创刊号珍藏

36 老人寄语

骨灰们的吐槽时间~



移动派报告

No.1 手机江湖

by Stingrey

很久很久以前，我们是这样定义移动达人的：某人拿着砖头般的大哥大，哼着小调“信号在哪里啊，信号在哪里”，到处跑动找信号。此时无关群众开始嘀咕：“我最讨厌这家烧砖的了。砖头烧成黑色也就罢了，还非要拿着黑砖到处跑。”

如今，已然是手机江湖的时代。如同每个人心中都有一座背背山，每个达人对自己的手机也会有一个定位。身为达人，怎可不了解手机江湖的二三事呢？说起江湖，那么……

巨人 Nokia

什么？没有用过 Nokia 的手机？那么你就不是江湖中人。有这么个说法：Nokia 的手机摔到地上散架了，捡起来装回去绝对还能用。Nokia 拥有敏锐的市场洞察力，全面的市场定位（从低端到高端都有），可以说 Nokia 几乎把握着手机江湖的动向。Nokia 的工业设计并不一定是最好的，但的确是最畅销的设计之一，每个时期 Nokia 推出的产品至少有 1/3 成为街机就足以证明这点。

Nokia 的定位

Nokia 曾说了一句“科技以人为本”，于是当时把爱立信吓到了。其实 Nokia 的定位很明显：让手机成为手机（当然最近发布的 N900 是另外一回事）。Nokia 手头握着 Symbian，想通过手机硬件设计与 Symbian 的结合让用户得到更好的手机体验，是的，仅仅是手机体验，不是终端体验。Nokia 很聪明，这个定位恰到好处地满足了大部分人群的需求，于是 Nokia 赚大了。Nokia 广告也做得好，来瞅瞅 Nokia 的产品线：



X 系列: 2009 年产品线整合时新推出的一个系列。定位大致是音乐娱乐手机，例如 X6，主要是用来代替原来的 Xpress Music 系列的。估计之后 N 系列也将融合到这个 X 系列中。

E 系列: 专注于商务的一个系列。主打产品是全键盘手机，例如 E61i, E71。普通键盘的有 E51, E66 以及刚推出的 E52。E 系列的设计和做工是整个产品线中最好的，通常是铝合金外壳，功能也是将 Symbian 系统发挥到极致。

其他系列: 面向低端的“具有通话功能的手电筒”等等(回忆一下光哥以前的那个手机)。还有就是 Nokia 为了涉足互联网领域，定位在移动终端设备的 Linux Maemo 系统系列，例如 N800, N900 (关于这个系列可以看看这篇文章 <http://www.ifanr.com/4526>)。

Nokia 的操作系统

Symbian: 中文雅称为“塞班”。Nokia 手握着大部分甚至是全部的 Symbian 股份，自然几乎所有产品都是用的 Symbian 系统。Symbian 主要分为两类：**S40**，非智能手机，但却是 Nokia 目前市场占有率最高，也是最方便实用的系统之一；**S60**，智能手机，丰富的扩展软件给了这个系统无穷的生命力。Symbian 还有 S70, S80, S90 等版本，都因为市场原因相继淡出。



Maemo: Nokia 主推的移动终端 Linux 发行版，从功能到界面都时刻吸引了人们的眼球。

壮士 Motorola

“风萧萧兮易水寒，壮士一去兮不复还”，用这句话描述 Motorola 再适合不过了。Motorola 悲情就悲情对市场几乎没有洞察力，于是只能回家吃老本。当人们还在聆听“Hello, Moto!”这句销魂的广告词时，曾经辉煌一时的 Motorola 已经把江湖第一把交椅拱手相让给了 Nokia。虽然 Motorola 在通信方面有着出色的研究成果，但是市场营销确实是一塌糊涂。因为一款 V3 的造型风靡了一下，于是之后 Motorola 的手机造型模具全部都是 V3，只能用变态来形容（虽然中途推出 Linux 系统的 E2, E6，但是已经回天乏力了）。吃老本吃成 Motorola 这样的算是绝了，只能说一句：“壮士，走好！”



双剑合璧 Sony Ericsson

索爱是为数不多合并之后还能赚钱的公司了（虽然现在在走下坡路，想起明基和西门子的合并，那真是悲剧+噩梦）。索尼有着华丽强大的工业设计，同时索尼的娱乐产品也有着庞大的市场。而爱立信在通信方面的研发实力在江湖上是数一数二的，蓝牙技术正式爱立信的独门武器之一，然而有着无数光环笼罩的爱立信却和 Motorola 一样，对市场的把握太差，可以说比 Motorola 还要弱那么一点。当爱立信的手机业务部门快要倒下时，正准备进军手机江湖的索尼抛来了橄榄枝。公元 2001 年，两家一拍即合，双剑合璧去了。双剑合璧非同小可，爱立信的技术，索尼的日系设计和娱乐产品的经验，第一款产品 T618 就使得索爱在江湖树立了地位。T618 的高分辨彩屏，摄像头，多方向摇杆，丰富的色彩搭配，华丽的外壳，瞬间就把 Nokia, Motorola PK 地遍体鳞伤。据江湖小道消息，T618 成为当时结婚订婚必备的物品之一。之后 Motorola 一蹶不振，索爱一鼓作气，通过 2 年左右的市场打拼，索爱成功坐上江湖老二的位置（2008 年，由于某几个产品定位的原因，索爱开始走下坡路，三星和 LG 纷纷跑来抢江湖老二的位置）。

Sony Ericsson 的定位

索尼做了 n 年娱乐产品了，在娱乐产品方面已经炉火纯青，自然索爱的定位也是以娱乐为主。凭借着日系的设计风格，索爱在娱乐手机领域的地位几乎没有人能撼动。索爱比较成功的产品线有作为开山之作的 T 系列，作为拍照手机先锋的 K 系列，作为音乐手机鼻祖的 W 系列。可以说索爱的每个系列几乎都成了江湖的风向标，在这方面不得不佩服索爱的功力了。索爱的所有产品太丰富了，于是就是一个字“乱”。不过，既乱之，则安之，捋一捋，很简单，也就是两个产品线：



Xperia 系列：2008 年新推出的一个系列。定位为以 Windows Mobile 为主要操作系统的商务手机。X1 可以说是黑市上卖 4000 RMB 也会被抢购一空的产品，弧形全键盘，侧滑盖高分屏，精良的 UI 设计，处处都是华丽。据 CCAV 报道，之后 X3 会使用 Android 系统(最新消息，应该是 X10)。

其他系列：是的，剩下的都是其他系列。音乐，拍照，索爱依然坚持自己的路线。



Sony Ericsson 的操作系统

Windows Mobile: 索爱的高端机所用的系统, 按照 Microsoft 的定义, 这个系统的目的是让手机不仅仅是手机。

Symbian UIQ: Nokia 没有买到 UIQ, 索爱买到了。Symbian 系统中唯一支持触摸屏的操作系统。界面和功能都要比 S60 浮云, 但是先天不足是对硬件要求过高, 好吧, 于公元 2009 年初 UIQ 正式悲剧。

其他: 索爱自主研发的系统, 索爱的大部分机型都沿用这个系统。简单, 华丽, 但是... 不是绝对好用。。。

掌中宝 Palm



“掌中宝”怎么像是某个菜名? 言归正传。

Palm, 仅仅是米国硅谷的一个智能设备制造商, 却在江湖中树立了自己独特的地位, 它一直这样表态: “我逗你玩, 我们不做手机的”。是的, 他们从来没有想过要做什么样的手机, 他们只是想做出更好的智能终端。Palm 的气质应该是顽皮中带着韧性。最近推出的 Palm Pre(中文昵称: 胖梨, 哈哈) 甚是可爱, 似乎一开始就要和苹果公司杠上。貌似 Palm Pre 故意要使用 iTunes 来进行同步, 而苹果公司就是不让它同步。每当 iTunes 版本更新一次, 从而限制 Pre 的识别, Palm 公司就很有爱地放出一个 Web OS 的更新, 于是 iTunes 的限制被破解了。正是这样一个有趣的公司, 几乎它的每一款智能产品都在改变着江湖的规则。从 Palm TE(这款 PDA, 几乎定义了江湖中关于 PDA 的规范) 到 Treo(Palm OS, 完全为江湖商人打造的智能设备, 让江湖中人明白了什么才是专业的机器) 再到 Pre(Web OS, 无线充电, 从此江湖中人明白了什么才叫做放毒和放雷)。



Palm 的定位

Palm 从一开始就很明朗，只做智能终端，而且专注于商务化，“Less is More”。因此 Palm 的用户群也比较单一，不过，Palm 的粉丝却有着很好玩的中文名称：“胖友”。由于定位专注，Palm 的产品线很简单，很有意思的是，所有产品都是全键盘的：

PDA 系列：这是 Palm 的开山系列，也是 Palm 最专注的系列。样子虽然不出众，但是绝对能让人体会到 PDA 的乐趣。经典代表作是 Palm TE，几乎成了 PDA 的模板了。

