

黑客与画家

黑客行为必须包含三个特点：好玩、高智商、探索精神

(1) 使用计算机以及所有有助于了解这个世界本质的事物都不应受到任何限制。任何事情都应该亲手尝试。

(2) 信息应该全部免费。

(3) 不信任权威，提倡去中心化。

(4) 判断一名黑客的水平应该看他的技术能力，而不是看他的学历、年龄或地位等其他标准。

(5) 你可以用计算机创造美和艺术。

(6) 计算机使生活更美好。

根据这六条“黑客伦理”，黑客价值观的核心原则可以概括成这样几点：分享、开放、民主、计算机的自由使用、进步。

1. 为什么书呆子不受欢迎

所以，如果智力本身与“受欢迎”无关，为什么聪明的小孩一直不受同龄人的欢迎呢？我认为，答案就是他们真的不想让自己受欢迎。

书呆子不受欢迎的真正原因，是他们脑子里想着别的事情。

2. 黑客与画家

真正竞争软件设计的战场是新兴领域的市场，这里还没有人建立过防御工事。只要你能做出大胆的设计，由一个人或一批人同时负责设计和实现产品，你就能在这里战胜大公司

开发优秀软件的方法之一就是自己创业

一个是自己开公司的话，必须处理许许多多与开发软件完全无关的事情

创业的另一个问题是赚钱的软件往往不是好玩的软件，两者的重叠度不高

正确的合作方法是将项目分割成严格定义的模块，每一个模块由一个人明确负责。模块与模块之间的接口经过精心设计，如果可能的话，最好把文档说明写得像编程语言规范那样清晰。

普通黑客与优秀黑客的所有区别之中，会不会“换位思考”可能是最重要的单个因素。

判断一个人是否具备“换位思考”的能力有一个好方法，那就是看他怎样向没有技术背景的人解释技术问题

软件的部分功能就是解释自身。为了写出优秀软件，你必须假定用户对你的软件基本上一无所知

程序写出来是为了让人看懂它的算法，附带告诉计算机如何执行。

3. 不能说的话

明明是专横武断、毫无依据的错误观点，但是大多数人却深信不疑，受到影响而不自知

这就是找出“不能说的话”的第一种方法：判断言论的真伪。

异端邪说

关注“异端邪说”，是找出“不能说的话”的第二种方法

我认为，类似的思维机制存在于每个人的头脑中，很多看似叛逆的“异端邪说”，早就“潜伏”在我们的思维深处。如果我们暂时关闭自我审查意识，它们就会第一个浮现出来。

如果我们通晓未来，那么找出当代的那些表面上正确、实际上可笑的想法是一件很容易的事。但是，不可能做到这一点。幸运的是，我们可以找到一种几乎有同样效果的替代方法：回顾过去

我们可以去找那些过去被认为理所当然，如今却被认为不可思议的事情，这是用来找出我们自己正在犯下的错误的第三种方法

第四种寻找“不能说的话”的方法：寻找那些一本正经的卫道者，看看他们到底在捍卫着什么

第五种方法，可以找出“不能说的话”，那就是去观察禁忌是如何产生的

做一个异端是有回报的，不仅是在科学领域，在任何有竞争的地方，只要你能看到别人看不到或不敢看的东西，你就有很大的优势

同样，如果你能“远远地”跳出传统思维，提出让别人一听就脑袋轰一声炸开的惊人观点，那么你就在“小小地”跳出传统思维方面不会有任何困难。要知道，人们把后面的这种情况称为“创新”。

与笨蛋辩论，你也会变成笨蛋。

4. 良好的坏习惯

下一个潮流

现在，可以有另一种方法发布软件，用户再也不会被迫当上系统管理员了。互联网软件运行在服务器上，用户界面就是网页。对于普通用户来说，使用这种新型软件将更容易、更便宜、更机动、更可靠，通常也比桌面软件更强大。

代码之城

对于开发者来说，互联网软件与桌面软件最显著的区别就是，前者不是一个单独的代码块。它是许多不同种类程序的集合，而不是一个单独的巨大的二进制文件。设计桌面软件就像设计一幢大楼，而设计互联网软件就像设计一座城市：你不仅需要设计建筑物，还要设计道路、路标、公用设施、警察局、消防队，并且制定城市发展规划和紧急事件的应对方案。

互联网软件的发布规则是：它运行不了，你就无法发布；一旦它能运行了，你就可以立刻发布。

这个行业的老手可能会想：你说得好听！软件运行不了，就不发布，但是如果你已经对外承诺了明确的发布日期，到时却没有准备好，怎么办？这个问题听起来有道理，但是事实上，你不会在互联网软件做出这样的承诺，因为它根本没有“版本”这个概念。你的软件是连续性渐变的，某些更新也许比较重大，但是“版本”这个概念不适用于互联网软件。

现在，创业公司有更多的理由选择互联网软件创业，因为开发桌面软件越来越乏味了。如果你现在开发桌面软件，就不得不接受微软公司的授权条款，调用它的 **API**，为它那个 **bug** 百出的操作系统伤透脑筋。历尽了千辛万苦，你最终写出了一个大受欢迎的软件，这时你可能会发现，你所做的一切其实只是在为微软公司做市场调查。

