

敬请:智慧的提问

How To Ask Questions the Smart Way

-- 王刚 <yafrank at 126 dot com> 版本《提问的智慧》

- 2001 by Eric S. Raymond
- 中文版Copyleft 译者 D.H.Grand
- 以及 CPyUG 典章:回答的智慧
- 强烈: 建议使用技巧: 如何有效地报告Bug <http://www.chiark.greenend.org.uk/%7Esgtatham/bugs-cn.html>

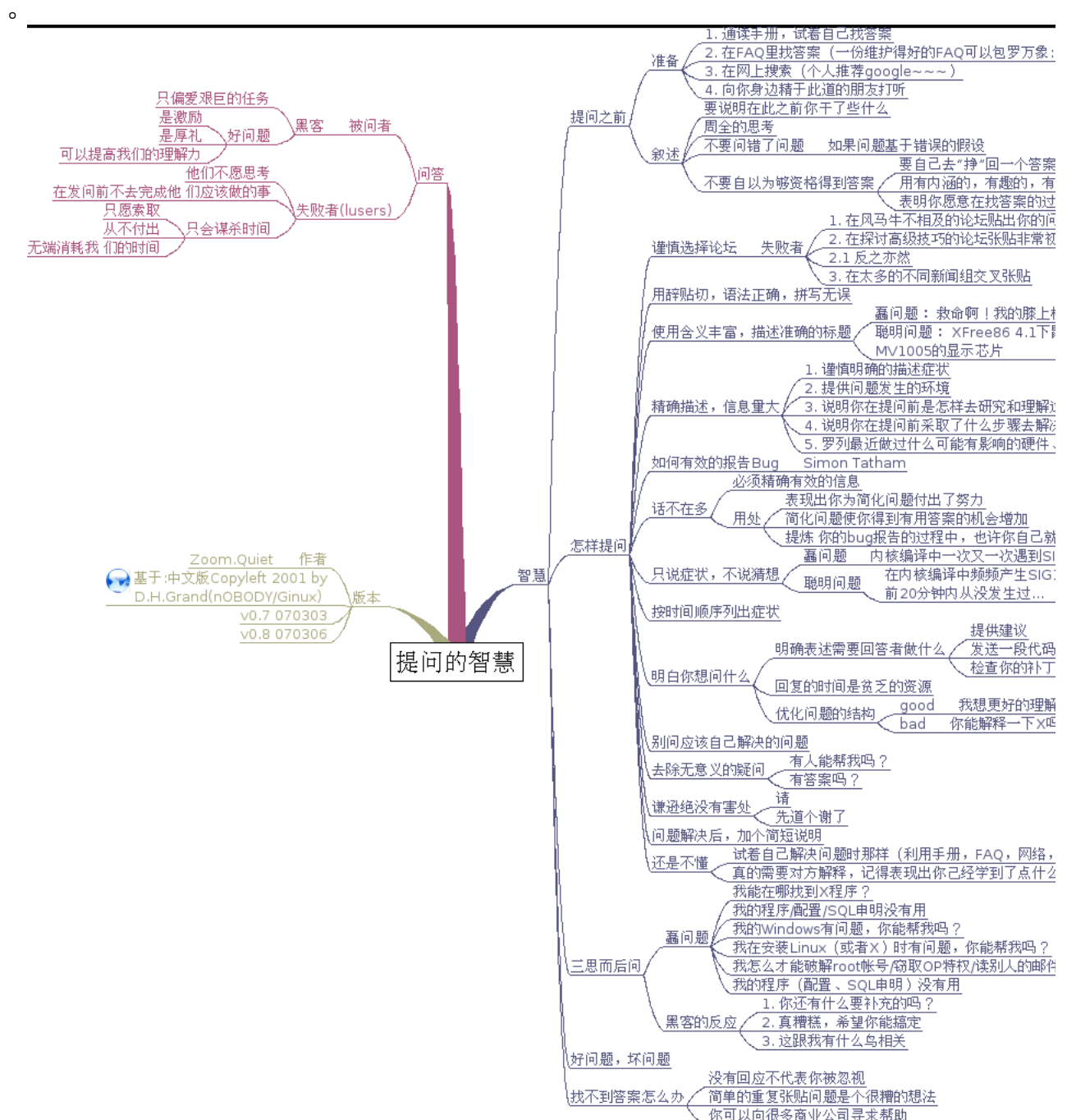
严正参考: 如何向开源社区提问 · Issue #545 · seajs/seajs