Smart Phone 系列：把这个系列叫做 Phone 还真是让 Palm 难过了一下，Palm 可没有想过做手机的，也许是屈从于市场需求吧，momo Palm！意外的是，Palm 一开始并没有用心做这个系列，Smart Phone 中的 Treo 系列有着可爱的胖胖的外型，优异的是智能商务终端。然而 Palm 很来用的，不是让人整天看屏幕看的屏幕分辨率就成了众矢之的。年发布的 Palm Pre 作为 Web OS 湖看客的眼球。



系列却给 Palm 带来了巨大的成更是让 Palm 走向了辉煌，Treo 性能，于是人们知道了，什么才固执地认为，Smart Phone 是拿浮云的，于是 Smart Phone 系列不知怎么的，Palm 开窍了，2009 系统的浮云之作倒是赚足了江

Palm 的操作系统

Palm OS：几乎 Palm 的所有产品统是目前商务智能手机中最好配置就能运行得很流畅，再搭配上全键盘，这样的组合成了 Palm 屡战屡胜的杀手锏。可惜的是，由于系统底层架构落后，开发进度拖延，Palm OS 6 迟迟不能发布，最终于 2009 年 3 月，Palm 宣布停止开发 Palm OS。

都使用这个系统。应该说这个系的系统了，高效、简洁，极低的

Web OS：这是 Palm 酝酿了很久的浮云系统，继承了 Palm OS 几乎所有的特点。就一个词“浮云”！HTML 5 的支持，全新的操作系统理念，脱离对手机概念的过分依赖，让一切掌中之物都变成智能终端，Palm 想利用 Web OS 铸造自己的帝国，或许 Palm 是真的想做回自己。目前发布的支持这个的“手机”只有两款：Pre 和 Pixi。



谁说黑莓不好吃 RIM

RIM 全称是 Research in Motion, 总舵位于枫叶国的滑铁卢。如果提起 RIM 大家不熟, 那么当说到 Blackberry 时, 大家该明白这公司是干啥的吧, 对, 就是一丫卖水果的, 不卖苹果只卖黑莓。RIM 是和 Palm 气质很像的一个公司, 只专心做邮件终端设备, 估计也是迫于生计, 才开始卖黑莓的。黑莓开始流行是在 9·11 事件之后, 据说当时美国前国务卿手上有很多手机



(估计当时黑莓充其量是个二奶机), 但是当悲剧发生后, 几乎所有通信网络服务都挂了, 只有黑莓的 Pushmail 邮件服务健在, 算是帮了美国大忙(于是 yy 一下 RIM 和拉登是不是有一腿)。现在连现任美国总统奥巴马都强制要把黑莓带进美国白宫, 可见这黑莓口感不是一般的好。黑莓到底哪好吃呢? 黑莓一开始外观不咋的, 有个性感诱人的小滚轮(之后很多其他门派的机子上的小滚轮都是学的黑莓), 凭借接近 Palm OS 系统性能的操作系统, 加上黑莓独特的 Pushmail 服务(事实上, RIM 赚的就是这个服务的钱), 让黑莓成了名副其实的邮件终端设备, 那么爱写邮件的米国人能不掏钱吗? 最近两年, 黑莓花了大价钱请人做工业设计, 于是黑莓比以前也好看很多。

说起黑莓的定位, 真是比 Palm 还单纯, 人家就是做邮件服务终端的。产品细分只能从产品硬件设计上分类, 键盘式(大部分是全键盘, 以前的少数产品是普通键盘)和全触屏式。黑莓的系统也单纯的不能再单纯了, 就只有 Blackberry OS。

苹果派 iphone

苹果就不用多说了, 江湖上从来只有模仿它的, 而它从未被超越过。苹果, 凭借着出色到极致的工业设计, 完善的技术, 几乎每一个动作都让果粉疯狂一阵子。苹果还有着很特别的魔力, 这是目前任何公司都难以做到, 用户会主动去适应苹果的设计, 好吧, 苹果是这么说的: 我就是最好的用户体验。说起 iphone, 多点触摸, 工业设计, 比较有特色而且足够牛掰的系统, 加上一个双赢的 Apple Store, 这样的组合足以让 iphone 成为最畅销的手机。可是, 难道是工业设计真的到极致了吗?



棒子联盟 LG Samsung

之所以要提到这两个韩系的门派，是有原因的。据报道称，手机一开始就是韩国人在远古时期发明的，笑~ 通过几年的蓄势，韩系的门派几乎要把索爱挤下第二把交椅的位子了。继承了日系的设计理念，这让手机有了炫耀的资本。当初 LG 涉足手机江湖，凭借一句“i chocolate you”，还真让这手机卖的跟巧克力一样快。华丽的外观，浮云的界面，就是不好用，还老贵。

本土势力 HTC

江湖之争，怎能少了本土势力呢？就不说山寨了，被联发科 MTK 害惨了，大家瞎搅和，还真当江湖是过家家啊。

先说 HTC，几年前吃掉 Dopoda，一下子坐上国内智能手机的第一把交椅。继承 Dopoda 的作风，只做 Windows Mobile 的手机，因此经过几年的积累就成为全球 WM 手机设备提供商的王者。而且几乎所有其他门派的 WM 手机都是经由 HTC 打造的。去年开始，可能是受到市场多维的压力，HTC 也开始寻求转型，让生产线多元化。之后迅速和 google 合作成功发布世界第一款 Android 操作系统手机 G1。再接着 G2 也慢慢浮出水面。本土势力中有实力和国际巨头抗衡的也只有 HTC 了，bless~ Acer 的 Android 也快出来了，看好，同样也 bless~

手机江湖

大致把最近的手机江湖的门派介绍了一下。查阅史前资料时发现门派还真是多啊，限于篇幅，以上只列举了对自己的产品有清晰定位，并且在江湖中有一定影响的大部分门派。还有像魅族啊（M8 还在成长中，当萝莉放这里讲感觉不太好），联想啊（做手机不淡定，还偶尔山寨），Vertu 啊（太奢侈了）就 patpat 祝好吧！文章内容纯属个人观点，如有雷同，纯属虚构，可能涉及门派之争，谢绝跨省追捕！有兴趣的筒子们还可以看看 8g 文章《无法预见的未来和难以割舍的梦想》：

<http://bbs.ifanr.com/viewthread.php?tid=2583&extra=page%3D1>

<http://bbs.ifanr.com/viewthread.php?tid=2610&extra=page%3D1>



浅谈 Scala

by pluskid

起源

Rails 让 Ruby 一夜成名，红极一时，然而随着越来越多的人开始关注，以及越来越多的应用迁移到 Ruby 和 Rails 上，人们渐渐开始发现，Ruby 是个很漂亮的语言，然而官方的 Ruby 实现的性能却有点不尽人意，于是一时之间各种 Ruby 实现开始崛起。

JVM 作为一个相当成熟的平台，自然不会忘记凑这个热闹，构建于 JVM 之上的 JRuby 也逐渐得到人们更多的关注。然而，人们逐渐发现虽然 JVM 是一个相当通用的平台，但是最初却是为 Java 而设计的，和动态语言有一些格格不入的地方，一些功能虽然可以实现，但是却要做很多折衷，或者要使用很复杂的 work around，不仅编码困难，而且性能也会受到很大的影响。

解决办法之一是期待 JVM 的改进，比如 JVM 中 invokedynamic 指令的引入将让实现动态语言中的方法分派变得更加自然，然而这是一个相当被动的解决办法，更何况 JVM 的改动需要经过层层审核，并且周期漫长。

另一个解决办法是对语言做折衷，Ruby 中有一些比较不常用的功能（比如 ObjectSpace 的遍历），可能对于官方的 Ruby 实现来说，要提供这样的功能是很方便的事，然而对于其他的实现方式也许就需要花很大的力气或者对结构做很大的改动才能实现，既然这些功能不常用，那干脆去掉好了，反而干净了许多，然而这是一个痛苦的过程，因为即使去掉细小的功能，不兼容的问题就出来了，从理论上来说，你这个语言就不能再叫做 Ruby 了，从实际来说，一些原本在官方 Ruby 实现上能工作的第三方库也许就不能在你的 Ruby 上工作了，这样的代价是很大的。

所以，一些痛不欲生的人决定选择第三条路：从头开始，重新设计一门语言，语法重新设计，可以参考 Ruby（和其他语言）的精华之处，青出于蓝，当然，更重要的是在设计之初就要迎合 JVM 的特点，于是实现起来就不会那么痛苦了，而且性能也得到了保证——这看起来就是那唯一正确的光辉大道呀！于是各种 JVM 上的语言如雨后春笋般