为什么不尝试一下

如果你是一个黑客，并且梦想自己创业，可能会有两件事情令你望而却步，不敢真正开始采取行动。一件是你不懂得管理企业，另一件是你害怕竞争。可是实际上，这两件事都是没有通电的电篱笆

首先，管理企业其实很简单，只要记住两点就可以了：做出用户喜欢的产品，保证开支小于收入。只要做到这两点，你就会超过大多数创业公司。随着事业的发展，你自己就能琢磨出来其他的诀窍

其次，让我们来看看竞争

6. 如何创造财富

金钱不等于财富

财富是最基本的东西。我们需要的东西就是财富，食品、服装、住房、汽车、生活用品、外出旅行等都是财富。即使你没有钱，你也能拥有财富。如果有一台魔法机器，能够按照你的命令变出汽车，为你洗衣做饭，提供其他你想要的东西，那么你就不需要钱了。要是你身处南极洲内陆，再多的钱对你也是无用的，因为没有东西可买，你真正需要的是财富。

要致富，你需要两样东西：可测量性和可放大性。

选择公司要解决什么问题应该以问题的难度作为指引，而且此后的各种决策都应该以此为原则。**Viaweb** 的一个经验法则就是“更上一层楼”。假定你是一个手脚敏捷的小男孩，身后有一条壮硕的大狗正在追你。你跑到楼梯口，这时应该上楼还是下楼？我觉得应该上楼。如果下楼的话，大狗可能跑得跟你一样快。上楼的话，大狗的庞大身躯就将成为劣势。不错，跑上楼你会比较吃力，但是大狗会感到更吃力。

在实际操作中，这就意味着我们故意选择那些很困难的技术问题。假定软件有两个候选的新功能，它们创造的商业价值完全相同，那么我们总是选择较困难的那个功能。不是因为这个功能能带来更多的收入，

而是因为它比较难。我们很乐于迫使那些又大又慢的竞争对手跟着我们一起走进沼泽地。创业公司就像游击队一样，喜欢选择不易生存的深山老林作为根据地，政府的正规军无法追到那种地方。我还记得创业初期我们是多么筋疲力尽，整天都为一些可怕的技术难题绞尽脑汁。但是，我还是感到相当高兴，因为那些问题连我们都觉得这么困难，那么竞争对手就更会认为是不可能解决的。

这不仅是创业公司运作的好方法，更是创业公司的本质。风险投资商（**VC**）知道这个道理，为它起了一个名字——进入壁垒（**barriers to entry**）。

我认为有三个原因使得我们对赚钱另眼相看。第一，我们从小被误导的对财富的看法；第二，历史上积累财富的方式大多名声不好；第三，担心收入差距拉大将对社会产生不利影响

事实上，财富与金钱是两个概念。金钱只是用来交易财富的一种手段，财富才是有价值的东西，我们购买的商品和服务都属于财富。你到海外旅行时，不用看当地人的银行账户就会知道你来到的是富国还是穷国。你只要看看他们的财富就行了：建筑、街道、服装、健康状况等。

财富从何而来？人类创造出来的。回到农业时代，这个概念就更容易理解。那时大多数人都务农，许多东西都需要自己生产出来。房屋、牲畜、谷仓等都是每个家庭自己生产出来的。这就很明显地说明，财富总量不是固定不变的，不像大饼那样会被分光。如果你想要更多的财富，自己生产就可以了。

这在今天的社会也成立，虽然已经很少有人直接创造财富供自己使用了（少量的家务活除外）。我们大多数人都在为其他人创造财富，然后用创造出来的财富交换金钱，再用金钱交换我们需要的另一种财富▲。

好设计是简单的设计

好设计是永不过时的设计

好设计是解决主要问题的设计

好设计是启发性的设计。

好设计通常是有点趣味性的设计

好设计是艰苦的设计

好设计是看似容易的设计

好设计是对称的设计

好设计是模仿大自然的设计

好设计是一种再设计。

好设计是能够复制的设计

好设计常常是奇特的设计。

好设计是成批出现的。

好设计常常是大胆的设计

10. 编程语言解析

“你用什么语言并不重要，重要的是你对问题是否有正确的理解。代码以外的东西才是关键。”

这当然是一派胡言。各种语言简直是天差地别。

Perl 语言

结果就是产生了一些也许可以称为“头重脚轻”的语言：它们的内核设计得并非很好，但是却有着无数强大的函数库，可以用来解决特定的问题。（你可以想象一辆本身性能很差的小汽车，车顶却绑着一个飞机发动机。）有一些很琐碎、很普遍的问题，程序员本来要花大量时间来解决，但是有了这些函数库以后，解决起来就变得很容易，所以这些库本身可能比核心的语言还要重要。所以，这些奇特组合的语言还是蛮有用的，一时间变得相当流行。车顶上绑着飞机发动机的小车也许真能开，只要你不尝试拐弯，可能就不会出问题[▲]。