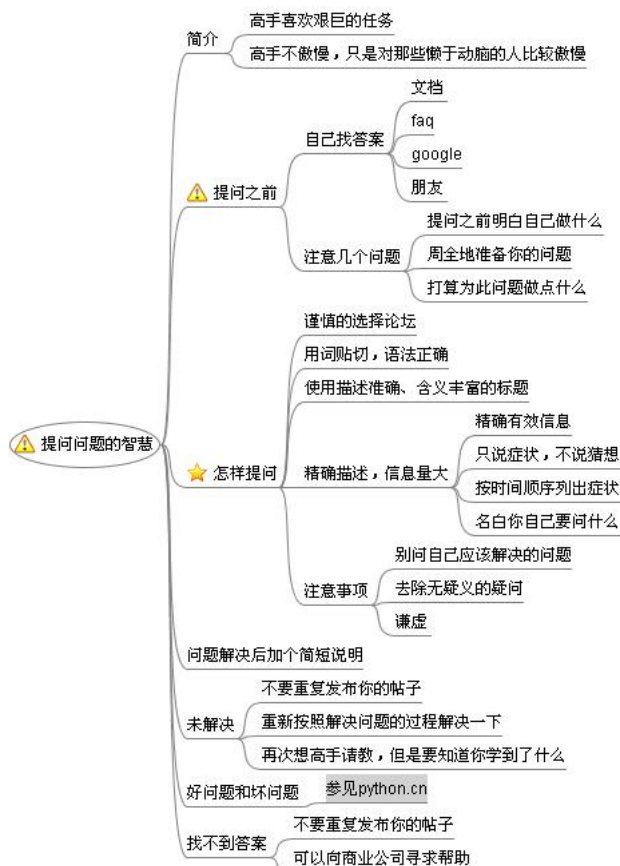
图说:提问的智慧

在你打算向某个论坛或者某个maillist请教时, 请好好看看下面这张图谱, 呵呵。尽快提高自己的能力是也!

图谱化理解::

- VYM 格式 questions4mind.vym





- FreeMind文档 asdquestion.mm

菜鸟提问需知

速记

1. 不要大喊大叫
2. 使用一个好读的呢称
3. 不要抱怨
4. 语气请低调
5. 请帮助他人
6. 尊重他人

全文

对新人来说，最好的学习方式是：去找个DM，让他手把手的教你。你会发现，这只不过是游戏而已，并不难学。新手需要的不是160RMB的“精装”PHB，而是一种态度——认真、热情、严谨、谦虚，呃，还有好奇心和想象力，在这个快餐文化的时代，并不是人人都具有这些东西。不要到IRC的公共频道里去寻求帮助，那里都是BT和烂人，他们最喜欢KUSO那些大喊大叫的新人 - ____ -b