出现，名声最响的当属严重兼容 Java 的脚本语言 Groovy、Lisp 派系的新秀 Clojure、以及集异壁之大成的 Scala。

概貌

“集异壁之大成”其实是我自己给它加上的头衔，因为它容纳了各种语言各种风格的优点，Wikipedia 上关于 Scala 的描述是“multi-paradigm programming language”——提起 multi-paradigm，一般都会想起 C++，然而 Scala 却并不像 C++ 那么繁杂和混乱，毕竟是从头开始设计的一门语言，看起来比 C++ 要干净了许多，并且在道上的评价也很不错[1]，当被问起目前除了 Java 之外，还会用哪一门基于 JVM 的语言时，Java 之父 Gosline 的答案是 Scala；而 Groovy 的发明者更是表示，如果 Scala 出现得更早一点的话，他就不会花力气去发明 Groovy 了（慢着，这句话好像似曾相识，Linus 也曾经说过[2]，如果当时有 386 BSD 早出现一年的话，他就不会去写 Linux 了，只不过两人说话的场景要表达的意思还是差别挺大的）。除了人们的评价之外，Scala 在实际项目中也得到积极尝试，最著名的例子应当是 Twitter 团队表示将一部分后端用 Scala 来重写了——对于一个还处于相当初步的语言来说，能被用到这么大的实际项目中去，实在是相当不容易，除了不得不感叹国外的开发团队敢于尝试之外，也真想一探究竟，Scala 的过人之处在哪里？

性能和实现上，我们应当信任这个参与了 Java 泛型库和编译器 javac 的工作的 Martin Odersky 同学的水平，并且，这样的东西只有实际的数据可以说明问题。但是，另一方面，如果是关于语言的语法，则可以掺入许多个人观点了。

关于 multi-paradigm，其实也主要是面向对象和函数式了。对于 OO 来说，由于 Scala 是基于 JVM 的，其底层自然是 OO 的，当然，由于号称 multi-paradigm，所以你也可以使用“过程式”的编程方式或者就像普通脚本一样来写这么一个 Hello World 程序：

```
println("Hello World!")
```

然而这也就是算是一种语法糖了。虽然 Scala 支持直接以脚本的方式来执行，但是它毕竟是 Java 派系的东西，并不适合当 Ruby、Python 脚本来使用，比如，解释器的启动速度会让你抓狂，并且动不动就会牵扯到 Ant、Maven 之类的构建系统，俨然一副完整应用程序解决方案的架势，实在不好意思拿来当脚本用。



所以，通常，我们还是会写一些类，并定义一些方法，最后，在我们期望的程序入口点那个类里面定义一个 main 方法，所以，一个 OO 的 Hello World 实际应该这样写[3]：

```
object HelloWorld {  
  def main(args: Array[String]) {  
    println("Hello, world!")  
  }  
}
```

这里的代码已经能大致看出 Scala 的语法模样了：如果从这个例子来看的话，整体是类 Java 的，比如函数的调用，类、方法的 block 都使用大括号来限定，也是很熟悉的，不过方法的定义采用关键字 def，似乎更像 Ruby 一些，然而这其实只是表面现象：因为在底层它对应着一个 JVM 上的方法，这个就和一个 Ruby 方法存在一个本质的区别：Ruby 是动态类型的语言，而 JVM 是静态类型的。然而这里确实没有写返回值类型呀，那是因为这段代码是我从官方网站抄来的，它省略了返回值类型（还省略了一个等号），如果你喜欢写完整的，那么可以这样写：

```
object HelloWorld {  
  def main(args: Array[String]):Unit = {  
    println("Hello, world!")  
  }  
}
```

Unit 相当与 Java 里的 void，而这里之所以可以省略并不是说 void 就可以省略，而在于 Scala 的编译器会做类型推演，这是一个很不错的东西，如果了解 Haskell 或者 ML 这样语言的同学的话（反正我是不了解），应该可以说出更高深的应用，然而，即使是对很普通的程序员来说，这个功能也是可以省去很多力气的。C++ 用户早就对这样的代码表示深恶痛绝了：

```
for (vector<int>::iterator it = vec.begin(); it != vec.end(); ++it)
```



所以希望给原来那个 `auto` 关键字加个功能，可以这么用：

```
for (auto it = vec.begin(); it != vec.end(); ++it)
```

反正编译器肯定是知道这里的 `it` 是什么类型的了，何必要我再写一遍呢？现在，在 `Scala` 里，连 `auto` 都不用写了，各种需要注明类型的地方，如果编译器能推演出类型出来，都可以省去类型定义，并且，既然可以号称“推演”，自然可以处理很多更加复杂的情况。不过，话又说回来，这只是编译器提供的一个功能，却不要被滥用了，毕竟程序大部分时间还是要拿给人看的。☺

再来看代码的第一行，这里居然写了一个 `object`，这是什么东西？其实原本应该是 `class`，不过 `Scala` 也提供了 `object` 关键字，用来干什么呢？构建 `singleton`，当 `C++` 还挣扎在如何完美地实现 `Singleton` 可以保证线程安全、初始化顺序良好、性能又高[4]时，`Ruby` 在标准库里提供了 `Singleton` 的支持，只要在类定义里加一句话即可实现 `Singleton`，而 `Scala` 则直接提供了这样一个关键字。也许你会觉得这只是语法糖而感到不以为然，然而不可否认，这样一来程序确实更加简洁也一目了然了，何乐而不为呢？

实际上，`Scala` 是一门相当纯粹的 `OO` 语言，`Ruby` 经常以 `-2.abs` 这样的例子来表明自己很 `OO`（事实上我最初看到的时候确实表示惊叹），这样的事在 `Scala` 里也可以做，不过由于运算符优先级的原因需要加上括号 `(-2).abs`，或者可以这样写 `-2 abs`，`Scala` 里调用一个方法可以不用点运算符，也就是说，`foo bar baz` 实际上等价于 `foo.bar(baz)`，这样一来，`2+3` 等价于 `2.+(3)` 其实就变得相当自然了，并且，在 `Scala` 里的方法名称约束相当少，比如，`Scala` 的 `Actor` 库就定义了 `!` 方法来发送消息，和 `Erlang` 里的语法对应了起来。

另外，`Scala` 的内置类型系统是相当完备的，不像 `Python`（暂且不说 `Py3k`）那样，整数居然有两个类型 `int` 和 `long`，并且这两个类型除了都继承自 `object` 类型之外没有任何关系了……-.-

还有一个需要提的就是 `Traits` 的概念，这个东西类似于 `Java` 的 `Interface`：

```
trait Similarity {  
  def isSimilar(x: Any): Boolean  
  def isNotSimilar(x: Any): Boolean = !isSimilar(x)  
}
```



不同的是，它也可以包含部分实现的代码，例如上面的 `isNotSimilar`，如果说 Java 的 `Interface` 是纯接口，而 Ruby 的 `module` 是纯实现的话，那 `Traits` 就是介于两者之间的东西。

再结束关于 OO 的讨论之前，最后还要提一句，那个 `Array[String]` 那里的方括号，并不是表示数组的意思，数组由 `Array` 类来表示，而方括号是类型特化，相当于 C++ 里的模版实例化时提供一个类型。此外，这里的类型还可以是更加复杂的情况，Scala 里有一个称作 `structral type` 的东西，类似于一个匿名的 `Traits`，不妨再来看一段代码[5]：

```
def using[A <: {def close(): Unit}, B](param: A)(f: A => B): B =
  try {
    f(param)
  } finally {
    param.close()
  }
```

看起来有点复杂，有很多括号，其实应该比想像中的要清晰，这里定义了一个 `using` 函数，先来看看这个函数怎么用的，也许能帮助理解：

```
using(new BufferedReader(otherReader)) { reader =>
  reader.readLine()
}
```

调用 `using` 的时候传入了两个参数，一个是 `BufferedReader`，另一个是一个 `block`（了解 Ruby 的应该都知道了，这是个匿名函数，写起来相当方便）。再回到 `using` 的定义，先看方括号里的内容，这里定义了两个类型，`A` 就是一个 `structral type`，这里就是说 `A` 这个类型要定义一个返回值为 `void`（我还是比较习惯 `void` 的说法，不清楚为啥 Scala 里用 `Unit`）的 `close` 方法——看到了吗？这就是 `Duck Typing`[6]，它不需要 `A` 类型去实现什么 `interface` 或者是 `traits` 之类的，只要你定义了 `close` 方法就可以了，这正是动态语言宣扬的 `Duck Typing`，然而这里的方式却更加优雅：不是在运行的时候发现没有这个方法而抛出一个异常，而是在编译的时候就进行类型检查。如果你想象力再丰富一点，应该会联想到 C++ 的 `Concept`，没错，就是那个 C++0X 里最受人关注的新特性之一，对的，对的，也就是前一阵子在 C++0X 中的 X 要变成十六进制时，突然被提出要从标准中移除[7]



然后引起轩然大波的 Concept，至少在我看来，Concept 和 structural type 在这里没有什么本质的区别了。

再回到前面那段代码，剩下的篇幅已经不多了（其实现在已经超了吧？）从这段代码出发，我们开始体验 Scala 里的函数式编程。首先，在 Scala 里，函数是一个对象，这个在前面也说过了，这里的 using 方法，第二个参数（为什么要写到两个括号里？因为这样第二个参数以一个 code block 的形式传递进来就方便了许多）的类型就是一个函数，这里 $A \Rightarrow B$ 表示参数类型为 A，返回值类型为 B 的函数，当然，既然号称支持函数式编程，高阶函数显然是支持的。另外，Scala 还支持 partially applied function，简单来说，就是把一个函数的某些参数“固定”变成另一个函数，比如原本一个有两个参数的函数 plus，固定其中一个参数为 43，变成 plus43，则调用 plus43(2) 会得到 45，当然这样的东西其实在几乎任意语言里都可以做，大不了再定义一个只有一个参数的函数 plus43，再里面调用原来的函数 plus 嘛。然而，在函数式编程语言里，这是被视为及其丑陋的，而 Partial Application 才是及其自然的事，比如在 Haskell 里，如果原来函数需要 3 个参数，但是你只提供了 2 个，那这就是 Partial Application，你会得到一个函数，当你再提供一个参数的时候，才会得到那个最初函数求值的结果。Scala 虽然不能做到 Haskell 那样，但是已经很方便了，这里用一个下划线来标记一个 Partial Application，例如，对于上面的 using 的调用（这里再次用了一下把参数分在两个括号里的好处）：

```
def foo = using(new BufferedReader(otherReader)) _
```

这里的 foo（我实在不知道取什么名字好了）就是 Partial Application 的结果，这是一个函数，当你提供另一个参数（比如那里的 block）的时候它才会被求值。当然在这里这样做似乎意义不大，不过我只是从这个例子出发开始讲而已。另外，在某些时候，如果类型推演系统能自动推演出需要一个函数类型的话，下划线有时候也是可以省略的。

最后，由于篇幅限制（我说了很多遍了，不过，此时还多了另一个理由，我已经困得不行了），我决定就此结束讨论，不过，在那之前，还有必要提一下 Scala 提供的 call by name 支持。

现在我们看到的大多数主流语言都是支持（并且通常是只支持）call by value 的，也就是说参数在进入函数体之前肯定已经被求值过了，不过 Scala 还提供了 call by name 的支持，参数在传递进函数的时候并没有进行求值，而是在函数里第一次使用的时候才会



实际求值——换句话说，如果这个参数最后没有被用到的话，就不会被求值了。一个比较明了的例子是日志，在 Scala 里，我们这样写：

```
def log(level: Level, msg: => String) =  
  if (logger.level.intValue >= level.intValue) logger.log(level, msg)
```

注意第二个参数 `msg`，看起来有点像一个函数类型的参数，这就是 `call by name` 的语法，这个和普通的 `call by value` 相比有什么好处呢？如果要记录的 `msg` 是一个构造起来开销比较大的东西的话，当我们不需要记录这个日志时（`log level` 被调高了），就可以完全避免掉这个构造的开销了。

`call by name` 能干一些什么有趣的事呢？这个也许你可以自己去想一想，因为 `code block` 的语法已经相当灵活了，所以要写一些诸如 `my_if`、`my_while` 之类的东西已经很容易了。另外，如果想尝试一下 SICP 里提到的那些“流”之类的，那么我友情提示一下：Scala 支持尾递归优化，并且也有类似于 Lisp 里的链表那样的列表结构，另外，除了函数参数之外，其实普通的值也是可以 `lazy` 的。

总的来说，Lisp 的宏号称灵活性极强，能够随意定义自己的语法，然而代价是我们必须面对一堆一堆的括号，现在看起来，似乎我们可以在 Scala 里以一种更加自然的方式来做这样的事了。接下来嘛，回去睡觉！

参考文献

- [1] <http://www.infoq.com/news/2009/07/scala-replace-java>
- [2] <http://www.abc.se/~m9339/linux/linuxdoc/linuxnews03a.html>
- [3] <http://www.scala-lang.org/node/166>
- [4] 《Modern C++ design: generic programming and design patterns applied》
- [5] 《Beginning Scala》
- [6] http://en.wikipedia.org/wiki/Duck_typing
- [7] <http://www.informit.com/guides/content.aspx?g=cplusplus&seqNum=441>



后记

若缘约我写一篇技术文章，原本我是打算写关于 Powershell 的东西的，因为我觉得这个东西看上去很不错，可是后来发现用起来也是相当麻烦，大概 Windows 下果然还是不太适合用 shell 吧，所以我还是回到最初想写的 Scala 这个话题了，其实我一直都想写的，但是我自己却对这个东西了解也并不多，所以怕写错。

不过后来若缘又叫我不要写太长，而且截稿日期也很快要到了（还是已经过了？），所以我打算不管怎么样还是写一点好了。我通常认为一篇技术文章的话，内容需要做到务必准确，所以通常写一篇文章要查阅很多资料，并附上很多链接，但是现在时间和了解都非常有限，所以这篇就当作休闲读物好了，里面的东西大多是我自己的观点，比如“起源”一节的内容，当然，这样的东西其实也没有一个所谓“官方的说法”吧；而后面的介绍，其实也是从我自己的认识和角度来说一下对 Scala 体验了，并不能当作全面的介绍文章来看，如果因为我介绍得片面性而让你对 Scala 产生一些误解，那就是大大的罪过了，所以，如果感兴趣，还请参考它官方的介绍。

偷偷地说：其实“后记”是最先写的。





一起来围观 Photoshop CS4 中各种实用的新功能

by itsuhane

Photoshop CS4 正式发布到现在已经有一些时间了。itsuhane 在几天前一个不冷静，在自己的电脑上安装了 Photoshop CS4，于是就有了今天这篇文章啦~~~



从关于中查看到 Photoshop CS4 的开发代号为 Stonehenge

在 Photoshop CS3 推出的时候，Adobe 就已经尝试性地在 Photoshop 中增加了一些“非主流”的新功能，这其中很多显得过分简单，使用起来存在着很大的局限性。在 Stonehenge¹ 中，这些功能纷纷走向成熟，无论在实现效果上还是控制精度上都有了提高。此外 Stonehenge 下还增加了许多便捷的新功能，并强化了许多常用功能的快速访问，界面更加紧凑，但更加高效。为了适应互联网上的设计需要，Stonehenge 更加提供了几个很实用的扩展功能。在性能上，Stonehenge 特别针对各种 GPU 进行了优化，使得图像处理的速

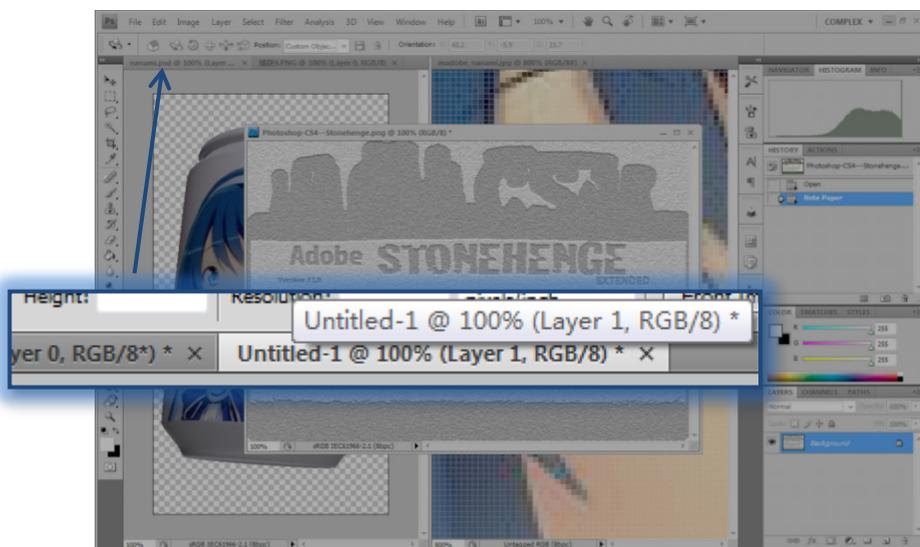
¹ 下文中我们使用开发代号指代 Photoshop 对应的版本，Photoshop CS3 的开发代号为 Red Pill。



度更快，效果也更好。下面就让itsuhane来为大家简要介绍一下其中一些十分实用的新功能。

可划分和吸附的编辑区

Photoshop 一直以来采用的是多文档界面 (MDI)，在同样的编辑区中，设计师可以同时打开多个文档，方便文档间参考、交换内容。但是使用过类似 Word 97、Visual C++ 6 等各种传统 MDI 应用程序的人一定体会过诸多文档在同一个编辑区中打开时，管理窗口的痛苦。从 Office 2000 开始 Office 便渐渐开始放弃 MDI 回归到传统的单文档界面 (SDI)；Visual Studio.net 则开始尝试标签页的方式 (Office 控一定也知道一个 Office 窗口标签化的插件 ^_^)。Photoshop 在 CS3 (Red Pill) 时还在使用着最传统的 MDI 界面，Stonehenge 中则引入了全新的可划分和吸附的编辑区：当有多个文档被打开时，每个文档可以以独立的窗口浮动在外，也可以像工具栏一样成为一个标签页，在编辑区中独自占有一个区域或者与其它文档标签合并在一起。这样的设计大大方便了多个文档的管理，并且节省了很多的界面空间。