——哥布林诗人给新手的忠告

- 以上文字采自最深的地下城，这是一个很上等的网站。在我做网游开发的时候，经常去看。我想以后我还会重新回到那里，毕竟我这种人对一个到处是坑的站点还是充满亲近感的。
- 上面那段话，给我深刻的印象，因为在国内的大多数IT社区，包括CSDN、Python中文，以及大大小小的QQ群，小论坛等等，无不充满了大喊大叫的新人。
- 很不幸，那里也游荡着各种不靠谱的BT和烂人……于是，我们每天都可以看到有无数的新人在抱怨对他们充满敌意的恶劣环境，这样的环境下，年轻人如何成才？中国如何有希望？等等等等……
- 然而，实际上，初学者们不正确的交流方式，大大影响了他们得到帮助的机会。总结起来，以下几种表现，几乎是一定会被BT的。
 - 大喊大叫。特别是在QQ群里，使用过大的字体，过于鲜艳的颜色，都会严重影响阅读。本来QQ默认的窗体尺寸就小，你放一堆二号字上去，何况又是鲜红或嫩绿的二号字，让人怎么读？技术交流首先要保证消息的可读性，不要没事儿走视觉系。要知道每天坐办公室啃代码的老家伙们骨子里跟非主流艺术青年八九是不搭界的。黑色或靛蓝等传统的书写颜色可以让人舒服的阅读。
 - 使用一个不好读的ID。这一点可能不太容易被人理解。然而如果你的ID别人都没办法念出来，大家就很难给你一个私密的回复。与其说是技术原因，更多的是心理因素。理由同上。不要指望每天坐办公室的老coder会跟你一样喜欢玩儿非主流视觉系。事实上靠谱的老程序员们往往跟这帮子家伙很不搭界。至少他们进入编码的心理状态时，是非常不喜欢看到一个花里胡哨的家伙跳出来在他面前画火星文。我从来没见过哪个靠谱的家伙顶着一个火星文ID在IM上写一堆鲜红嫩绿的二字号初号字。这些人最多也就是在QQ或MSN的签名上放几个表情而已。我曾遇到一个非常悲惨的实例。有个朋友多年来一直因为ID的问题在我们这个小圈子里被人BT，事实上他的名字一个火星文都没有，甚至也是一个合乎文法的名词词组。然而这个名字非常奇怪，念上几遍就会让人有想去抽他的冲动。于是大家不约而同的对她进行疯狂的BT。好在这位兄台有非常强健的神经，就像踩不死的小强一样生存了下来，并且在这种互相精神攻击的恶劣交流状态下学习到了很多东西。他的顽强还表现在工作的风格上。事实上这些年来他解决了很多很BT的技术问题，写出的代码搞定了很多我看来完全发疯的需求。就这一点，我是非常佩服他的学习精神、钻研精神以及娱乐大众的牺牲精神。为此我就不告诉你们他的ID叫“想你的我”，因为太难念，我们一致称他为水煮鱼。另外我要说的是，如果你没有水煮鱼那种被人切成一片还要丢进锅里水煮再浇上辣椒也不怕的精神，还是起一个正常点的，好念的ID吧……事实上用自己真名的新人往往会得到比较亲切的帮助，不信你试试？
 - 不要没事儿就在公共地盘抱怨没有人帮助你。要知道在自由软件社区帮助你的人，九成九都是义务劳动，他们还要牺牲自己的工作和休息时间。没人欠你的。有个家伙因为五分钟没有人在QQ群里回答他的白痴问题，就开始大骂，于是他被我用各种方式羞辱了几个星期。没人会同情他。好吧，我不是一个心胸开阔的人。就是我这样一个小心眼又虚荣的家伙曾经在没有任何报酬的情况下翻烂了一本牛津高阶词典来翻译Python tutorial。这件事是我在失业的时候蜷缩在冬天冰冷的小屋里完成的，而且五

年来我还把这件傻事重复做了四遍，正准备做第五遍。如果你能做到同样的程度，你也一样有资格去BT那些乱喊乱叫的家伙。如果你觉得应该支持比我更有RP的前辈，那可以无视我，请你去花钱买一本陈儒的Python源码剖析，或者去吹捧一下ZQ、Limodou、黄冬这些为大家做了很大贡献的人（不知道他们是谁？你真的是Python中文社区的成员吗？那你一定是新警察……）。