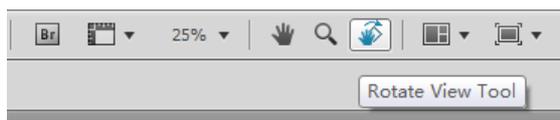


分离的文档窗口与标签式的文档窗口

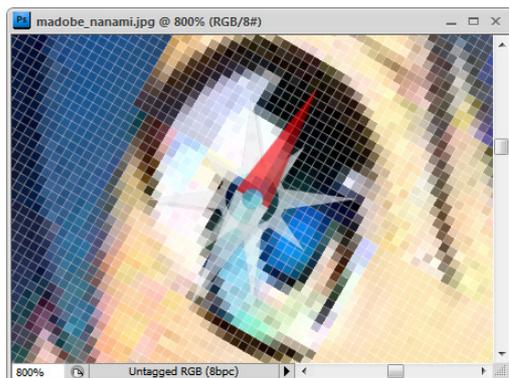


改进的 2D 视图控制

现在，越来越多的平面设计师开始在 Photoshop 中直接进行数码绘图。为了加强对画布的控制，Stonehenge 中加入了一个旋转视图的功能，可以将当前文档或当前全部文档的视图进行旋转，而不实际影响文档内容。不过这一功能有一个小 bug：在对全部文档同时进行旋转时，如果在工具选项里选择角度，则只能应用到当前活动的文档，而不会应用到其它文档。



从菜单栏紧凑的按钮可以发现好多常用功能，这里还提供了一个快速平铺文档的工具，十分方便。图像放大后可以看到像素网格，这个网格显示时竟然是消锯齿的！话说回来，旋转文档时出现的指南针看上去好土呀 >_<



此外，为了提供对文档细节的控制，在 Stonehenge 中，放大工具提供了像素网格的显示，当文档放大达到一定倍数时，用户可以选择是否显示像素网格，在需要调整细微位置时这个功能显得十分有用，可以往的 Photoshop 版本中却始终遗忘了这一功能。

增强的变换工具

Stonehenge 中增加了一个新的变换工具：Content-Aware Scale。这是一个神奇的变换工具，它最强大的地方在于：你可以指定一个不希望在缩放变换时发生较大变形的区域，之后你进行的缩放变换将会尽可能保证该区域不变。

例如下图(a)，普通的缩放后效果如图(b)，而如果我在缩放时希望图像中的路灯和树冠不会被影响，可以建立一个如图(c)所示的保护区域，此时用 Content-Aware Scale 完成缩放后的效果就比图(b)要自然很多了，可见图(d)。



图(a) 原始图像



图(c) 设置保护区



图(b) 普通地沿 y 方向缩小

受到上级指示,扣人心弦的图(d)将留作下期继续放送,敬请期待。

to be continued...

图(d) 在 CAS 下缩小的结果?!!



记明明白白用 WORD 小课堂

BY QUARK

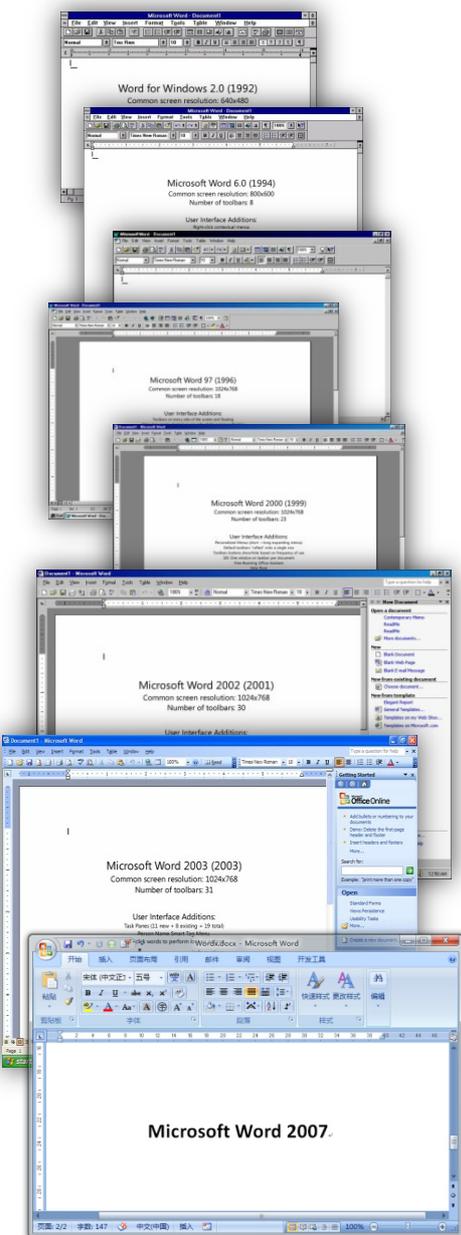
面向全校学生的小课堂算是 MSTC 的一大招牌节目了。新学期，许多人都搬到玉泉校区了，俱乐部紫金港校区人手短缺，要在多快好省地给新生留下一些好印象，所以就选择了一个老少皆宜的话题——Word。

我现在用 Linux 比较多，而且大部分经验还是在 Word 2003 时代，不过看了俱乐部内部流传的各种人的各种 doc 文档之后，觉得还是自己对 Word 更了解，艰巨而光荣的主讲任务就落到我头上了。

为了这次小课堂，我把试用中的 Office 2010 换成了 Office 2007 中文版。Word 中可以讲的话题也比较多，我选择了一些我认为大家必须知道的和一些看起来有点技术含量的东西，参阅了从小胖 ftp 那里搜刮来的一些电子书（实际上真的没有什么收获，只是心理安慰而已 -.-b），精心排版了一本小册子，并且组织了一次试讲，必须要做得比较好了~

虽然在打印小册子、现场投影仪的使用上都出现了未预料的问题，但最终都化险为夷了。这是我的第二次小课堂，尽管如此还是有点紧张，一个小时的小课堂平缓地结束了，自我感觉和大家评价都不错，最后还有热心同学来找我签名，这也是一个未预料的情况啊。

其实 Rhythm 在很早以前的小课堂就有一次介绍过 Word，后来找到那本尘封的小册子，Word 变化得比较大，这两本小册子差别蛮大的，觉得自己做得内容实在一些 😊





展望未来

传 Google 开始撤退，撤走中国境内的所有服务器

今天上午，资深互联网人士胡延平的性浪围脖出现一条消息：

确切消息：Google 开始撤退，所有在中国境内的服务器全部搬走。

此消息一出立刻被大家广为传播，目前这一消息准确性尚不明确，但谷歌公关部已经给予了否认，同时新总裁刘允的手机处于关闭状态。

另据 sina 的一条新闻，称：

另有知情人士对新浪科技表示，谷歌的确准备撤走全部在华的服务器。不过该不愿具名的人士表示目前时机比较敏感，暂时不方便透露更多的内情。

据谷歌在行业内部的消息灵通人士称：

确实要撤，但只是撤掉一部分服务器，也不是要退出中国市场。这次动作对最终用户没有什么影响。

看来只是一次普通的 IDC 调整被炒大了？

新闻来自：谷歌——探寻谷歌的奥秘



苹果终止 ZFS 项目



苹果正式宣布了终止 ZFS 项目。之前，苹果曾广为宣传将在 Snow Leopard 服务器版中使用 ZFS 文件系统。然后几个月之后，苹果网站所有涉及 ZFS 的文章都销声匿迹，到 Snow Leopard 正式上市，ZFS 也不见踪影。但苹果并没有正式宣布放弃 ZFS，直到现在才在项目主页上贴出一份终止声明。邮件列表和库也将在不久以后移除。有人呼吁开发者继续开发苹果 ZFS 项目另一个分支 zfs-mac。 (via cnBeta)



Seven Languages in Seven Weeks

如果顺利的话, Pragmatic Press 将会有一本书《Seven Languages in Seven Weeks》在将来出炉, 七种语言包括:

- Ruby
- Io
- Scala
- Erlang
- Clojure
- Haskell
- Prolog

只是不知道写一本书大概要多久, 也不知道出来以后到我们这边能找到可以看的版本又需要多久。当然, 指望 7 weeks 能学会七门语言是不现实的, 不过作为了解各种编程范式的一个总结性材料, 应该还是不错的资料。

硅时代的终结?