- 尽量少问软件破解的问题。我是说在自由软件社区。我不想争论道德高度的问题，这样的问题已经讨论过太多次了。我只想说，在自由软件社区找人要破解版或注册机，真的看起来很蠢。
- 语气请低调一些。趾高气扬的菜鸟只会被加倍的BT。这一点不需要多解释了，提醒一句：这个世界比你想像的NB的多。
- 遇到一个厚道的解答者，请珍惜，不要把他当作垃圾桶，什么白痴问题都丢过去问。学着自己解决问题永远是一种美德。如果你是一位青春美少女，对方是一个单身的宅男coder，另当别论，看在他厚道的为新人解答问题，任劳任怨的份上，嫁给他吧。
- 尽量帮助你一样的菜鸟，不要BT他。这会使高阶的人力资源得到更有效的利用。就好像不要总是在游戏公会里请求顶级玩家帮你做新人任务一样，其实找个高你5级的帮手就很豪华了。同样你也应该多帮一帮比你更低级的玩家。
- 被BT时，请先检讨自己是否使用了挑衅技能。对于一个小圈子，往往BT新人会是老鸟们的重要福利之一。乱凑热闹或者灌水过多，都有可能招致灭顶之灾。在QQ群里发连锁消息尤其愚蠢。如果被BT，请先不要回击，通常来说，正是你这个新来的家伙破坏了生态平衡。如果觉得留下来学习更重要，就闭上嘴巴，多听少说。

IT行业人员的提问建议

里面提到的两个例子比较典型。另外后面有人评论中也给了一个例子。

- via: <http://www.cnblogs.com/confach/archive/2008/09/10/1288034.html>

实例郁问

- 这几天对几个网友的请教方式颇感无奈。这里举2个实例：
 1. 有个网友因为项目比较急，而且之前也没有怎么接触过该项目的一些相关知识。正好我对这方面熟悉，于是找到我给出一些建议和提示。我大概知道了其要点，然后从头到尾给出了一些架构和技术上的要点。我觉得凭这些应该没有什么大问题了。没想到在未来几天里，该网友一直问我一些我已经解答过的问题。更意外的是同一问题问了至少5遍。我很郁闷，就问了一句，你工作几年了，他告诉我4-5年。我不再说什么了。如果工作4-5年，按照我的理解是不应该有这样的情况的。
 2. 另外一个网友因为一个小问题卡住了。我说了一下我的想法。他说他以前也做过，没有问题，而且特别坚持自己的意见。最后我只能说可以试一试我的建议。一个礼拜之后，他看见我，问我同样的问题，我惊诧道，“你还没有解决吗？”。他说还没有。我继续把我的建议重复了一遍。他过了一会高兴的回答我说可以了。

在上面的2个实例中，我感慨太多了。

- 针对第一件事情，我觉得至少存在以下方面的问题：

1. 做事情太着急了
2. 应该有把握整个project的能力
3. 应该能够控制自己的心态
4. 问问题之前最好总结一下，或者是思考一下人家给你的提示。不要一二再再二三的去问同一个问题大于3次。

- 对于工作4-5年的人，已经培养了自己解决问题，分析问题的方法。而且在把握一个项目上应该有一定的经验。冷静思考，沉着应对，都是现在浮躁的环境必须要有的。这些技能和心态和技术没有直接的关系。对中国的IT业，我一直认为是比较浮躁的。在这样的环境下，难道不能有自己的做事风格来行走着浮躁之面上吗？
- 针对第二件事情，我觉得可以这么理解：

1. 每个做IT人骨子里或多或少都自以为是，包括我自己也是。请教别人时还是那样
2. 既然自己没有解决，何不试一试别人的建议呢，也许会给你带来意想不到收获。
3. 多听听人家的意见或建议，对自己是有帮助的。放下一些不必要的面子。