英特尔 CEO 欧德宁(Paul Otellini)周四表示, 英特尔不久后将放弃硅材料, 转而使用其他新材料制造处理器。欧德宁说, 在处理器市场, 硅材料即将被淘汰。英特尔最多还会生产三代的硅材料处理器, 之后便使用新材料。

欧德宁还透露, 英特尔已经在使用新材料制造处理器, 但并未透露更多详情。欧德宁说: “新材料很酷, 相信我。”

(via cnBeta)

Opera Mobile 即将登陆 Android 平台

Opera Mobile 9.7 虽然是目前 Windows Mobile 和 Symbian 平台上最流行的手机浏览器, 但一直没有推出 Android 版本

(Android 平台上只有 Opera Mini)。不过, 看来这个结论很快就要被改写了。昨天, PCMag 杂志对 Opera CEO Jon von Tetzchner 的采访中, Jon von Tetzchner 说道: 是的, 我们正在开发 Android 版本。

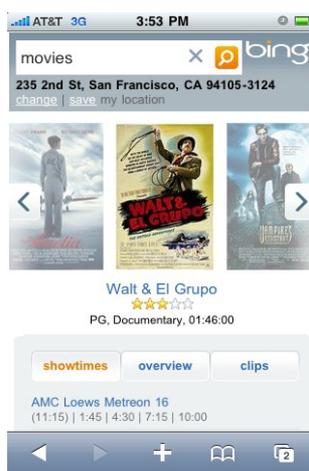
Opera Mobile 可以在你的移动设备上完整的支持 HTML 渲染, 同时还可以支持 Widget, 而目前 Android 和 BlackBerry 用户都只能使用功能被大幅度阉割的 Opera Mini。

Android 自带基于 Webkit 内核的浏览器, 尽管这款浏览器也非常不错, 但相对于 Opera Mobile 的下载管理器、灵活的书签管理器以及更快的速度来说, Opera Mobile 显然更加值得期待。我们很乐意看到 Opera Mobile 如何在 Android 平台上击败自带的 Chrome-lite 浏览器。 (via cnBeta)



Bing 移动版更新 iPhone 完美展现

微软周五更新了其移动搜索网站 m.bing.com, 为触控式屏幕提供了优化,



许多按钮都被加大, 页面结构也更类似于 PC 版本。

目前多款手机平台都已经可以完美使用这款手机搜索引擎。(中文版未更新)

(via cnBeta)



quark 听不到 bg 的宿命

寒 = 寒仔; Q = quark; L = lam

脚本/ kid; 绘画/ lam





by lam



原来是 quark sama 被指控。。伟大的爱情需要一点波折，lam 后面帮 quark 洗

你完了，你要同时被 lam 和 quark 踩死.....

1.你还没 quark 正太虽然你很正太
2.lam 心中是没有谁能比 quark 正太的

我预言 quark 被逐出 TG 被迫和 lam 私奔，寒仔成为下一任 TG Leader，谢谢

lam 的表情好 kawaii! 配合 PS 的文化衫，真是绝了！quark，你和 lam 真是绝配！

lam 把考拉拼成的 QUARK 送给了 quark，结果 quark 吃饭的时候都不好意思坐在 lam 旁边了~~

quark在2009/8/26 23:59:55的发言
exception

```
Story story = null;
Flower flower = new Flower();
flower.sendFlower(
    );
Dreamer dreamer = new Dreamer();
dreamer.dream(
    );
story = dreamer.imagineStory();
```

友情提供 lam 搓衣板惩罚 quark

quark 和 lam 在审美上出现了分歧!!!

在得知 lam 的心意之后，这下 quark 乐了！

以下是引用 lam 在 2009/8/27 21:28:15 的发言：
quark 太坏了

在 lam 眼里 quark 最可爱

lam 别跟我抢 quark

真想撮合小 quark 和小 lam 呀~

写代码的时候YY别的手指就死了好。。。

lam 紧张了，quark 不疼了！

欢迎围观水楼

41345

lam 救得太及时了。。。于是 quark 的私房钱看来是进不了大家的胃了。。。

lam 现在已经开始受 quark 习惯的影响了？赶紧考古去

最新消息
88 MSTC 版 9825

小 quark 你就随了 lam 姐吧，下次再讲吧



特别篇 ~玫瑰盛开~

- ♥ 9月12日，zsn 和 MSTCbaby 举行婚礼
- ♥ 10月2日，windreamer 婚礼。
- ♥ 10月5日，cyourworld 婚礼。
- ♥ 10月17日，seemy 婚礼。
- ♥ 10月24日，Rhythm 向粽子求婚成功。







This Summer

2009-09-15 14:50:06

by hzqtc

ebay@Shanghai

刚刚过去的这个夏天对我来说很有意义。回想起 7 月 5 号三个男孩一起从杭州辗转来到张江，每个人都拖着一个大箱子，一直到晚上 10 点多才把房子租好，还不能马上住进去要等到 10 号。只能去张江镇上的小旅馆里面开了一个标准间，三个人把两张床拼在一起挤了 5 天才搬进现在的房子。那几天条件最艰苦了，但是每天都很 有活力。现在过的好多了，睡在舒服的大床，有空调有电视有冰箱。不过每次路过那个小旅馆我都会向曾经战斗过的地方致敬。

现在每天的生活都很规律，早上 7 点 10 分起床，洗漱后骑自行车半小时去公司上班（偶尔会坐小黑车）。公司楼下的 Lawson 买个包子，加上公司免费的牛奶作为早餐。工作其实比较轻松，一开始的一个月基本上没做什么事，都是 training 和自己学习。后来干的活也都比较轻松，没有什么紧迫的 deadline，也不是非常难，而且还有同事会帮我解决难题。11 点半左右在 basement floor 的小餐厅吃中饭，然后午休的时候上网灌水睡觉。下午 1 点继续工作，其实工作的时候也是比较随意的，做其他的事情也不会有人管。外企的好处就是非常自由和人性化，只要能完成工作任务，没有人会管你做什么的。下午 5 点下班，有很多人自己留下来加班或者做其他事情，公司的网络比家里好很多的。我们研发部门留下来加班的人比较少，我是一下班就跑回去的= =除了这些之外，每周会有点不同的地方，比如周三早上有免费的早餐，每周都不一样；周四下午又冰淇淋，不过只有暑假有，现在已经没了，好怀念夏天啊；周五中午有 BrownBag，就是实习生跟老员工的交流会，提供午饭的，不过也是只有暑期期间有。

下班后的生活比较爽了，一般先找地吃饭，最近去的比较多的是一个类似食堂的快餐店，有时候会去撮一顿。然后回去打打球什么的，回家后原来还一起玩 dota 的，最近不怎么玩了。一天工作下来还有点累的，上上网玩玩游戏到 11 点多就睡觉了。不过遇到周末就会玩的晚一点，现在很享受周末的闲暇，不像在学 学校里每天都跟周末一样。周末一般待在家里休息，有时候我会去阿姨家蹭饭，偶尔去逛街买东西。



来实习很重要一个目的是学东西，我感觉来 eBay 学到的东西还不少，主要是 Java 和 J2EE 方面的东西，尤其是设计模式。eBay 全部的源代码和文档都是对全体员工开放的，所以每个人都可以从中学到很多东西。但是我觉得这也有弊端，我现在简历上一般会写在 eBay 实习过，对 Java 的 Web 开发有 经验之类，然后对方往往就会给你一个这方面的 offer，就失去了尝试其他方面工作的机会了。其实我现在正在申 MS 的实习，还有百度的招聘，推荐人都把我推到类似的工作职位上去了，很无奈。其实我觉得我做什么方面的工作都没问题的。（又是盲目自信么？）

暑假经历了不少事情，见识了不少以前没见过的东西，人或多或少成熟了一点吧~继续奋斗吧，向着我定下的[三个目标](#)前进。



October 2009 by 令狐冲

Google@Beijing

话说 Google 里 Engineer 一般是要用 Linux 的，什么谷歌拼音啦，搜狗输入法啦全部葛屁，只能老老实用 SCIM（什么？你用的不是 SCIM？）。我当时就在想为啥 Chrome 有 Linux 版，谷歌拼音就没有呢？看来不是总部生的孩子待遇是不一样的。扯远了，继续说 SCIM。可是这 SCIM 吧在我机器上面就是调不出来，导致我必须去一个叫 www.inputking.com 的网站用一个极其难用的线上输入法，其智能程度尚不如 www.google.cn 的拼音智能推荐功能。日子一天一天过去了，正当我渐渐遗忘 SCIM 的时候，我迎来了实习生转正面试，头一面是一位叫 [suzhe](#) 的大哥。吃饭的时候我将此事告诉了组里的同事：

我：“第一个面我的是 [suzhe](#)，第二个是……”

同事：“哈哈哈哈哈，[suzhe](#) 大叔啊……”

我：“怎么啦？”

同事：“你必须去找他问问输入法为什么不能用了”

我：“啊？”

同事：“他就是 SCIM 的作者啊”

我：“……”

居然第二天吃早饭正好碰见 [suzhe](#) 大哥，上去打了个招呼：

我：“Hi [suzhe](#)，你下个星期面我啊”