- 如果坚持自己是对的，又不肯听从人家意见，那你问人家干嘛呢？
- 对待学习，对待工作，我们确实应该保持自信，但这绝不是自以为是。虚心请教他人，听取别人意见，在整个过程中我们会学到不少东西。别总以为自己是对的，每个人的知识面和知识的掌握程度都是有限的，这样自己的理解出现偏差和错误在所难免。
- 从另外一个角度上讲，我们在处理学习和工作上，应该知道一件事情如何去问，如何以什么样的心态去做。对于一个工作4-5年的人出现上面的情况我觉得实在不应该。这样的情况在刚工作时存在。随着自己的阅历增长，在态度，方式上都会逐渐成熟，都会有自己的一套方法。这些方法在面对一些复杂事情，未熟悉事情都是有帮助的。

Service Is Living 建议

所以我的想法是：

1. 戒浮躁，踏踏实实做事情
2. 谦虚，自信，不是自以为是
3. 有自己的做事风格，包括工作方式和心态。
4. 沉着，冷静，从大局考虑。
5. 做事情（例如向别人请教）前自己先做好必要的准备，想想。
6. 态度
7. 沟通。

上面的情况中，我是否真的让人家明白了，也许自己以为说清楚了。后来我也开始有些着急了，我心里还是告诫自己应该平和一些。尽管有些做的不好，我也一直按照上面的思路做。转几个评论：

- #23楼 2008-09-10 11:28 李胜攀

"做事情（例如向别人请教）前自己先做好必要的准备，想想。" 这一点很重要 我记得很多次别人问我问题时，我都会让他把问题详细描述一遍，结果到最后，不用我回答，他自己就知道怎么做了。

- #24楼 [楼主] 2008-09-10 11:33 Confach

@lkc 冷静，很重要。我见过好多人一遇到问题就开始这里找找，哪里问问，其实有这些时间还不如先分析一下原因所在，然后再继续寻找，这样节省时间，也锻炼自己的分析能力

- #30楼 2008-09-10 11:43 绝世无才

博主说得很对，碰见再急的问题时，也需要冷静，冲动是解决不了技术问题的 一次，两个医院客户那里同时有一个新模块（门诊收费）上线，这个模块我做的，并且也测试了几周，但我知道真正用的时候肯定会有问题，而且我并不在现场，于是我不断的告诉自己，再急也

要冷静，于是我远程指挥两个同事，并且不断的收到Bug，不断的编译 发邮件过去。从早上 8 点开始直到 1 2 点医院下班，我竟然发送了将近 3 0 封邮件。我和同事交流的过程中不断的说：冷静 他们说晕了，挺不住了，要不就停吧，此时我也说冷静。结果又经过一个下午的苦战，两边的系统都成功的上线了。此刻比踢完一场球还痛快！

- #36楼 2008-09-10 14:01 4681 [未注册用户]

30楼做的东西真牛,到最后还有这么多的bug.

- #38楼 2008-09-10 14:40 绝世无才
 - @4681

有些时候，我们无法决定自己的软件装到客户机器上后是如何运行的，但我们要保证在自己的电脑上和测试用的电脑上是能正常运行的，另外我们更不能保证客户一定要按我们想象的那些步骤进行操作，所以问题是无法避免的。
我们能做到的是经过仔细的需求分析，设计架构，编写代码，出现问题时，尽量不要因为一个修改带来更多的问题（耦合性）当然，最重要的是需要时时保持一个冷静的头脑

--

- 刘未鹏(pongba)|C++的罗浮宫 <http://blog.csdn.net/pongba>
- TopLanguage <http://groups.google.com/group/pongba>

反馈

- 嗨嗨！谢谢有心人！不过应该提供 FreeMind 原文件哪？否则，相关的链接可是点击不了的是也乎是也乎 -- ZoomQuiet
- ZoomQuiet，我现在都不知道什么时候保存这个页面的，这是怎么回事呢？ -- dfar

AskForHelp (2013-11-15 02:09:29由ZoomQuiet编辑)