[Suzhe](#)：（看了我一眼）“就是你啊”

我：……

果然又被大牛 bs 了……



又一日，晚上去 K 歌，中间出去了一趟，回来的时候又多了一些人。正好轮到我点的《珊瑚海》，问之有没有 mm 一起唱啊，话音未落一 PPJJ 大声响应。很好。没想到我刚开口唱半句就被鄙视了——“音不对！低了半度”。好吧，又被 bs 了。待轮到 mm 的部分的时候，oh my god，太牛了，声音奇好无比。全部唱下来（还好撑住了，不至于太丢脸），那 PPJJ 意犹未尽的样子，又来了首韩红的，依然是 Oh my god，太震撼了……后来组里的同事小声告诉我：“别介意啊，人家北大合唱团团长”……

又一日，中午在餐厅吃饭时突然觉得 1 点钟方向有异样，有巨牛出没迹象。望去只见一人穿一紫色 ACM/ICPC Final T 恤，上有 Tsinghua U. 字样，再定睛一看，这人怎么这么像楼教主？后来打听得知楼教主今天手机电池爆炸了，过来蹭 BG 寻求一下安慰。

又，去实习之前湖南卫视有一期《天天向上》做的是给 Google 打广告，找了一批 Google 的员工去上节目。实习过后据不完全统计，去上节目的人里面，我的 Manager 一人，共事一人，一起 K 过歌一人，一起打过篮球一人。

总结：Google 里面遇见传说中的牛人的机会还是蛮多的。



读的不是研，是寂寞

第一次学院大会，到会三百多个人，后来大多再没有见过，见过的也大多叫不出名字，我的同学们。

-- houshui

团长要我用这个主题写篇文章，恰好我就是这么看待这一年的研究生生活的。

研究生生活是一段非常特殊的经历，周围的人们以一种与此前截然不同的方式与你相处，仿佛拿到研究生证的一瞬间你就已经身处另一个世界。老师们仍然上课，但会跟学生商量怎么上课，商量怎么考试，跟学生开玩笑，但再不会来管学生，老师们甚至还主动抹



平了及格线这个最后的纠结。老师和学生忽然变得平等起来，但距离感也前所未有的强烈。而同学这个概念，已经名存实亡，如果夸张一点，可以说早已缩水成了室友。记得 ZSN & MSTC Baby 婚礼那天，回去的路上 Mazha 感慨说，“以前在学校的时候，只要上 88 所有的朋友都跑不了，工作了，连见面的机会都没有了，参加婚礼才能碰得着”。

实验室是研究生生活的重心，导师和学生的关系也前所未有的微妙。很多人背地里称呼导师为老板，大家听了也不觉得突兀。有些事我也觉得奇怪，有些老板把学生骂得跟狗一样，他的学生多到分成两个实验室，有些老师从来不骂人，学生一届比一届少；有些学生把老师当老板，老板让他去实习，有些学生把老板当老师，老师只想让他去转博。有些人对研究生阶段的师生关系看得特别悲观，认为导师和学生之间，几乎只剩下雇佣关系。我觉得那只是少数特例，师生关系不佳的原因无非是导师的忙碌、对学生漠视以及学生对老师的过度敬畏和不主动沟通。学生对于导师个人以及实验室的项目都有自己的看法，但真正考虑这些事对自己有什么影响的不多，能够和导师主动交流的更是少之又少。本科生升为研究生时，还有人去考虑是不是换个导师，换个实验室，有不少人顶住压力做了这样的事，但研究生换导师，退学出国的，则极为罕见。不论读博还是读硕，很多人都有一个观念，就是一定要毕业，以及一定要按时毕业。不知道是不是每个人都认真考虑过这个观念的合理性和必要性。

研究生生活最大的坏处，就是整个环境和体制压缩了我的交际圈，把我从想象中的正常生活里抽离出来，周围的人和事变得比以前疏离。不过物极必反，因为与周围人和事的疏离，我开始关注自己的内心。而研究生生活的好处，则在于相对充裕的可自由支配的时间以及对现实的正视。大部分研究生其实有很多可自由支配的时间，博士的可自由支配时间更多，虽然这些时间往往是不连续的。我见过有些人是经过一年的艰苦复习才终于考上了研究生的，他们之中的一部分觉得自己收获的东西完全对不起自己的付出，不禁去担忧自己现在的努力是不是又会在将来让自己后悔。两年半的研究生学制不比四年的本科学制，能分给彷徨的并不太多。好在，先行一步的同学们总是愿意献身说法，工作的会跟你说，“待在学校多轻松啊，工作了压力太大”，出国的会跟你说“忙死了忙死了，睡觉的时间也没了”，忽然你就觉得自己居然是最幸福的。读了研究生，我仍然不清楚自己喜欢什么样的生活，但见过的多了，多少知道哪些生活是自己不喜欢的，这也是一种进步吧。

想起无间道里的两句话，觉得适合用来结尾，“出来混总是要还的”，而“路怎么走，要自己选”。



飘到枫叶国的熊猫 by 南宫星

学术

在滑铁卢是要选 3 门课 1.5 分来维持 full time 学生身份的。于是考虑到有毕设的要求都选了叫 499 的一个大 Project 课加两门 CS 方向选修（之前学院提到过的"和滑铁卢的学生一起上一门类似软工的大课，然后在期末 presentate 这个应该是指 ECE 391/491/492 的系列课，然后被系里的小米告知是不能选因为是 exchange。那是给本校学生做的毕设，FML \ (/ _ \) /)

生活

这个项目第一次去没有学长，加上 3 个人分别从祖国的大西北，天子脚，好江南三个地方分三天出发的飞机=___=（机票应该要早买阿阿阿...），到了这里后找了个住在学校边上 2 分钟（当然是开车！走路乘以 8.....）的坑，约莫 380 刀（1:6.3<-- 口胡，现在都 6.6 了-->左右，汇率市场是云谲波诡阿...）一个月，偶的这个恰好带了落地窗+外面小阳台+100 平方的后花园...HOHO。ECE 的项目不要学费，不过还是交了 380 多刀的必须的费用+保险，所以公交车费就被强制包含进去了... 不过由于交通相当便捷有严格时间表+不太会晚点的车，作为出门首选的"免费"公交还是很划得来的，因为单程要 2 刀半....滑铁卢的地方不大，啥都有但是除了超市啥都只有一个_-，开始两周的休闲活动就是逛超市\农场比价格算汇率 blabla... 接下来两周开始就学乖了，休闲活动是逛超市找打折\特价物品... 有时候还是会有惊喜的譬如 3 刀一大袋虾，3 刀一大个 pizza，3 刀一大盒哈根达斯..... 食物的开销大致会在 1-3 刀自己烧，5-7 刀外面吃每人每顿正餐。一顿日本寿司 buffet 约莫 16 刀 <_+ 比那啥绿啥回转好多了..._+> 一个月 700-1k 是个比较合理的开销。当然如果能在现在诡异的汇率上打点脑筋应该可以省下 10%以上。

其他

这里的中国人很多，特别是在学校周围，Orientation 的时候见到数学学院的 7 成是中国人（ps: PPMM 居多），在 CS 也有大概 4 成... 以致于在路上走撞倒别人了说 sorry 都很有



可能会听到一句没关系=_=#这里的旅行社也很多，似乎如果有米国 visa 去玩的话很划算。加东枫叶红了，强烈有着去外拍的打算 HOHO。来后第二周去了美加边境的传说中的大瀑布，来回车费+一顿赌场中的地道中国自助居然只要 15 刀（被人说赌场是慈善机构了..），冬天的瀑布景观听说会大不一样，于是还打算再去~

花絮

- 在机场要打电话给 carpool，于是买了张电话卡，找到了公用电话后拿起电话卡往机器里塞，5 分钟后发现电话卡后有个密码，是用那个来拨号的...
- 第一天去采购房东开车带去的，在超市结算的时候发现房东只买了一块钱的东西于是打算拿过来帮他一起结账，结果房东的表情...象是说我们要拿这个里面还有.....
- 第二天去超市买食物，结果发现那些形容词和好像是法语衍生出来的单词雌雄前缀都看不懂..... 于是无奈从此带上了文曲星...
- orientation 活动多，而且基本上都带送 pizza 饮料的食物可以蹭，于是在一个月黑风高的夜晚，我们伙同几个 CS 的 PHD，ZJU 的 ISEE 的学长，新加坡/香港同学，一共蹭了 3 场三明治/pizza/咖啡.. 从此开始了各种蹭食活动
- 去超市买特价 pizza，3 刀多一个，结果在自助 Checkout 上发现刷成原价了，于是乎举手检查后，这个 pizza 就送给偶了..... 最后 4 刀买了两 pizza...
- 说起中国人多，那么
 - ✓ 某同学和偶一起上网络课，边上做着个东亚面孔象中国人，课间某同学去搭讪，
Q: r u Chinese?
A: yep
Q: Oh~~ 你好你好 我也是 很高兴认识你..
A: what? i can't follow u...(confused)
Q: -_-!! Where do u come from
A: HongKong
Q: %#@^#&\$%*%^(!#!\$#@!
 - ✓ 偶去上人工智能的 TUT，TA 是个东亚面孔，开场白: "hello, this is my first year to teach on this, 然后呢, e., then let's..." 然后我就瀑布汗下来了...
 - ✓ 某天第一次去 coordinator 的实验室开会，立马被一个印度人搭讪了，说要请教我们一个 C 里的 UTF-8 问题，说因为我们可能遇到过....<---难道中国脸长得就那么明显么.....



中秋出游

10月3日中秋节，白天暴走九溪+龙井山，傍晚若缘bg元祖雪月饼，晚上西湖赏月杀人，最后部分人通宵K歌。



金牌保研

10月11日ACM/ICPC Asia regional 合肥赛区，浙江大学yukkuri、ForWarD两支队伍均获得金牌，俱乐部成员hsys@yukkuri、winsty@ForWarD得以保研。

2009年秋季纳新

9月初至10月初接受报名表。10月10日至12日对通过筛选的同学进行面试。10月12日公布纳新名单（附）。

PP的纳新gmd

CG

王一(流浪的流心)
王嗣徽(suncorner)
刘嘉瑜(鳞片)
傅莹
齐书尧(forella)

OG

张玮(zzh119)
杨雨荷(becky)
黄晓(hnlysmile)
张杨(zy1991)

PG

王嘉慧(牛奶曜)
潘遥
刘澈(liustrive)
王昱辰(锦年)

TG

陆俊哲(tyh1k)
李喆(aislex)
朱玉可(秋心)
曾鸣聪(彼得)
柳云超(-Fly ぁ梦- Flymeng)
张耕源(连过五人)
莫璐怡(moondly/依然微笑)



破冰

10月18日上午在植物园破冰，内容为杀人和三国杀，下午在bgs举行会员大会，通报了一些俱乐部近期事宜。



2009 年换届

10月19日正式公布换届结果（附）。

President : pluskid (张驰原)

----- YQ -----

Campus Leader : 景楼重天 (毛翔博)

CG Leader : 湖边 (党楠)

OG Leader : gaojiaze (高嘉泽)

PG Leader : 星未眠 (董立星)

TG Leader : 水寒 (李成)

----- ZJG -----

Campus Leader : LUNA-C (吕唱)

CG Leader : jiabin (张嘉斌)

OG Leader : pynm (彭妍)

PG Leader : LUNA-C (吕唱)

TG Leader : pluskid (张驰原)



老人吐槽

CS03，大四在 ATC 实习，本科毕业后去了百度，10 月 5 日在宁波结婚，现居北京。

马铃薯 (cyourworld)

想了半天，煽情了：我觉得我 25 岁人生中最正确的两件事情，一个是找到了人生的伴侣，另一个是加入了 MSTC。

CS02，本科毕业先去道富，后就职于某创业公司，现居杭州。

Madog

7 年前的这个时候，我刚刚下课，欢庆即将到来的周末；5 年前的这个时候，我记得加入了 MSTC；4 年前的这个时候，刚刚开始了第一天的实习工作；3 年前，毕业了；1 年半前，跳槽了；1 个月前，MSTCbaby 有 baby 了；半个月前，鼠 mm 和田叔分别婚了；上周，seemy 婚了；昨天，小 re 求婚成功了；今天，我在电脑前码字……

时间总是过的太快，转眼间，当年一起组织活动的 MSTCers 早已各奔东西成家

立业。而把我们联系在一起的，不是 MSTC，而是在 MSTC 的时候所累积下来的感情。MSTC 不仅仅是一个社团，更像是一个家，在这里除了技术，更能体会到在其他地方所体会不到的那种深深的友情。昨天看着 ddmm 们一起为小 re 祝福，一起在饭桌上谈笑风生，让我觉得这个家仍然温馨仍然幸福。

我们是幸运的，因为 MSTC 我们聚集在了一起。

MSTC, My Home!

Rhythm (RhythmBox)

MSTC 就像一个魔方，纵有万般缤纷组合，六个轴心面永远不变：开放、自由、热情、进取、快乐、家。

CS02，MSTC 四大才子之一，本科毕业后先去网易杭研院，后去百度，现居北京。



CS02, MSTC 四大才子之一,
本科毕业后先去网易杭研
院, 现在在乐港科技。

Mazha (蚂蚱)

MSTC 是我大学记忆中最闪光的部分, 在 MSTC 收获的友情、成长和荣耀, 一直是我前进路上最大的支持。看到 MSTC 的精神在一代代传承着发扬着, 我很有幸也很自豪地说: 我, 是她的一员。

tomwon (汤王)

推背图

第六十一象: MSTC 必将一统江湖

SC03, 某 IT 企业员工,
现居杭州。

CS08 研, 现居玉泉。

水之精灵 (neptunee)

对我来说, MSTC 与其说是一个社团, 不如说是一个家庭。回想着 MSTC 的点点滴滴, 想起当初和 MSTCers 一起贴海报, 一起 FB, 甚至一起自习的那些日子, 让我觉得无比的甜蜜。总之, 加入 MSTC 是进入大学以来我做的最明智的决定。最后, 祝 MSTC 越来越好。

Poggale (盆景男)

大一新生时的我, 也加过好多社团, 有把会员费当成是收入来源的社团, 有活动次数少得可怜的社团, 有组织涣散的社团, etc. 相比而言, MSTC 是一个大家庭, 交流的氛围非常浓厚, 互相帮助也是常有的事。美中不足的是, 我认识 MSTC 太晚, 才加入一年多, 就毕业闪人了。

BME03, 毕业后去中国
民航信息, 现在在阿里
巴巴云计算。

在 MSTC 里, 我最大的收获应该是认识了 N 多的朋友吧, 而且全是大牛级人物, 让我等小辈有了一个学习的榜样。只是老人们毕业之后, 四散而去, 不知何年能再重逢。茶几上有一只杯具应该是, 朋友换了一批又一批, 却只有可怜巴巴的那么几个朋友能相伴一生。



地科 02, CS06 研, MSTC
四大才子之一, 毕业后先去了欧酷, 现在在乐港科技。

中华人民共和国六十年十月二十六日, 为公司产品上线前两天, 我加班到凌晨, QQ 上遇见嫫儿同学, 前来问道: “萝卜可曾为 MSTC 月刊创刊号写点什么没有?” 我说“没有”。她就正告我, “萝卜还是写一点罢。”

这是我知道的。时常我也在反思我的人生, 回顾这二十多年来那些重大的战略失误或者是难以启齿的伤痛, 每每此时, 一阵阵小媳妇心态涌上心头, 情何以堪! 为了找到一些心理的平衡我往往去回忆一些令我欣慰的事, 常令我沾沾自喜的便

萝卜

是我在 MSTC 的这段回忆。经常一听到一个同学说我以前是 MSTC 的, 虽然不认识, 但也有一种莫名的亲切。

很难去形容这是什么样一段经历, 这是一群什么样的人。一段难忘的经历? 一段美好的回忆? 一群有激情有梦想的人? 一群有技术有能力的人? 似乎这样的形容过于平淡了。我贴近我心目中的 MSTC 的形象应该是“富有激情又踏实努力, 为人低调又做事张扬, 活泼又严肃, 羞涩又大方”。

午夜只言片语, 不成体统, 就当乱弹一番了~~

特别鸣谢

以下小朋友们哈：

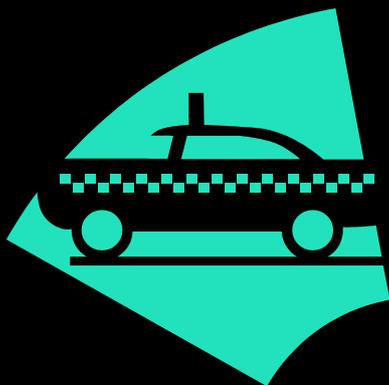
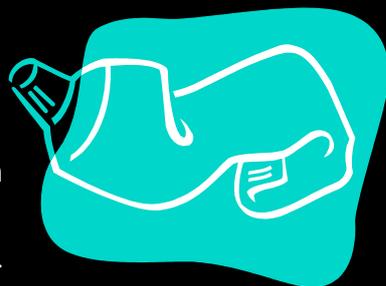
(排名不分先后)

感谢金牌月刊苦力 若缘、Stingrey、pluskid、lam

感谢辛勤的供稿者 cerror、itsuhane、quark、hzqtc、

令狐虫、houshui、南宫星

最后特别感谢各骨灰的寄语~



大家多多支持，**欢迎来稿！**

如有任何意见或反馈，请跟帖 98 月刊楼

欢迎鸡蛋鲜花！

(鸡蛋请炒好送到 8 舍 鲜花送到 bgs 吧)